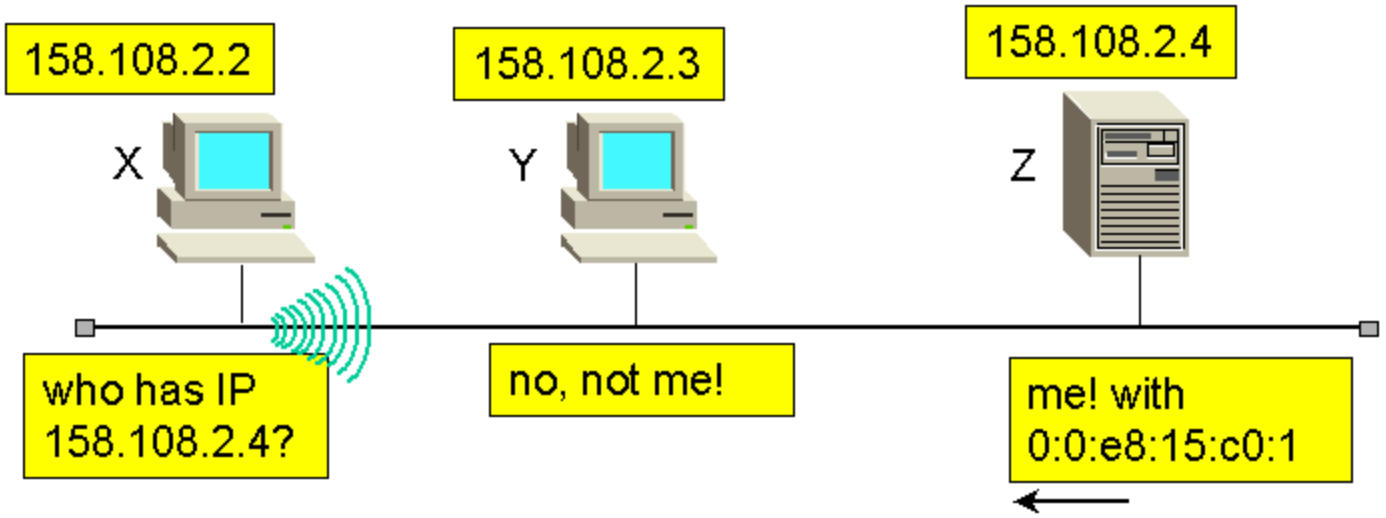
# Address Resolution Protocol - ARP

IP over Ethernet



# Introduction

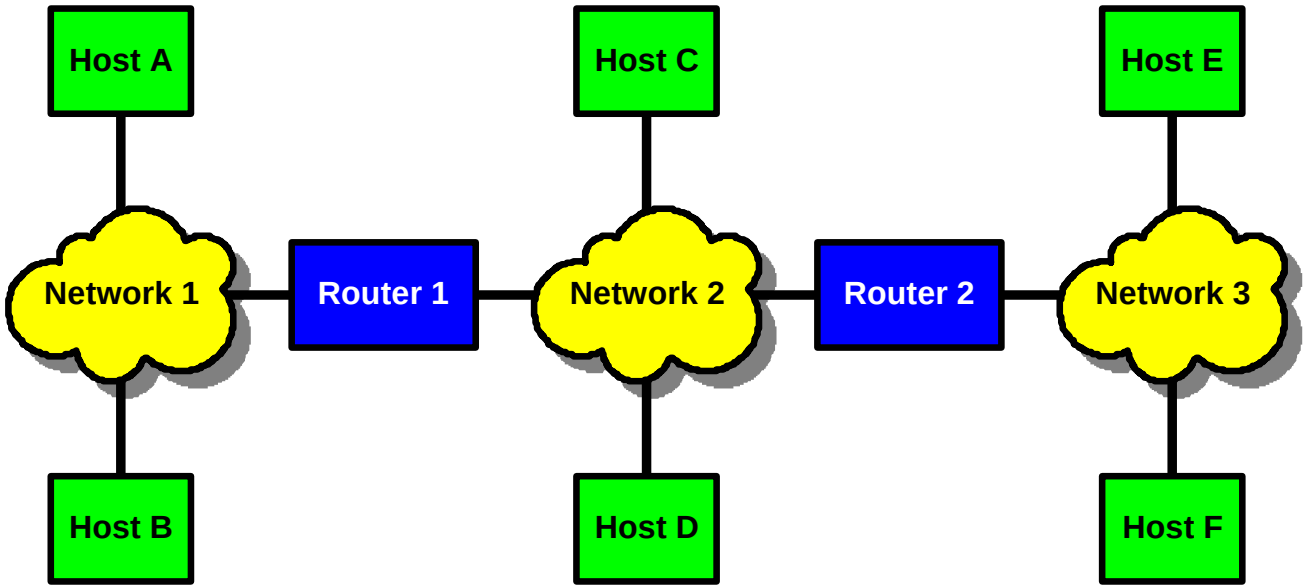
- Upper levels of protocol stack (TCP/IP, IPX/SPX, DECNet, etc.) use protocol addresses
- Network hardware must use hardware/physical/link-level address for eventual delivery
- Protocol address must be translated into hardware address for delivery

