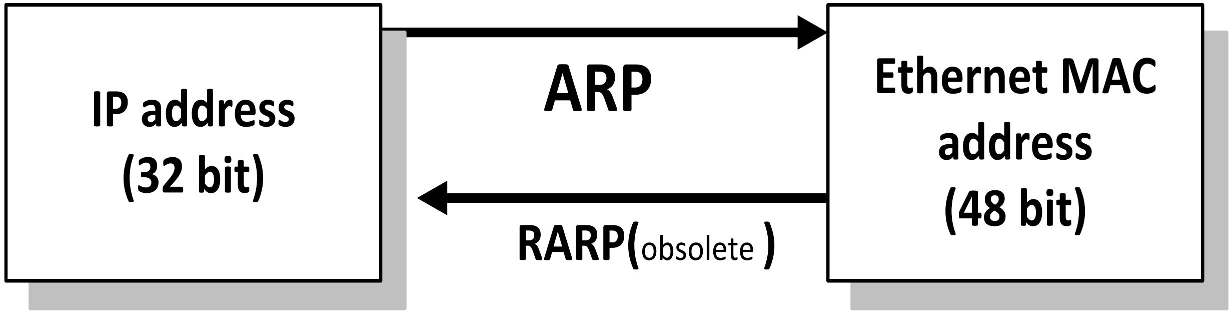
Address Resolution Protocol (ARP), RFC 826

Prof. Lin Weiguo

Sept. 2019

ARP & RARP

- Note:
  - The Internet is based on IP addresses
  - Data link protocols (Ethernet, FDDI, ATM) may have different (MAC) addresses
- The ARP and RARP protocols perform the translation between IP addresses and MAC layer addresses
- We will discuss ARP for broadcast LANs, particularly Ethernet LANs

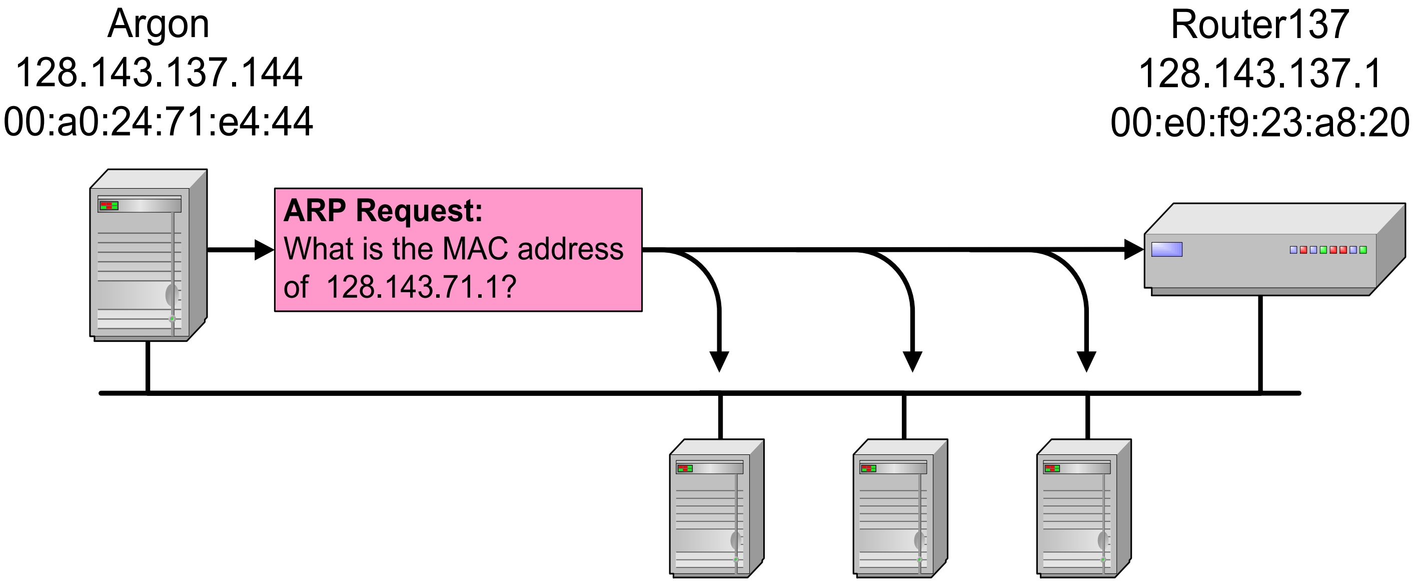


Problem of mapping

- The problem of mapping Internet addresses to physical addresses is known as the address resolution problem.
  - Each Ethernet device has its own unique number. Change the card and you change its physical address.
  - Physical address are 6 bytes long, too large to multiplex within an Internet address.
  - New machines can be added to the network with no disruption of service.
  - But, adding new hosts should not require reconfiguring existing hosts to inform them of the new machine.

Address Translation with ARP

- **ARP Request:**
  Argon broadcasts an ARP request to all stations on the network: “What is the hardware address of 128.143.137.1?”



