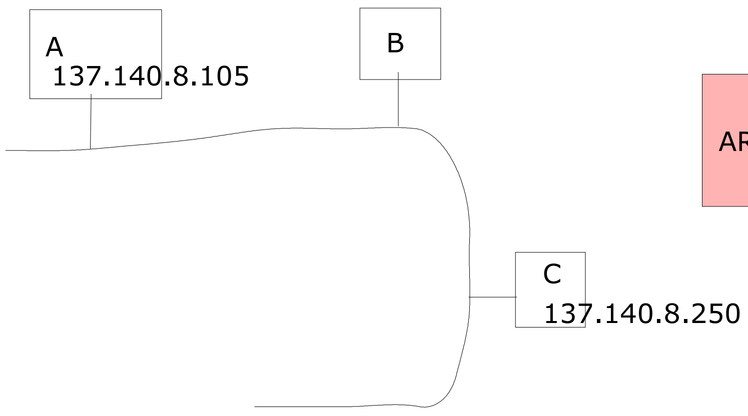
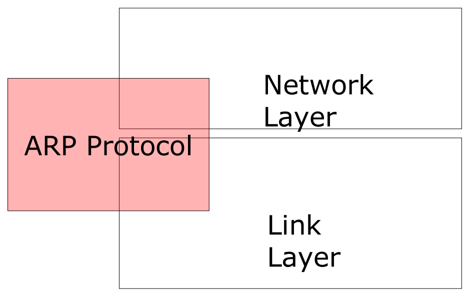
Address Resolution Protocol (ARP)

Each Ethernet frame carries a destination MAC address and a source MAC address. Once IP layer has determined the next-hop destination IP address (which may be a router or the final destination of the datagram- in any case, it is where the datagram will go next), the link layer has to deliver the datagram to the NIC that is hosting that IP address.

To be concrete, consider the following scenario. The network layer on the host 137.140.8.105/24 has decided that a datagram needs to be sent to 137.140.8.250, which is on the same network, directly connected to it. In order for the link layer of A to construct the Ethernet frame, it needs to know the MAC address of the NIC that is bound to the IP address 137.140.8.250. (Remember that MAC addresses are fixed properties of NICs. IP addresses, on the other hand are set at the software level; one or more IP addresses can be associated with a NIC using */sbin/ifconfig* (for linux) or similar commands.

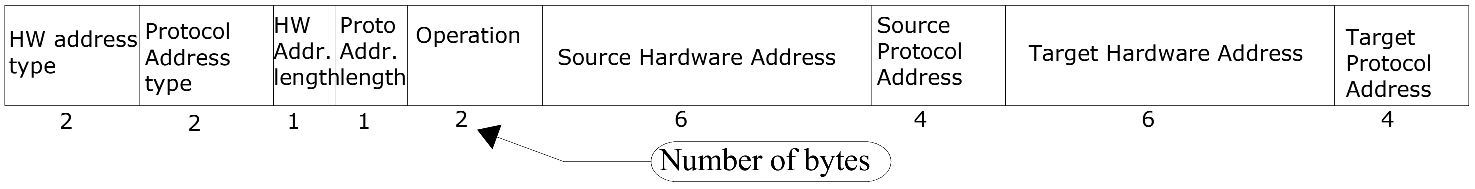
It is the role of the ARP module to resolve this issue - ARP finds out the hardware (MAC) addresses associated with a particular IP address. This is similar to DNS where a mnemonic name is resolved into an IP address at the application level. We can think of ARP as sitting between the Network and Link layers.





In order to find the MAC address of C, A's ARP broadcasts an ARP request, basically asking “who has the IP 137.140.8.250? Tell 137.140.8.105”. All hosts on the network will receive this ARP request broadcast.

The structure of the ARP packet is as shown below:



HW address type	Protocol Address type	HW Addr. length	Proto Addr. length	Operation	Source Hardware Address	Source Protocol Address	Target Hardware Address	Target Protocol Address
2	2	1	1	2	6	4	6	4