# Address Resolution

- Finding hardware address for protocol address is called Address Resolution
- Data link layer resolves protocol address to hardware address
- Resolution is local to a network
- Network component only resolves address for other components on same network

# Address Resolution (continued)

- A resolves protocol address for B for protocol messages from an application on A sent to an application on B
- A does not resolve a protocol address for F
- Through the internet layer, A delivers to F by routing through R1 and R2
- A resolves R1 hardware address
- Network layer on A passes packet containing destination protocol address F for delivery to R1



# Address Resolution Techniques

- Association between a protocol address and a hardware address is called a binding. Three techniques:
  - **Table lookup** - Bindings stored in memory with protocol address as key - data link layer looks up protocol address to find hardware address
  - **Closed-form computation** - Protocol address based on hardware address - Data link layer derives hardware address from protocol address
  - **Dynamic** - Network messages used for "just-in-time" resolution - Data link layer sends message requesting hardware address; destination responds with its hardware address

# Closed-form Computation

- If hardware technology uses small, configurable hardware address, network administrator can choose hardware address based on IP address
- Example - hardware uses one octet address that can be configured
- Simply choose hardware address to be hostid
- Now, any host can determine hardware address as:
- hardware_address = ip_address & 0xff

# Dynamic Resolution

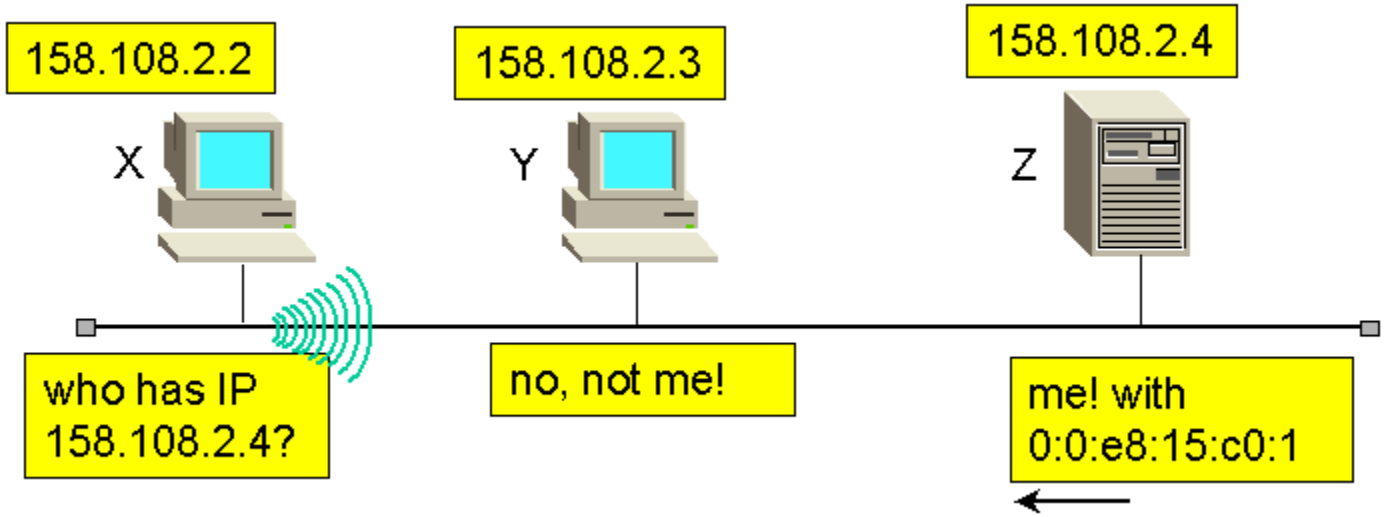
- Use the network to resolve IP addresses to hardware addresses
- Message exchange with other computer(s) returns hardware address to source
- Two designs:
  - Server-based - computer sends message to a server to resolve the address. Every computer would need:
    - List of servers OR
    - Broadcast to locate servers
  - Distributed - all computers participate; destination provides hardware address to host