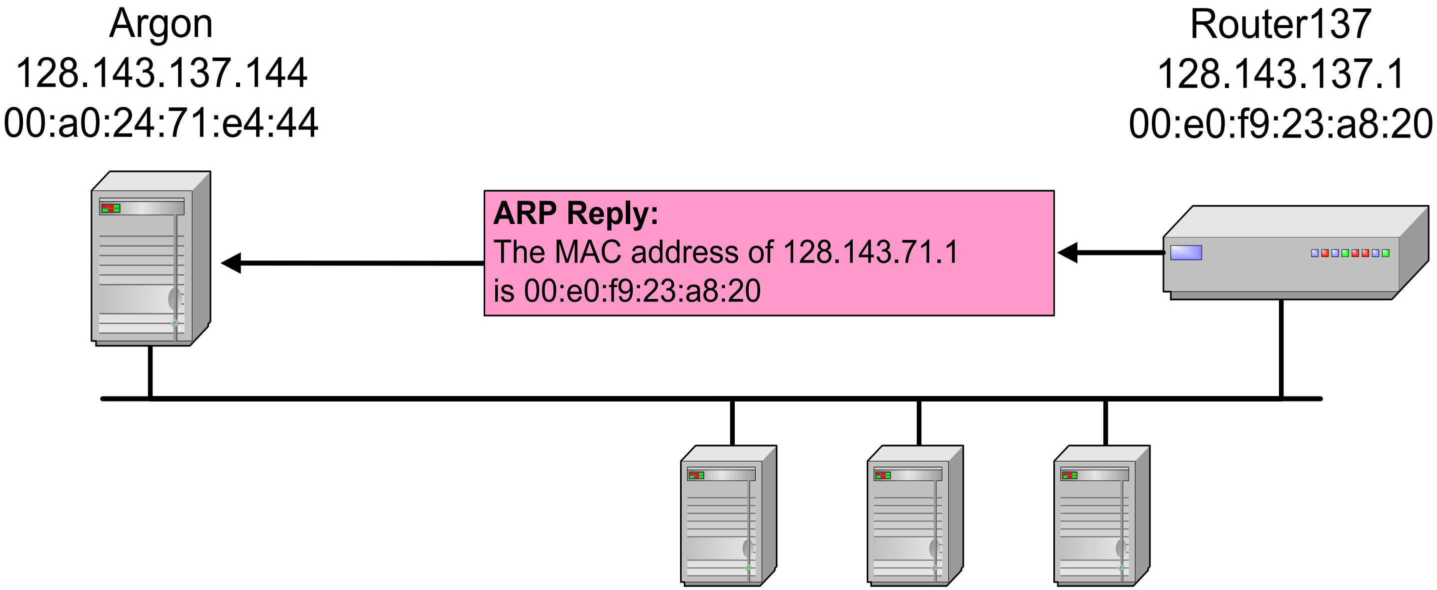
Argon
128.143.137.144
00:a0:24:71:e4:44

ARP Request:
What is the MAC address of 128.143.71.1?

Router137
128.143.137.1
00:e0:f9:23:a8:20

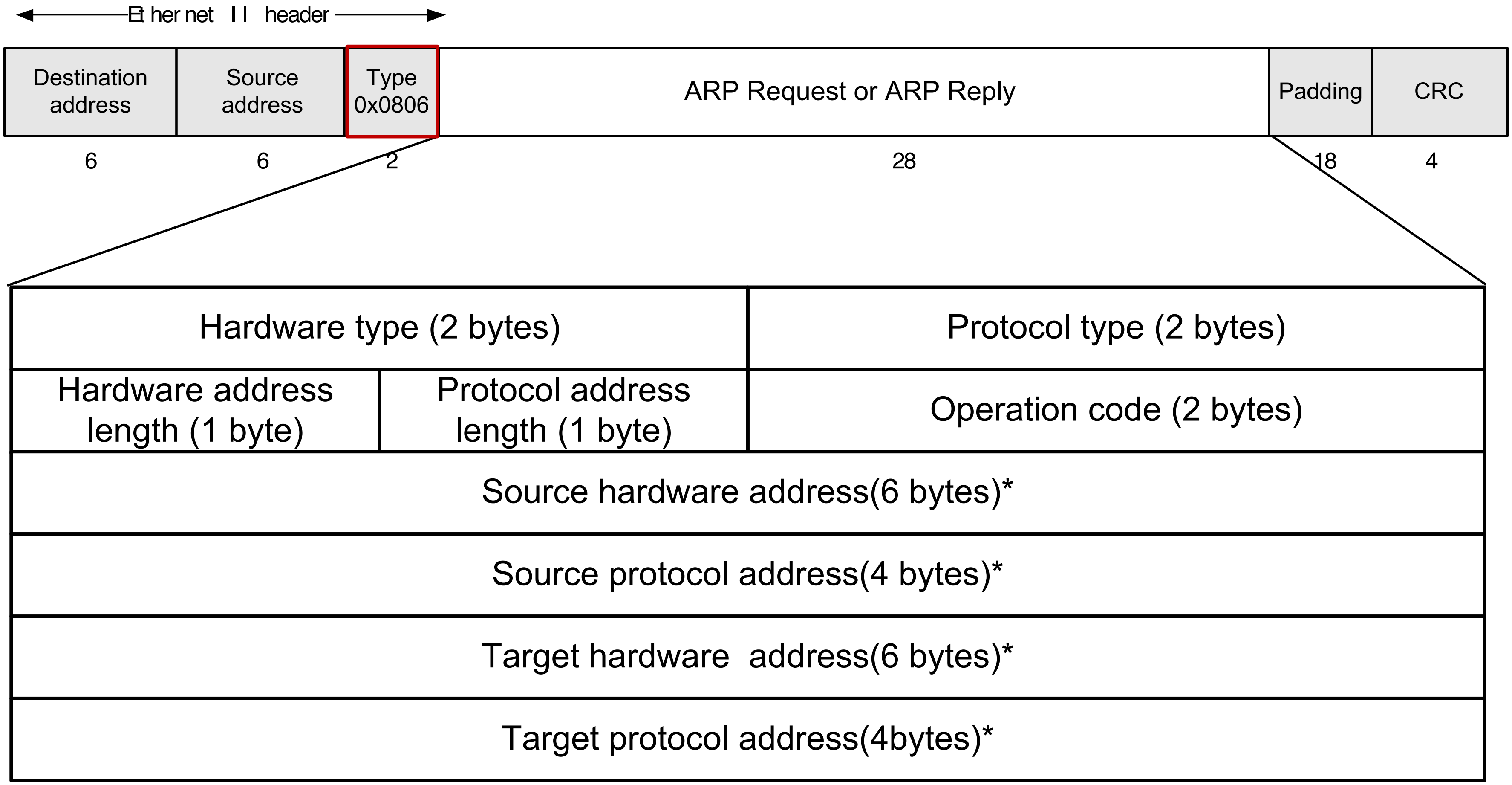
Address Translation with ARP

- **ARP Reply**:
  Router 137 responds with an ARP Reply which contains the hardware address



ARP Reply:
The MAC address of 128.143.71.1
is 00:e0:f9:23:a8:20

Ethernet Frame Format (ARP Packet)



Ethernet II header

Destination address	Source address	Type 0x0806	ARP Request or ARP Reply	Padding	CRC
6	6	2	28	18	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(6 bytes)*

