Binding IP Addresses To Physical Hardware Addresses

- Must use physical addresses to send and receive packets
- Therefore, must map IP addresses to physical addresses
- Particular method depends on underlying hardware technology
- Software to perform mapping built into network interface software in the operating system

Static Address Binding

- Useful when hardware addresses chosen from small, dense set
- Key ideas:
  - Local site administrator is free to choose host portion of an IP address
  - Choose IP address that encodes hardware address in its host portion
  - Mapping becomes trivial

Example Of Static Binding

- Network portion of Internet address fixed when Internet address assigned
- Use machine names as part of the address
- Choose Internet address with low-order byte equal to machine name
- To map destination Internet address to hardware address, extract the low-order byte
- Example:
  - bin00001.cs.ohiou.edu is 132.235.2.201
  - bin01000.cs.ohiou.edu is 132.235.2.208

Dynamic Address Binding

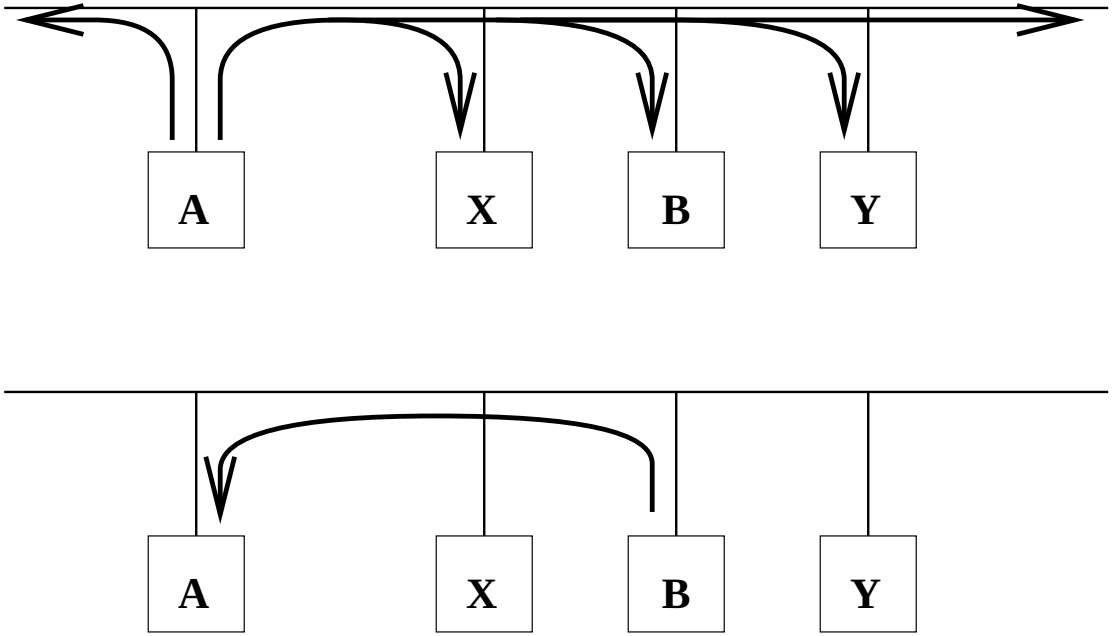
- Useful when hardware addresses are large and fixed
- Requires hardware broadcast
- Best for local area network
- Key ideas:
  - Maintain a table of pairs (A_I,A_H) where A_I is Internet address and A_H is hardware address for machine A
  - Use the network to obtain new bindings

Internet Address Resolution Protocol (ARP)

- Internet standard for dynamic address binding
- Allows machine A to find machine B's physical address knowing only B's Internet address
- Uses hardware broadcast
- Note: ARP only used to map addresses within a single physical network, never across multiple networks
- Machine A broadcasts ARP request with B's IP address
- All machines on local net receive broadcast
- Machine B replies with its physical address
- Machine A adds B's address information to its table
- Machine A delivers packet directly to B

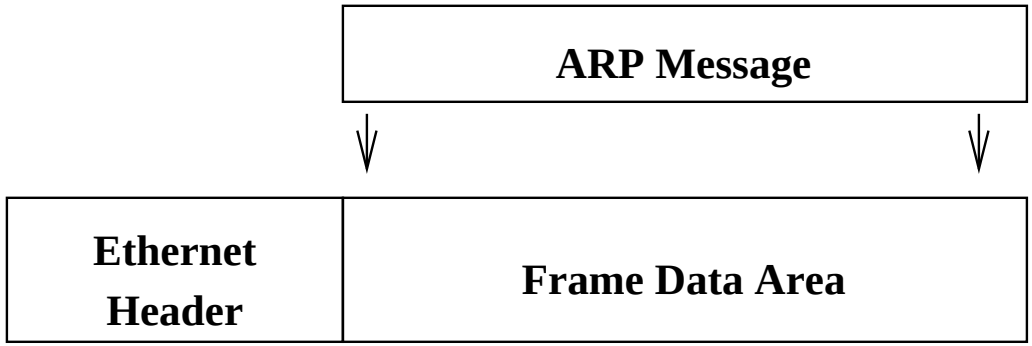
Illustration Of ARP Request And Reply Messages

- Example of machine "A" looking for machine "B"
  - Machine "A" sends a broadcast that goes to every machine on the net
  - Machine "B" responds back to machine "A" with its hardware address



ARP Encapsulation

- The ARP request/response rides as data on top of the underlying physical network frame



ARP Packet Format When Used With Ethernet

- The general format

0	8	16	31
Hardware-Type		Protocol-Type	
HLEN	PLEN	Operation	
Sender HA (octets 0-3)			
Sender HA(octets 4-5)		Sender IA (octets 0-1)	
Sender IA (octets 2-3)		Target HA (octets 0-1)	
Target HA (octets 2-5)			
Target IA (octets 0-3)			