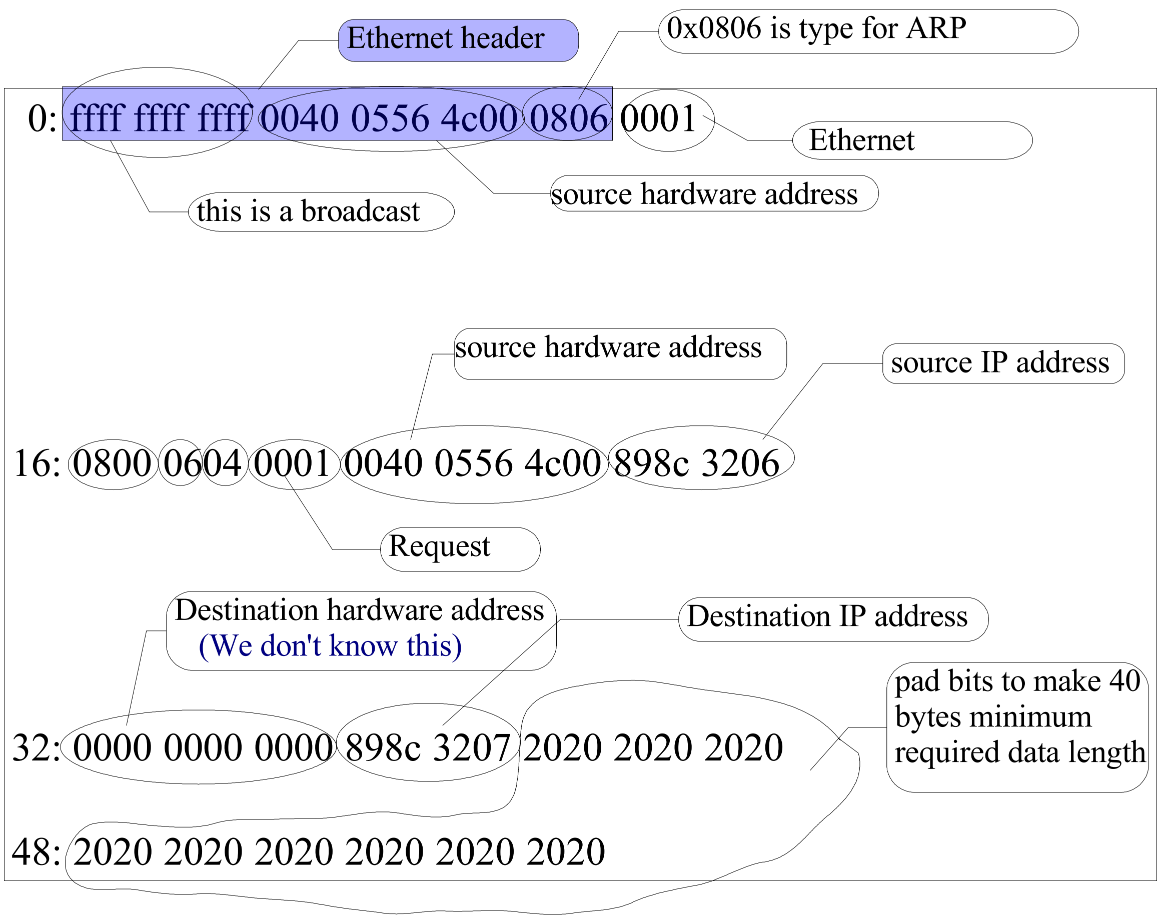
Number of bytes

For a broadcast, the target hardware address will be ff:ff:ff:ff:ff:ff.
For an ARP request the Operation field is 1, and for ARP reply it is 2.
The Protocol Address Type for IPv4 is 0x0800, and Protocol Address Length is 4; HW Address length is 6 for Ethernet. The Hardware Address Type is 1 for Ethernet.

Here's an example ARP request:

```
 0: ffff ffff ffff  0040 0556 4c00 0806 0001
16: 0800 0604 0001 0040 0556 4c00 898c 3206
32: 0000 0000 0000 898c 3207 2020 2020 2020
48: 2020 2020 2020 2020 2020 2020
```