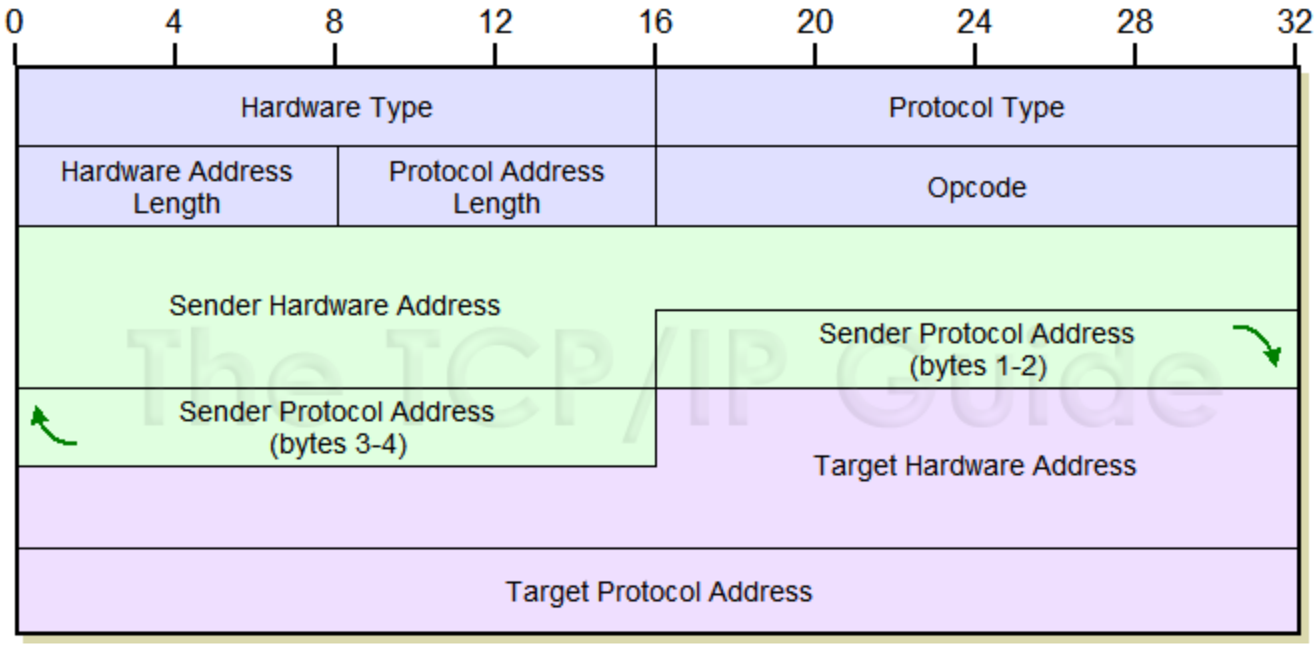
Source protocol address(4 bytes)*

Target hardware address(6 bytes)*

Target protocol address(4bytes)*

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

ARP Packet Format



Hardware Type

Protocol Type

Hardware Address Length

Protocol Address Length

Opcode

Sender Hardware Address

Sender Protocol Address (bytes 1-2)

Sender Protocol Address (bytes 3-4)

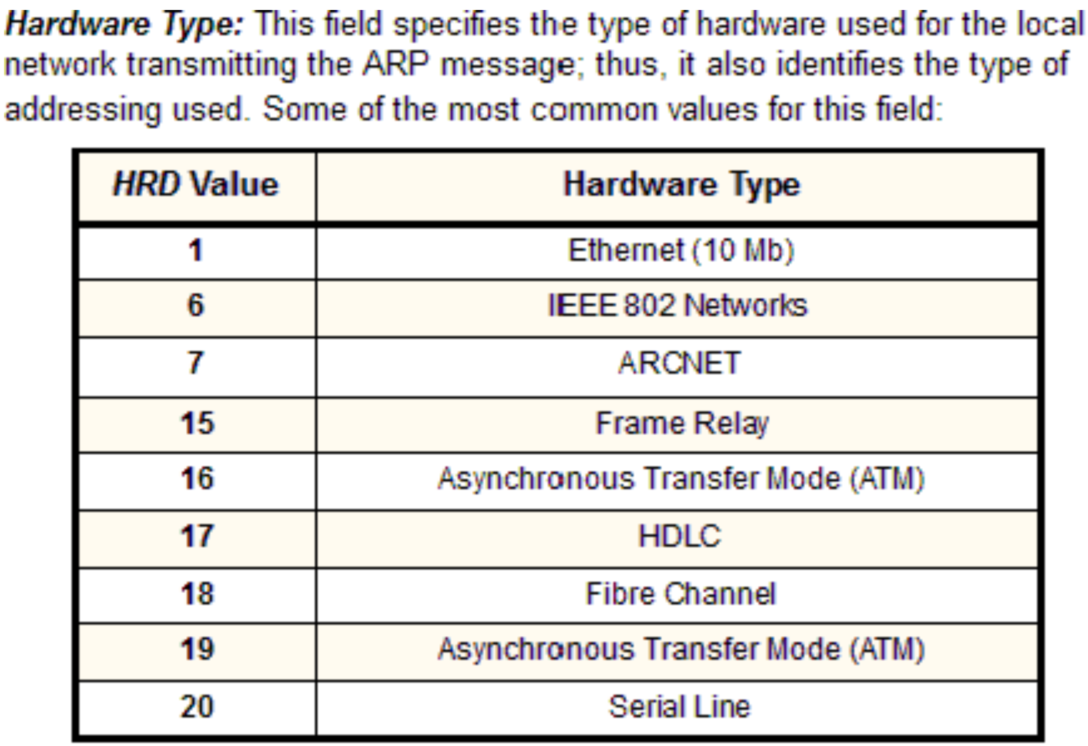
Target Hardware Address

Target Protocol Address

Ethernet frame of ARP

- RFCs – 826, 1122
- Size
  - 64 bytes frame
    - Frame Header :14 bytes
    - ARP packet: 28 bytes
    - Padding: 18 bytes
    - FCS: 4 bytes CRC32
- Characteristic
  - Requests are addressed to a broadcast address.
  - Replies are addressed to an unicast address.

Hardware Type Field



Hardware Type: This field specifies the type of hardware used for the local network transmitting the ARP message; thus, it also identifies the type of addressing used. Some of the most common values for this field:

HRD **Value**	**Hardware Type**
1	Ethernet (10 Mb)
6	IEEE 802 Networks
7	ARCNET
15	Frame Relay
16	Asynchronous Transfer Mode (ATM)
17	HDLC
18	Fibre Channel
19	Asynchronous Transfer Mode (ATM)
20	Serial Line

http://www.iana.org/assignments/arp-parameters/

Protocol Type Field

- ***Protocol Type:*** This field is the complement of the *Hardware Type* field, specifying the type of layer three addresses used in the message.
- For IPv4 addresses, this value is 0x0800, which corresponds to the EtherType code for the Internet Protocol.