- Ethernet example

0	8	16	31
(Ethernet)		(IP)	
6	4	(request)	
8:0:2b:1b			
:b7:2c		132.235	
.1.2		xx:xx	
:xx:xx:xx:xx			
132.235.1.1			

Algorithm For Processing ARP Requests

- Extract sender's pair, (A_I,A_H) and update local ARP table
- If this is a request and the target is "me"
  - Fill in target hardware address
  - Exchange sender and target entries
  - Set operation to *reply*
  - Send reply back to requester

ARP Details

- ARP table is merely a cache
  - Entries should time out and be invalidated
- Machine can broadcast new binding when it boots
- Broadcasts are very expensize
  - Every machine on the network sees the request
  - Doesn't scale well to large networks
- Problems
  - What do you do if the machine doesn't answer?
  - What do you if you're trying to send "many" packets to a machine for which you have no address?

ARP's Unix Interface

- The Unix command "`arp -a`" on most Unix hosts will list the current arp table:

```
KSH:p1> arp -a
Net to Media Table: IPv4
Device   IP Address               Mask      Flags   Phys Addr
------ -------------------- --------------- ----- ---------------
ce0    prime                255.255.255.255       08:00:20:bc:6b:38
ce0    boss.cs.ohiou.edu    255.255.255.255       08:00:20:9e:f4:f5
ce0    gateway1.cs.ohiou.edu 255.255.255.255       00:00:0c:07:ac:2b
ce0    132.235.14.76        255.255.255.255       00:07:e9:84:cc:fe
ce0    p1                   255.255.255.255 SP    00:03:ba:69:00:c2
ce0    132.235.1.235        255.255.255.255 U
```

- The Unix command "`arp host`" on most Unix hosts will give the physical address of machine "`host`", if known

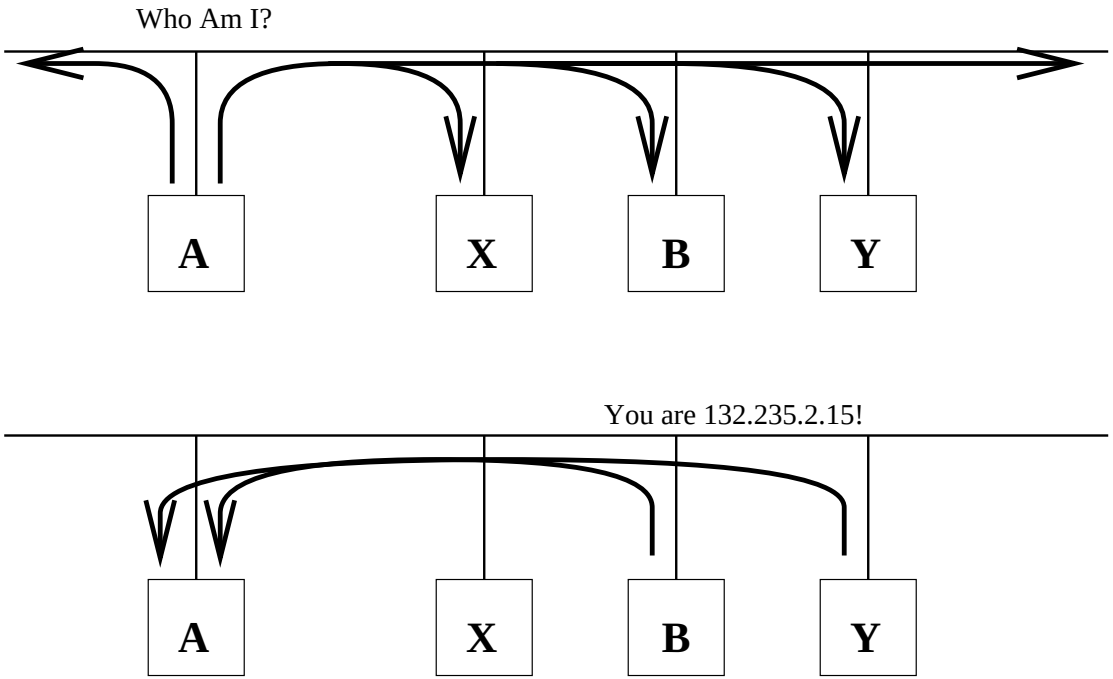
```
KSH:p1> arp bin00001
bin00001 (132.235.2.201) at 0:0:c:7:ac:2b
KSH:p1> arp bin00010
bin00001 (132.235.2.202) -- no entry
KSH:p1> arp bin11111
arp: bin11111: unknown host
```

Reverse Address Resolution Protocol RARP

- We have already seen that establishing a binding between a virtual (IP) address and a physical (e.g. ethernet) address is useful
  - Locating the physical addresses of "neighbor" machines
- Is it useful to provide a way to do the binding in the other direction?
  - Given a physical address, what's the virtual address
- RARP is generally used by diskless machines
  - The can read their physical address from their (ethernet) interface
  - Before the can become network citizens, they need their IP address

RARP Details

- A machine knows it's physical address and wants to find out its virtual (IP) address
- Sends RARP broadcast on local network
- Neighbor machines that know its IP address respond directly to that machine



RARP Details (continued)

- Like ARP, RARP also travels in a physical frame
  - RARP even uses the same packet format at ARP, but has a different physical type to distinguish it from ARP
    - ARP is ethernet type 0x0806
    - RARP is ethernet type 0x8035
- Like ARP, the RARP protocol is unreliable
  - A machine may have to ask several times before it gets a response
- Unlike ARP, a machine cannot generally continue if it can't get a response to its RARP message
- Unix machines that answer RARP messages use a server called "`rarpd`" reading from a file called "`/etc/ethers`"