# Address Resolution Protocol - ARP

- IP uses dynamic distributed resolution technique
- Address Resolution Protocol (ARP) - part of TCP/IP protocol suite
- RFC 826 - Address Resolution Protocol
- Two-part protocol:
  - Request from source asking for hardware address
  - Reply from destination carrying hardware address

# ARP Message Exchange

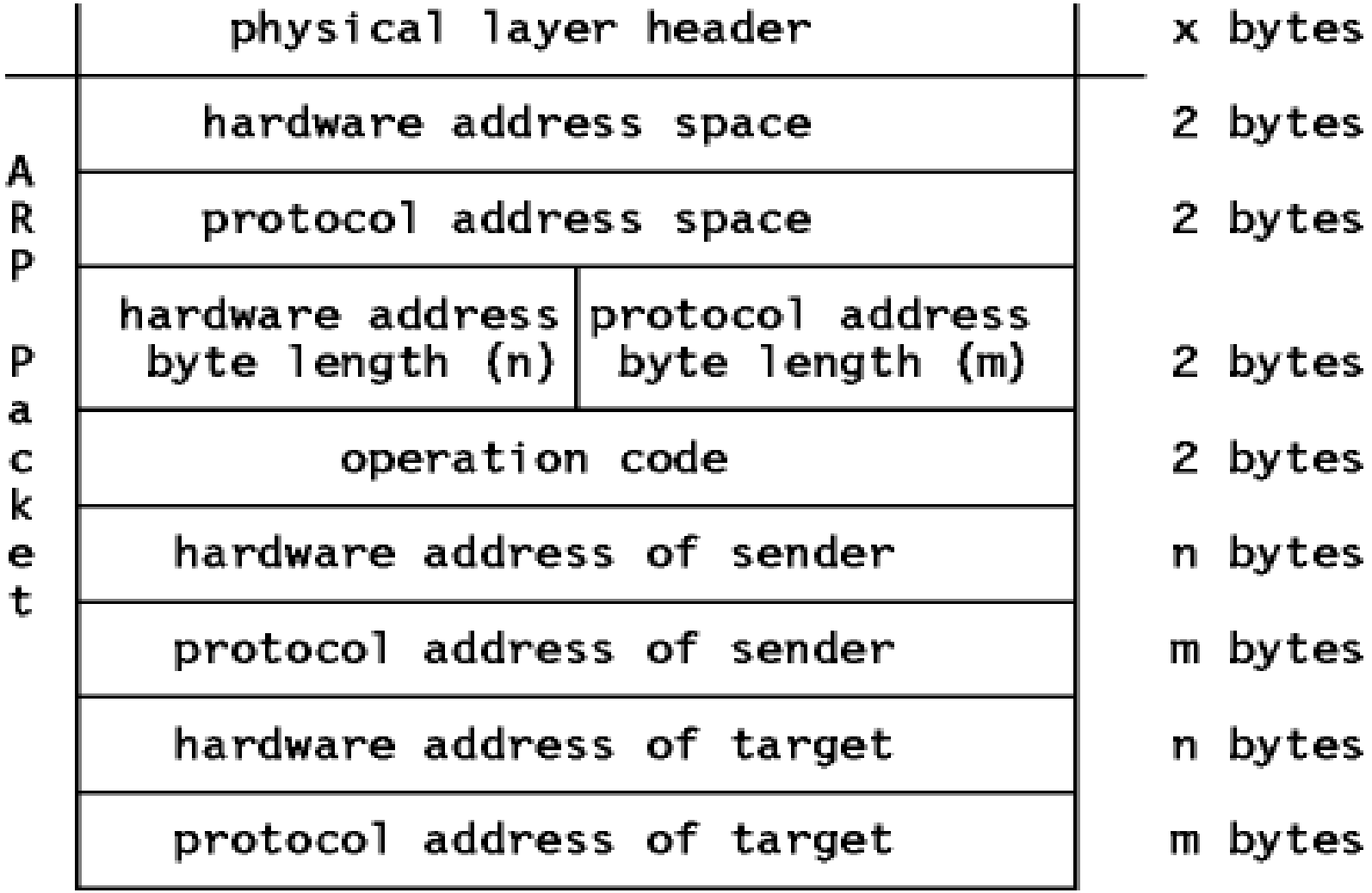
- ARP request message dropped into a hardware frame and broadcast
- Sender inserts IP address into message and broadcast
- Every other computer examines request