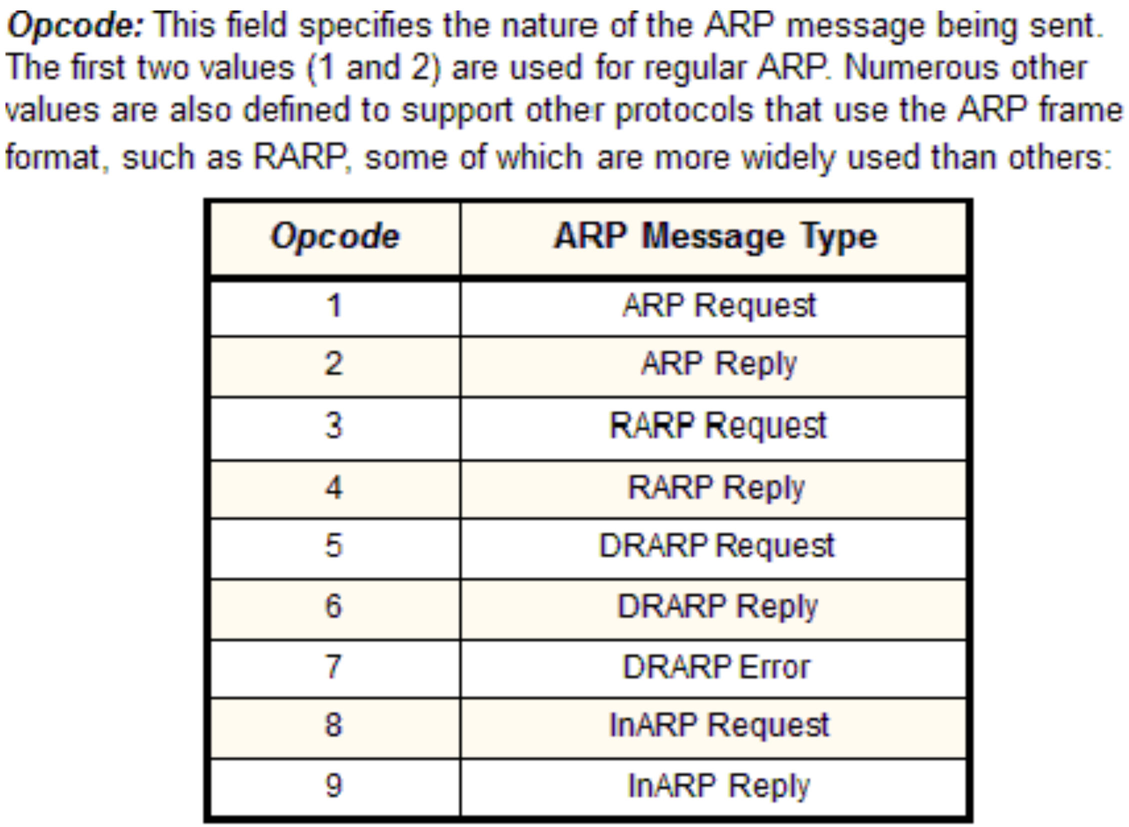
(HLN)Hardware Address Length Field

- ***Hardware Address Length:*** Specifies how long hardware addresses are in this message.
- For Ethernet or other networks using IEEE 802 MAC addresses, the value is 6.

(PLN)Protocol Address Length Field

- ***Protocol Address Length:*** Again, the complement of the preceding field; specifies how long protocol (layer three) addresses are in this message. For IP(v4) addresses this value is of course 4.

Operation Code Field



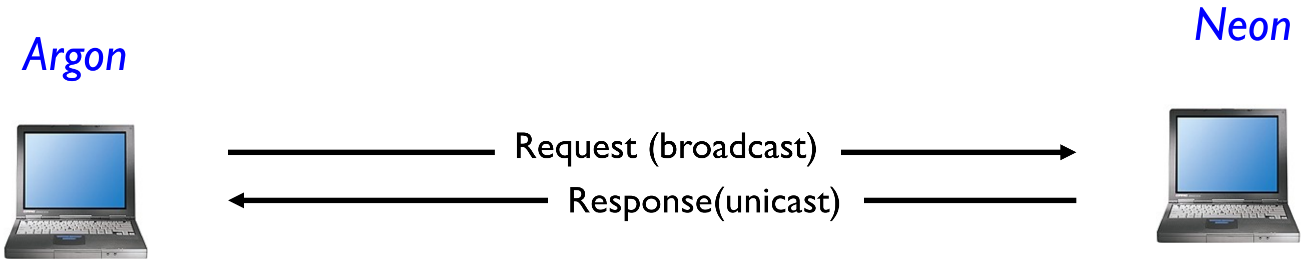
Opcode: This field specifies the nature of the ARP message being sent. The first two values (1 and 2) are used for regular ARP. Numerous other values are also defined to support other protocols that use the ARP frame format, such as RARP, some of which are more widely used than others:

Opcode	ARP Message Type
1	ARP Request
2	ARP Reply
3	RARP Request
4	RARP Reply
5	DRARP Request
6	DRARP Reply
7	DRARP Error
8	InARP Request
9	InARP Reply

Sender and Target Addresses

SHA	(Variable, equals value in *HLN* field)	***Sender Hardware Address:*** The hardware (layer two) address of the device sending this message (which is the IP datagram source device on a request, and the IP datagram destination on a reply, as discussed in the topic on ARP operation).
SPA	(Variable, equals value in *PLN* field)	***Sender Protocol Address:*** The IP address of the device sending this message.
THA	(Variable, equals value in *HLN* field)	***Target Hardware Address:*** The hardware (layer two) address of the device this message is being sent to. This is the IP datagram destination device on a request, and the IP datagram source on a reply)
TPA	(Variable, equals value in *PLN* field)	***Target Protocol Address:*** The IP address of the device this message is being sent to.