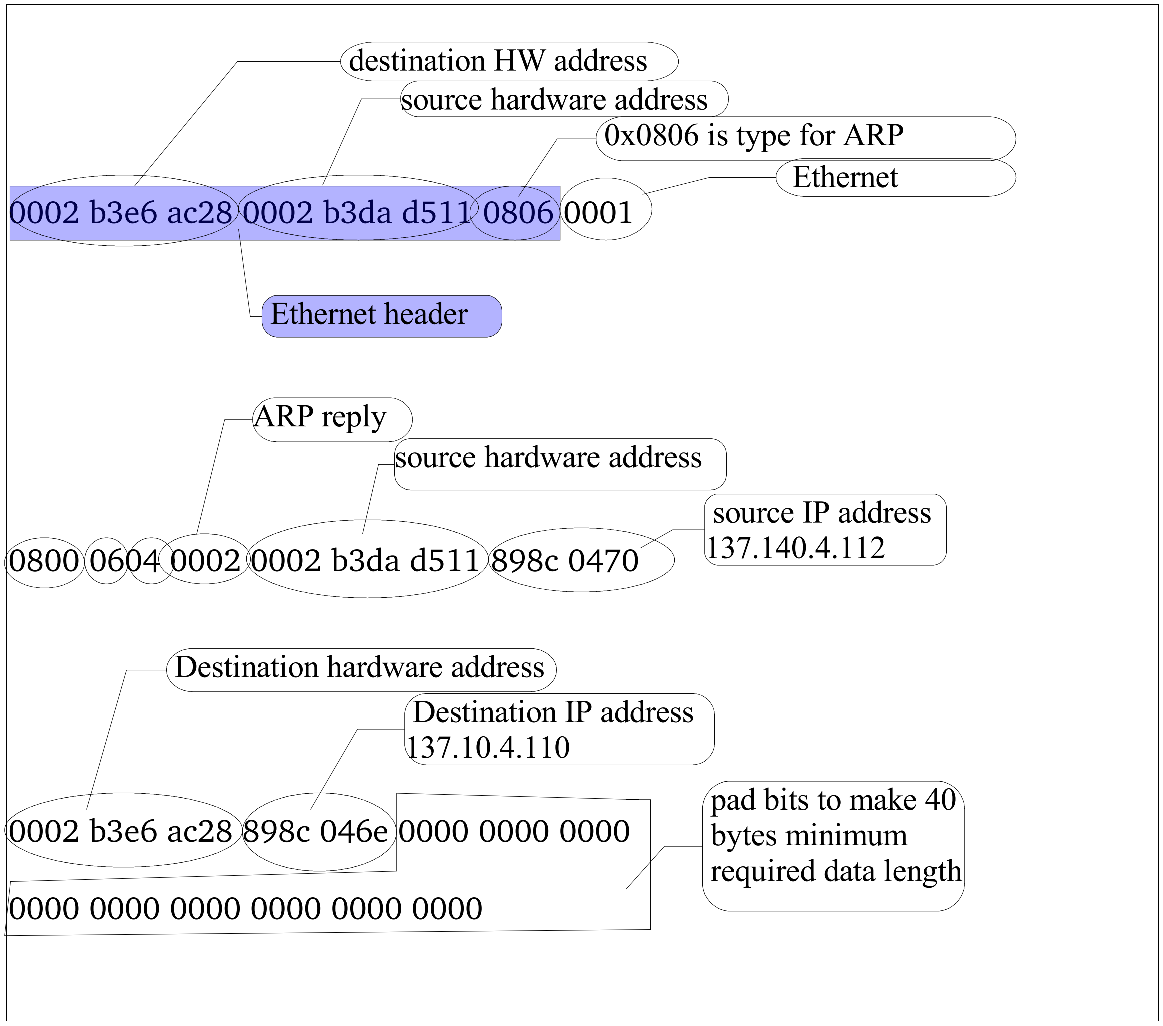
Let us pick this apart and see what it says:


Ethernet header
0x0806 is type for ARP
0: ffff ffff ffff 0040 0556 4c00 0806 0001
Ethernet
source hardware address
this is a broadcast
source hardware address
source IP address
16: 0800 0604 0001 0040 0556 4c00 898c 3206
Request
Destination hardware address
(We don't know this)
Destination IP address
pad bits to make 40 bytes minimum required data length
32: 0000 0000 0000 898c 3207 2020 2020 2020
48: 2020 2020 2020 2020 2020 2020

This is an ARP request from interface with IP address 137.140.50.6 to all hosts on the LAN, asking for the MAC address of the interface that has the IP address 137.140.50.7. Everyone on the LAN will get this broadcast; if 137.140.50.7 is present, the ARP module of that host will issue an ARP reply to the host 137.140.50.6 giving its MAC address. Notice that the ARP reply is not a broadcast, it is a unicast to the requester.

Here is an example of an ARP reply (note:this is NOT the response to the above request):

```
0002 b3e6 ac28 0002 b3da d511 0806 0001
0800 0604 0002 0002 b3da d511 898c 0470
0002 b3e6 ac28 898c 046e 0000 0000 0000
0000 0000 0000 0000 0000 0000
```


destination HW address
source hardware address
0x0806 is type for ARP
Ethernet
0002 b3e6 ac28 0002 b3da d511 0806 0001
Ethernet header
ARP reply
source hardware address
source IP address 137.140.4.112
0800 0604 0002 0002 b3da d511 898c 0470
Destination hardware address
Destination IP address 137.10.4.110
0002 b3e6 ac28 898c 046e 0000 0000 0000
0000 0000 0000 0000 0000 0000
pad bits to make 40 bytes minimum required data length

This is a reply from 137.140.4.112 to 137.140.4.110 saying that "137.140.4.112 is at 00:02:b3:da:d5:11" . This must be in reply to an ARP request broadcast from 137.140.4.110 asking "who has 137.140.4.112, tell 137.140.4.110".

Exercise: Reconstruct the ARP request that produced the above reply.

Once 137.140.4.110 receives this reply, it can construct an Ethernet frame with data to be sent to 137.140.4.112. Usually, the receiver of the reply, in this case 137.140.4.110, caches this ARP information for future use. On most unix machines, executing " */sbin/arp -a* " gives the ARP cache.

Exercise: Decode the following ARP request/reply messages:

```
 1.   0x0000:  ffff    ffff    ffff   0002 b3da d511 0806 0001
      0x0010:  0800 0604 0001 0002 b3da d511 898c 0470
      0x0020:  0000 0000 0000 898c 046e 0000 0000 0000
      0x0030:  0000 0000 0000 0000 0000 0000


 2.   0x0000:  0002 b3da d511 0002 b3e6 ac28 0806 0001
      0x0010:  0800 0604 0002 0002 b3e6 ac28 898c 046e
      0x0020:  0002 b3da d511 898c 0470
```

(Note: the hex values are byte counts).