# ARP Message Exchange (cont'd)

- Computer whose IP address is in the request responds
- Puts its own hardware address in the response
- Unicasts the response to the sender
- Original requester can then extract hardware address and send IP packet to destination using recently acquired hardware address

# ARP Message Format

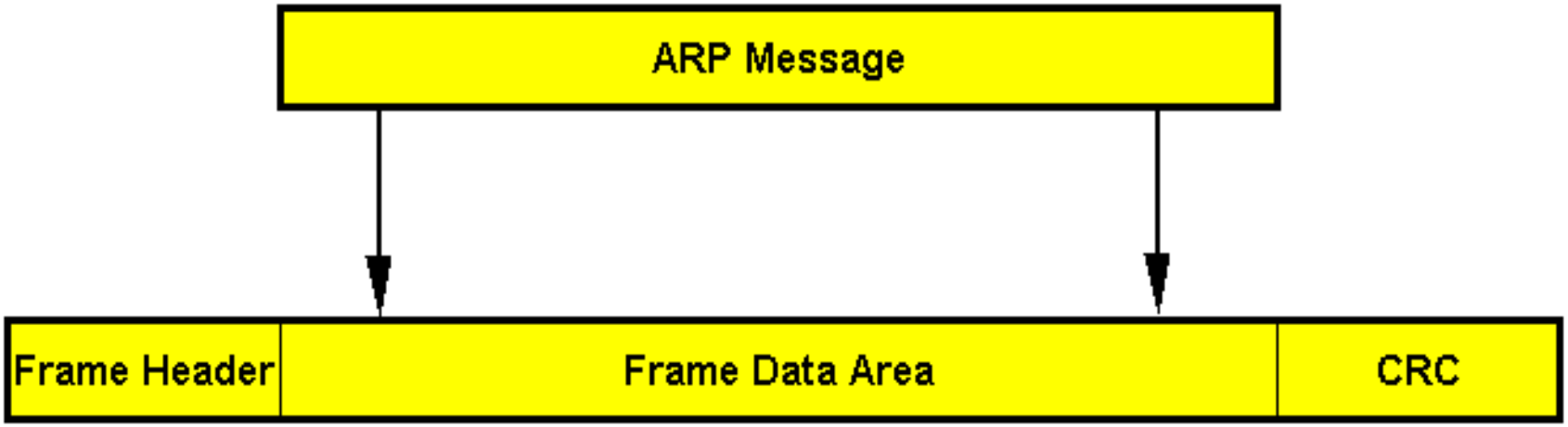


# ARP Message Contents

- HARDWARE ADDRESS TYPE = 1 for Ethernet
- PROTOCOL ADDRESS TYPE = 0x0800 for IP
- OPERATION = 1 for request, 2 for response
- Contains both target and sender mappings from protocol address to hardware address
- Request sets hardware address of target to 0
- Target can extract hardware address of sender (saving an ARP request)
- Target exchanges sender/target in response