Example 1



Argon

Neon

Request (broadcast)

Response(unicast)

Ethernet Frame

- Destination Address:FFFFFFFFFFFF
- Source Address: 00:a0:24:71:e4:44
- Ethernet Type=0x0806 (ARP)

Address Resolution Protocol (request)

- Hardware type: Ethernet (0x0001)
- Protocol type: IP (0x0800)
- Hardware size: 6
- Protocol size: 4
- Opcode: request (0x0001)
- Sender hardware address: 00:a0:24:71:e4:44
- Sender protocol address: 128.143.137.144
- Target hardware address: 00:00:00:00:00:00
- Target protocol address: 128.143.137.1

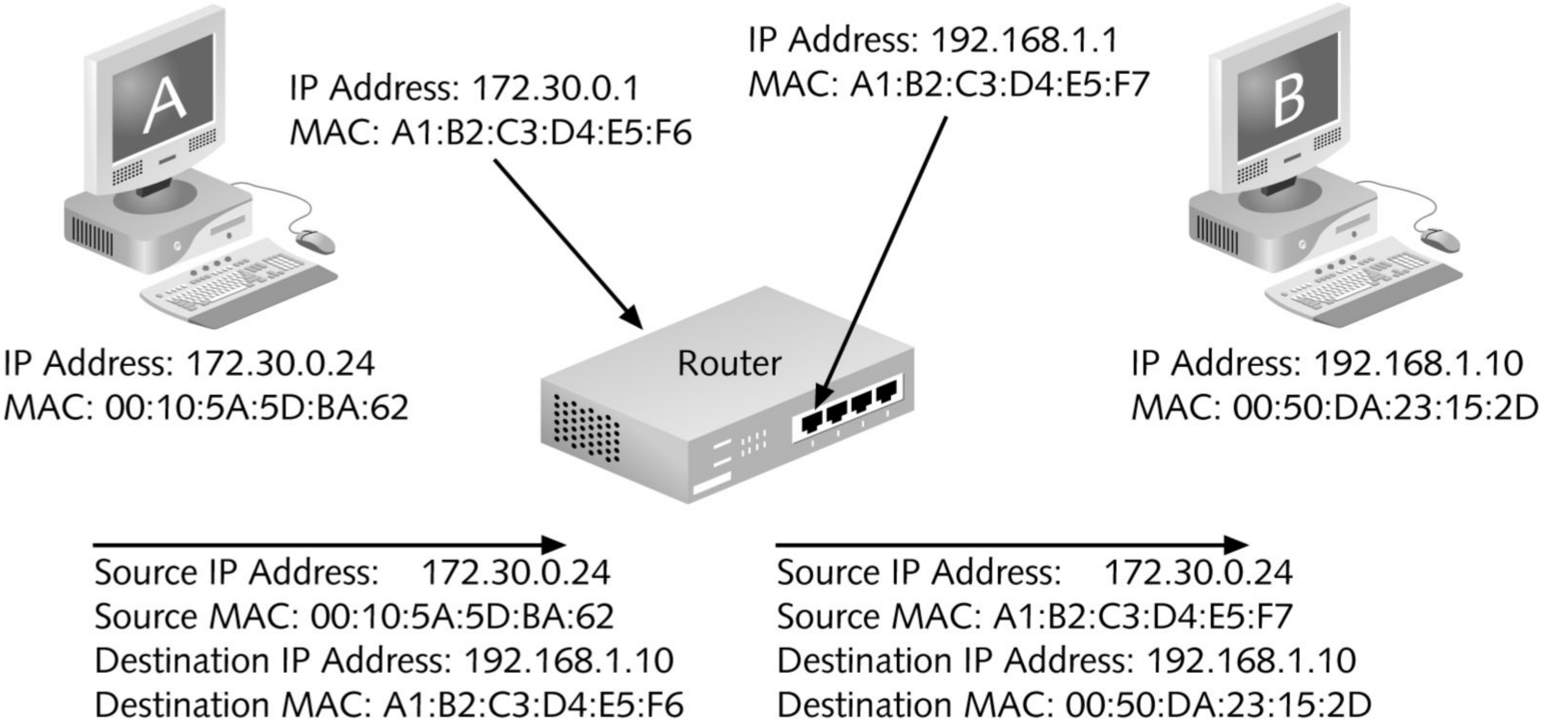
Ethernet Frame

- Destination Address: 00:a0:24:71:e4:44
- Source Address: 00:e0:f9:23:a8:20
- Ethernet Type=0x0806 (ARP)

Address Resolution Protocol (reply)

- Hardware type: Ethernet (0x0001)
- Protocol type: IP (0x0800)
- Hardware size: 6
- Protocol size: 4
- Opcode: reply (0x0002)
- Sender hardware address: 00:e0:f9:23:a8:20
- Sender protocol address: 128.143.137.1
- Target hardware address: 00:a0:24:71:e4:44
- Target protocol address: 128.143.137.144

Example 2



IP Address: 172.30.0.1
MAC: A1:B2:C3:D4:E5:F6

IP Address: 192.168.1.1
MAC: A1:B2:C3:D4:E5:F7

IP Address: 172.30.0.24
MAC: 00:10:5A:5D:BA:62

IP Address: 192.168.1.10
MAC: 00:50:DA:23:15:2D

Source IP Address: 172.30.0.24
Source MAC: 00:10:5A:5D:BA:62
Destination IP Address: 192.168.1.10
Destination MAC: A1:B2:C3:D4:E5:F6

Source IP Address: 172.30.0.24
Source MAC: A1:B2:C3:D4:E5:F7
Destination IP Address: 192.168.1.10
Destination MAC: 00:50:DA:23:15:2D

Figure 3-6 Computer A communicates with Computer B across a router

ARP Cache

- Since sending an ARP request/reply for each IP datagram is inefficient, hosts maintain a table (ARP Cache) of current entries for each network adapter installed.
- The entries expire after 2 minutes.
- Contents of the ARP Cache:

```
(128.143.71.37) at 00:10:4B:C5:D1:15 [ether] on eth0
(128.143.71.36) at 00:B0:D0:E1:17:D5 [ether] on eth0
(128.143.71.35) at 00:B0:D0:DE:70:E6 [ether] on eth0
(128.143.136.90) at 00:05:3C:06:27:35 [ether] on eth1
(128.143.71.34) at 00:B0:D0:E1:17:DB [ether] on eth0
(128.143.71.33) at 00:B0:D0:E1:17:DF [ether] on eth0
```

Arp Command on Windows

- Displays and modifies entries in the ARP cache.
- Syntax
- **arp [-a** [*InetAddr*] [**-N** *IfaceAddr*]] [**-d** *InetAddr* [*IfaceAddr*]] [**-s** *InetAddr EtherAddr* [*IfaceAddr*]]

  Parameters

  -a [InetAddr] [-N IfaceAddr] : Displays current ARP cache tables for all interfaces. To display the ARP cache entry for a specific IP address, use arp -a with the InetAddr parameter, where InetAddr is an IP address. To display the ARP cache table for a specific interface, use the -N IfaceAddr parameter where IfaceAddr is the IP address assigned to the interface. The -N parameter is case-sensitive.