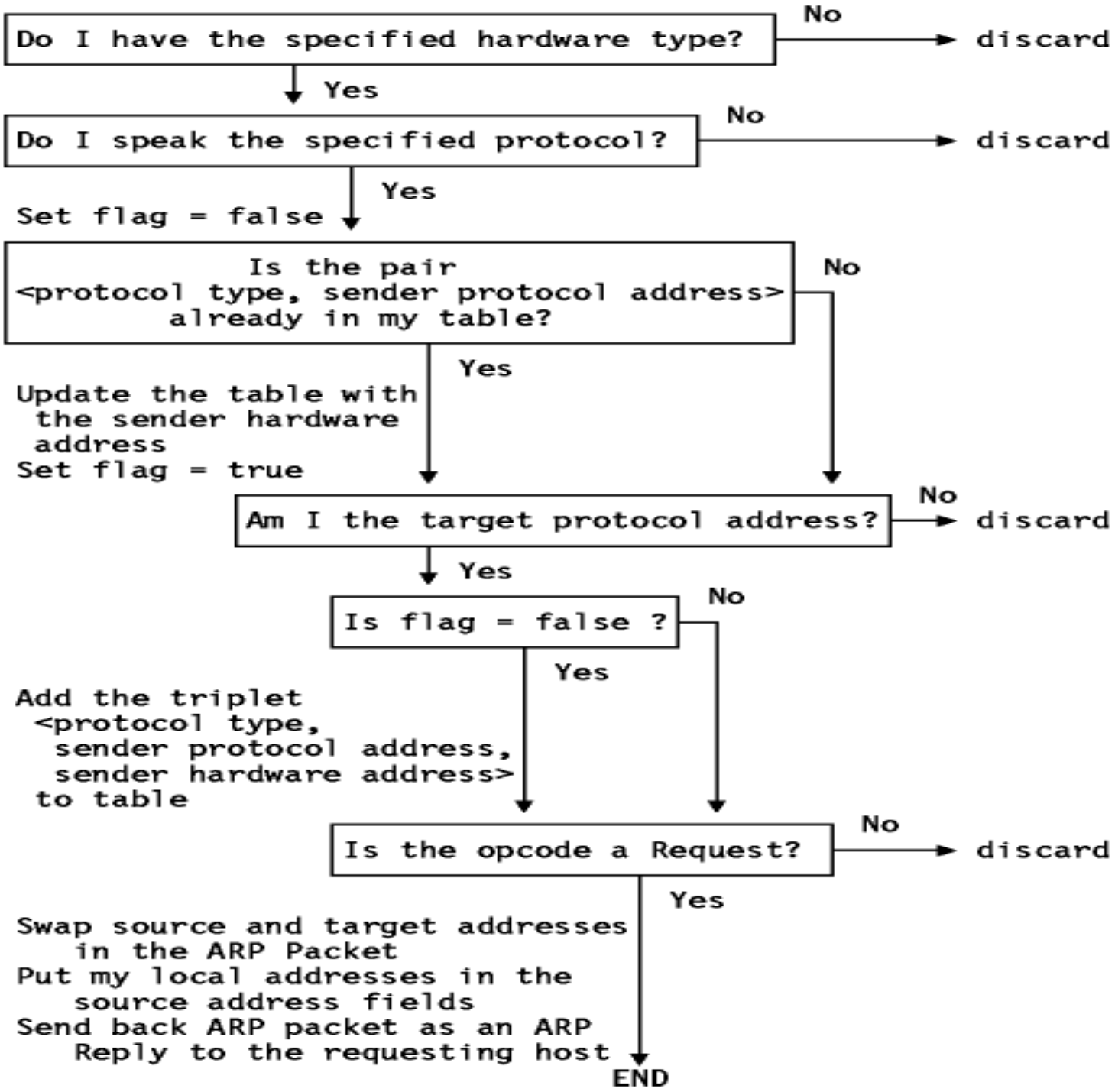
# Sending an ARP Message

- Sender constructs ARP message
- ARP message carried as data in hardware frame - encapsulation



# Processing the ARP Messages

- Receiver extracts sender's hardware address and updates local ARP table
- Receiver checks operation - request or response
  - Response: Adds sender's address to local cache
    - Sends pending IP packet(s)
  - Request: If receiver is target, forms response
    - Unicasts to sender
- Adds sender's address to local cache
- Note:
  - Target likely to respond "soon"
  - Computers have finite storage for ARP cache
  - Only target adds sender to cache; others only update if target already in cache


Do I have the specified hardware type?
No
discard
Yes
Do I speak the specified protocol?
No
discard
Yes
Set flag = false
Is the pair
<protocol type, sender protocol address>
already in my table?
No
Yes
Update the table with
the sender hardware
address
Set flag = true
Am I the target protocol address?
No
discard
Yes
Is flag = false ?
No
Yes
Add the triplet
<protocol type,
sender protocol address,
sender hardware address>
to table
Is the opcode a Request?
No
discard
Yes
Swap source and target addresses
in the ARP Packet
Put my local addresses in the
source address fields
Send back ARP packet as an ARP
Reply to the requesting host
END

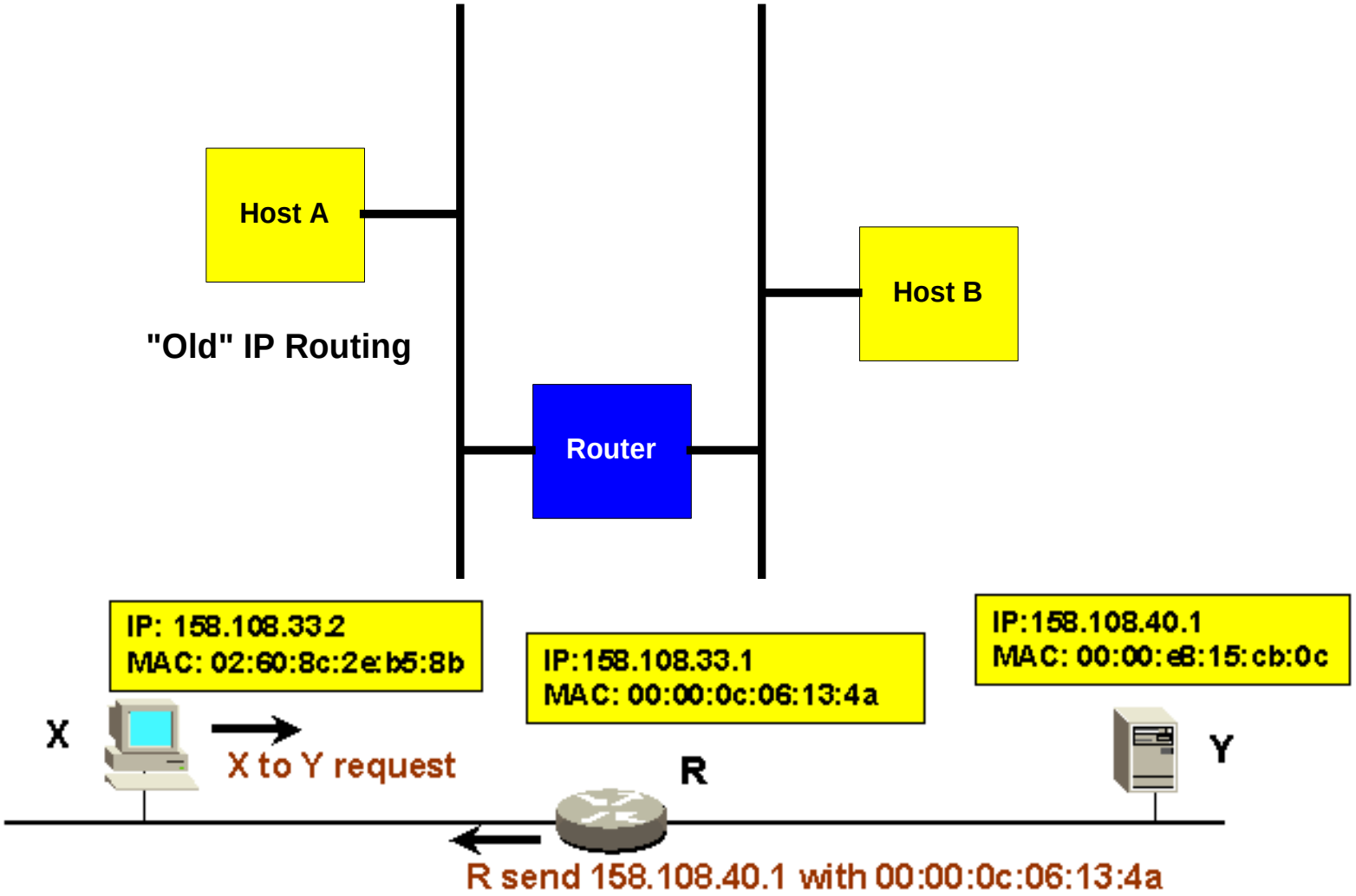
# ARP, Bridging and Routing

- ARP is transparent to bridging, since bridging will propagate ARP broadcasts like any other Ethernet broadcast, and will transparently bridge the replies.
- A router does not propagate Ethernet broadcasts, because the router is a Network Level device, and Ethernet is a Data Link Level protocol. Therefore, an Internet host must use its routing protocols to select an appropriate router, that can be reached via Ethernet ARPs.
- After ARPing for the IP address of the router, the packet (targeted at some other Destination Address) is transmitted to the Ethernet address of the router.

# Proxy ARP

- Proxy ARP is a technique that is can be used by routers to handle traffic between hosts that don't expect to use a router as described above. Probably the most common case of its use would be the gradual subnetting of a larger network. Those hosts not yet converted to the new system would expect to transmit directly to hosts now placed behind a router.
- A router using Proxy ARP recognizes ARP requests for hosts on the "other side" of the router that can't reply for themselves. The router answers for those addresses with an ARP reply matching the remote IP address with the router's Ethernet address (in essence, a lie).

# Proxy ARP Use



# Proxy ARP - Problems

- Proxy ARP is best thought of as a temporary transition mechanism, and its use should not be encouraged as part of a stable solution. There are a number of potential problems with its use, including the inability of hosts to fall back on alternate routers if a network component fails, and the possibility of race conditions and bizarre traffic patterns if the bridged and routed network segments are not clearly delineated.