  -d InetAddr [IfaceAddr] : Deletes an entry with a specific IP address, where InetAddr is the IP address. To delete an entry in a table for a specific interface, use the IfaceAddr parameter where IfaceAddr is the IP address assigned to the interface. To delete all entries, use the asterisk (*) wildcard character in place of InetAddr.

  -s InetAddr EtherAddr [IfaceAddr] : Adds a static entry to the ARP cache that resolves the IP address InetAddr to the physical address EtherAddr. To add a static ARP cache entry to the table for a specific interface, use the IfaceAddr parameter where IfaceAddr is an IP address assigned to the interface.

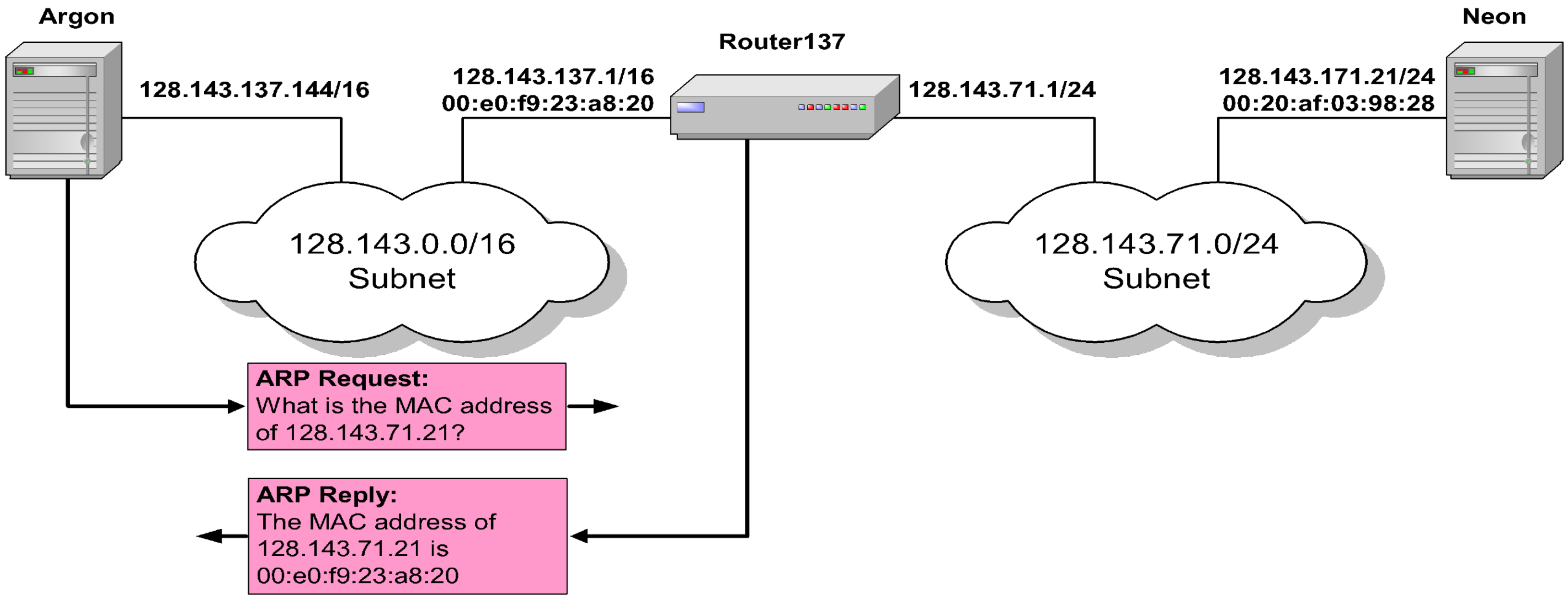
Arp Command examples

- To display the ARP cache tables for all interfaces, type:
  - `arp -a`
- To display a specific entry of ARP cache table
  - `arp  -a 172.16.7.26`
- To display the ARP cache table for the interface that is assigned the IP address 10.0.0.99, type:
  - `arp -a -N 10.0.0.99`
- To Deletes an entry with a specific IP address
  - `arp –d 172.16.7.82`
  - `arp –d *`
- To add a static ARP cache entry that resolves the IP address 10.0.0.80 to the physical address 00-AA-00-4F-2A-9C, type:
  - `arp -s 10.0.0.80 00-AA-00-4F-2A-9C`

(In order to run –d/-a command you'll need admin privileges.)

Proxy ARP

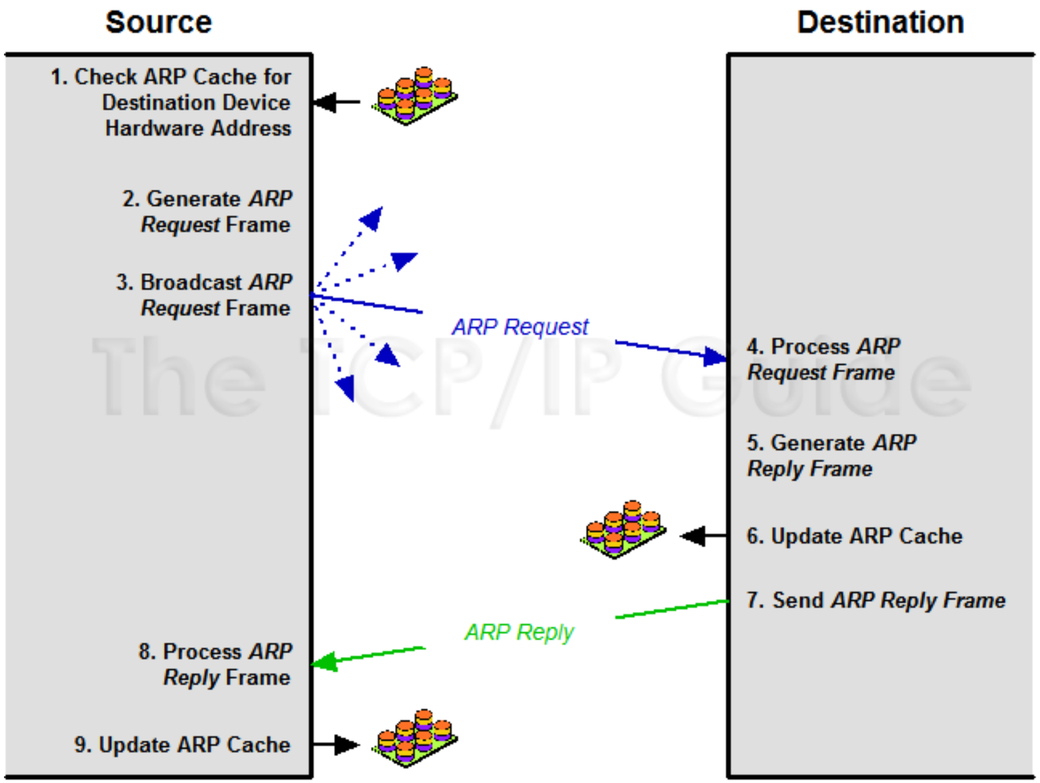
- **Proxy ARP:** Host or router responds to ARP Request that arrives from one of its connected networks for a host that is on another of its connected networks.



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

Review: ARP General Operation



Source

1. Check ARP Cache for Destination Device Hardware Address

2. Generate *ARP Request* Frame

3. Broadcast *ARP Request* Frame

8. Process *ARP Reply* Frame

9. Update ARP Cache

Destination

4. Process *ARP Request Frame*

5. Generate *ARP Reply Frame*

6. Update ARP Cache

7. Send *ARP Reply Frame*

ARP Transaction Process

Things to know about ARP

- Gratuitous ARP(also called a courtesy ARP):
  - When a host is verifying that its IP address is unique, there should not be an ARP Reply frame. (or ARP)
- Microsoft
  - Windows XP timeout value: 2 minutes
  - Vista and 2008 has lowered this time to a random value between 15 and 45 secs
- Linux
  - 60 secs timeout for RedHat
- Cisco
  - ARP command syntax: show arp
  - Configuration of Catalyst 6500 Distribution Switch:
  - The ARP cache timeout on the MSFC is four hours. However, the Layer 2 CAM table times out in 300 seconds by default. This may result in some IP unicast traffic being flooded. In the following configuration the CAM timeout "agingtime" is set to 4hours*60min/hour*60sec/min. = 14400 seconds to match the ARP cache timeout.