# Proxy ARP Use

- When host A wants to send an IP datagram to host B, it first has to determine the physical network address of host B through the use of the ARP protocol.
- As host A cannot differentiate between the physical networks, his IP routing algorithm thinks that host B is on the local physical network and sends out a broadcast ARP request. Host B doesn't receive this broadcast, but router R does. Router R understands subnets, that is, it runs the ``subnet'' version of the IP routing algorithm and it will be able to see that the destination of the ARP request (from the target protocol address field) is on another physical network. If router R's routing tables specify that the next hop to that other network is through a different physical device, it will reply to the ARP as if it were host B, saying that the network address of host B is that of the router R itself.

# Proxy ARP Use

- Host A receives this ARP reply, puts it in his cache and will send future IP packets for host B to the router R. The router will forward such packets to the correct subnet.
- The result is transparent subnetting. Normal hosts (such as A and B) don't know about subnetting, so they use the "old" IP routing algorithm.
- The routers between subnets have to:
  - Use the "subnet" IP algorithm.
  - Use a modified ARP module, which can reply on behalf of other hosts.

# Reverse ARP - RARP

- Sometimes, it is also necessary to find out the IP-address associated with a given Ethernet address. This happens when a diskless machine wants to boot from a server on the network, which is quite a common situation on local area networks.
- A diskless client, however, has virtually no information about itself-- except for its Ethernet address! So what it basically does is broadcast a message containing a plea for boot servers to tell it its IP-address.
- There's another protocol for this, named Reverse Address Resolution Protocol, or RARP. Along with the BOOTP protocol, it serves to define a procedure for bootstrapping diskless clients over the network.

# Reverse ARP - RARP

- Reverse ARP, document in RFC 903, is a fairly simple bootstrapping protocol that allows a workstation to broadcast using its Ethernet address, and expect a server to reply, telling it its IP address.

# Inverse ARP

- Used on point to point links
- Find IP address of the host on the other end
- Used in frame relay and ATM
- Uses codes 8 (request) and 9 (response)
- Ref: RFC 1293

# Summary

- ARP allows converting IP address to MAC addresses
- Proxy ARP, RARP, Inverse ARP