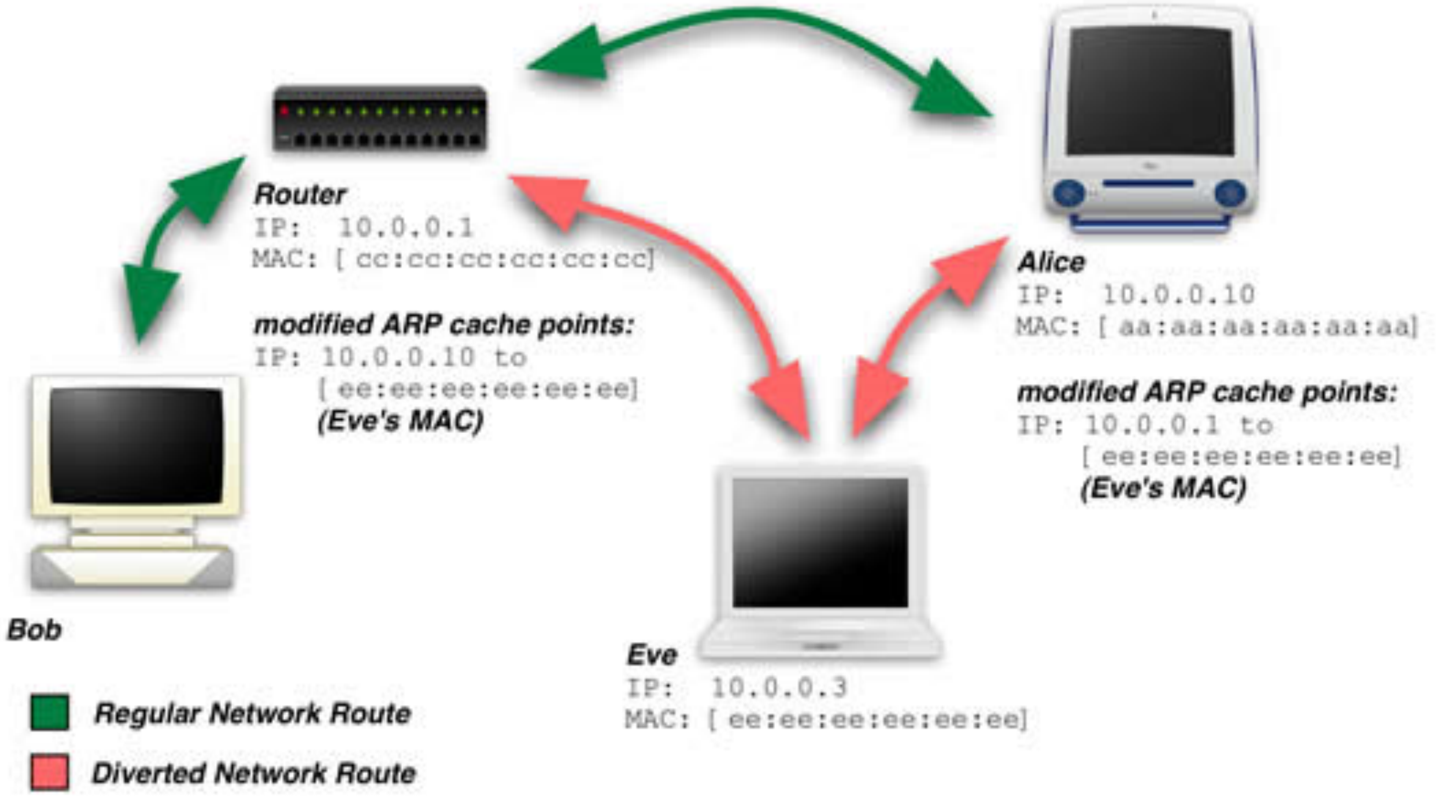
Vulnerabilities of ARP

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

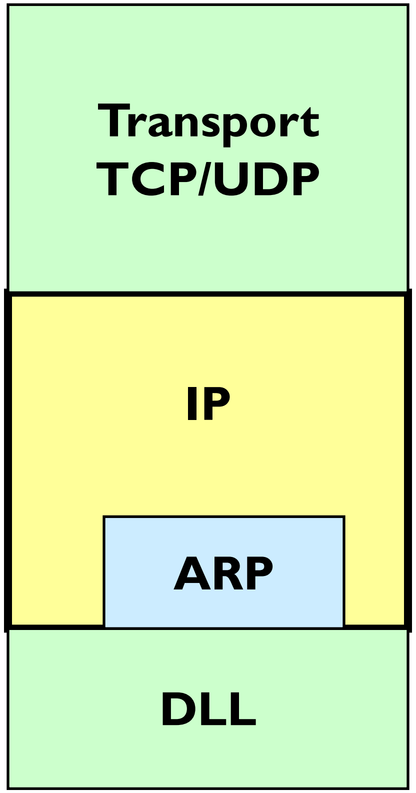
Man in the middle



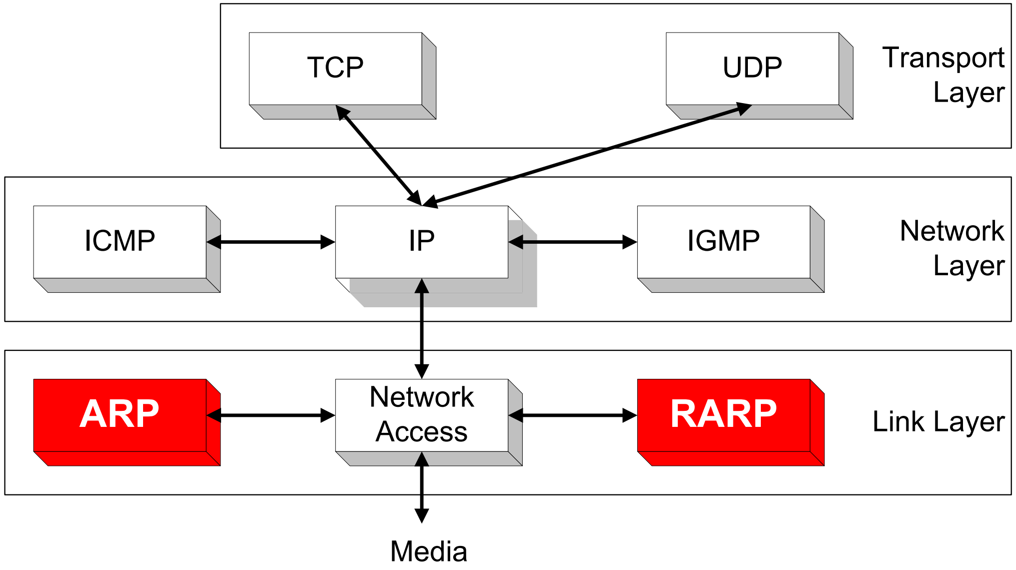
man in the middle attach via ARP spoofing

Layering: Different views

From a layering point of view, ARP sits below IP, but above the data link layer.



While others don't agree. They consider that ARP sites in the data link layer.



arping

- arping is a computer software tool that is used to discover hosts on a computer network. The program tests whether a given IP address is in use on the local network, and can get additional information about the device using that address.
- The arping tool is analogous in function to ping, which probes hosts using the ICMP at the Internet Layer. Arping operates at the Link Layer using the ARP for probing hosts on the local network only, as ARP cannot be routed across gateways (routers).

Arping implementations

- There are two popular arping implementations. One is part of Linux iproute2 suite, and cannot resolve MAC addresses to IP addresses. The other arping implementation, written by Thomas Habets, uses the platform-independent libraries libpcap and libnet, and works with a wide range of operating systems.

Example arping (iputils version) session:

```
ARPING 192.168.39.120 from 192.168.39.1 eth0
Unicast reply from 192.168.39.120 [00:01:80:38:F7:4C] 0.810ms
Unicast reply from 192.168.39.120 [00:01:80:38:F7:4C] 0.607ms
Unicast reply from 192.168.39.120 [00:01:80:38:F7:4C] 0.602ms
Unicast reply from 192.168.39.120 [00:01:80:38:F7:4C] 0.606ms
Sent 4 probes (1 broadcast(s)) Received 4 response(s)
```

IPv6 over Ethernet

- IPv6 packets are encapsulated in Ethernet packets just like IPv4 packets, but with a new Ethertype (*86DD* rather than*0800*).
- IPv6 multicast over Ethernet
  - To send an IPv6 multicast packet over Ethernet, one simply takes the last 32 bits of the destination IPv6 address, prepends *33-33-* and uses that as the destination Ethernet address.
- Neighbour discovery (RFC 2461)
  - Where IPv4 has ARP, IPv6 has NDP, the neighbour discovery protocol. For simple purposes, NDP and ARP are very similar: one node sends out a request packet (called a neighbour solicitation in NDP), and the node it was looking for sends back a reply (neighbour advertisement) giving its link-layer address. ***NDP is part of ICMPv6***, unlike ARP, which doesn't even run over IP. NDP also uses multicast rather than broadcast packets, and that deserves a little more explanation.

References

- http://www.cs.virginia.edu/~itlab/book/
- http://en.wikipedia.org/wiki/Address_Resolution_Protocol
- http://www.tcpipguide.com/free/t_ARPMessageFormat.htm
- http://www.iana.org/assignments/arp-parameters/
- http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/arp.mspx?mfr=true
- http://www-uxsup.csx.cam.ac.uk/courses/ipv6_basics/index.html
- RFC 826 - Ethernet Address Resolution Protocol
- RFC 903 - A Reverse Address Resolution Protocol
- RFC 3927 - Dynamic Configuration of IPv4 Link-Local Addresses
- RFC 2461 - Neighbor Discovery for IP Version 6 (IPv6)

Question?

- What if you ARPing a IP address which is not within your LAN?
  - Can you get a ARP Response? If Yes, from whom?
- What if you fill in the Source Hardware address field of ARP Request with wrong number (should be your own MAC address) ?
  - Can you still get a ARP Response? Why?