IP/MAC Address Translation

Today

-Go over quiz answers

-ARP

-DHCP

-NAT

Transition from Network to Datalink

- How do we get datagrams to the right physical host?
  - Tricky part comes when a router is forwarding to a LAN with multiple hosts (which is typically the case)
- IP datagrams contain an IP address
  - Configured in OS
- NIC's only understand addressing of their particular network
  - Ethernet's 48 bit MAC addresses

Address Translation Problem

- We need a means for mapping IP addresses into MAC (physical) addresses
  - Destination host
  - Next hop router
  - We can then encapsulate (surpirse!) IP datagrams inside a *frame* with link level address
- Possible mapping techniques
  - Encode physical address in host part of IP address
    - Make physical address the same as the host portion of IP address
    - Obviously not possible using IPv4 and Ethernet
  - Build a table of IP/MAC pairs
    - How is it maintained?

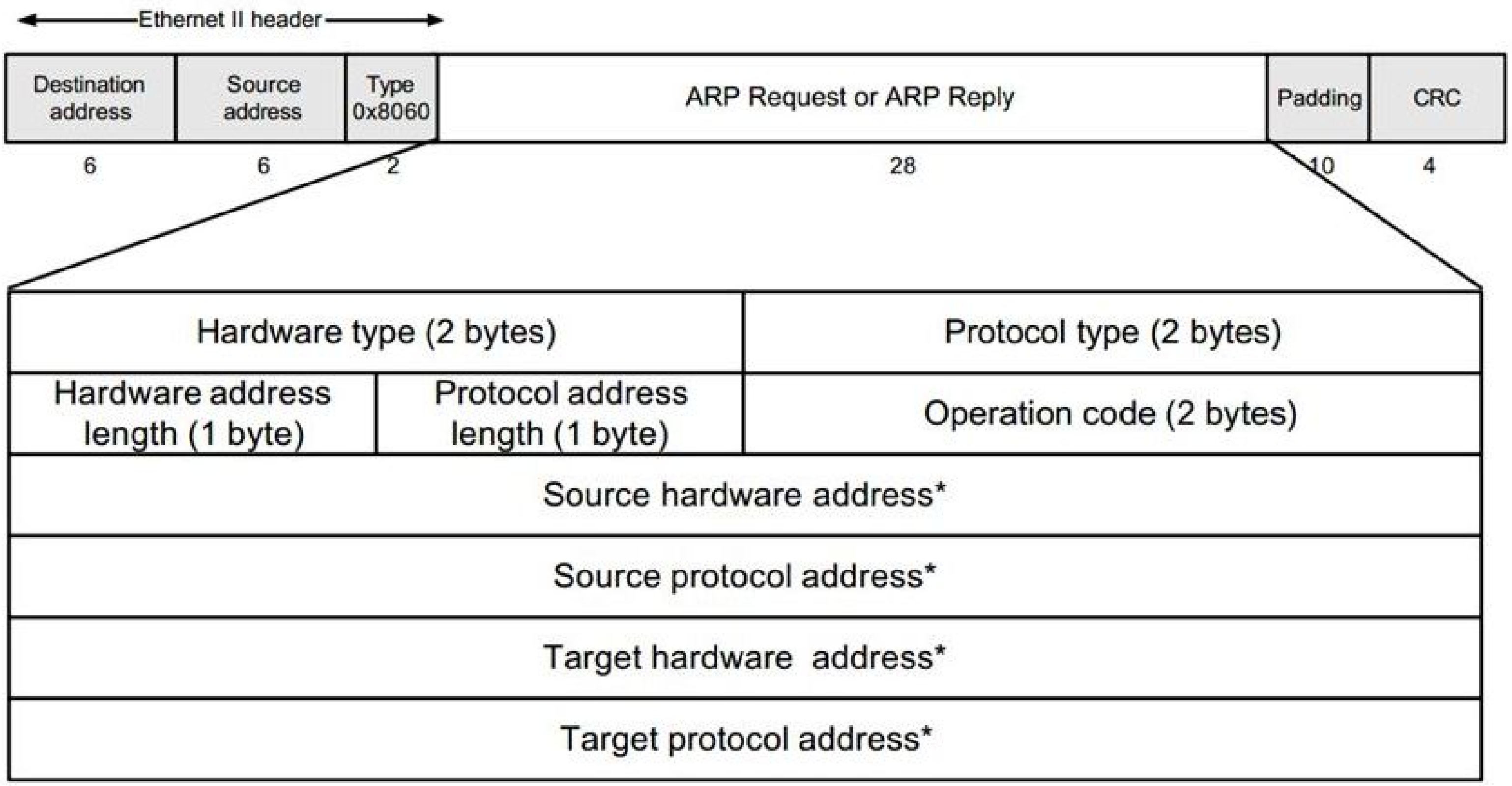
Address Resolution Protocol (ARP)

- ARP is part of the TCP/IP specification
- Enable each host to build table of IP to physical address bindings
  - Dynamic binding protocol – no static entries in table
  - Allows new nodes to be easily added to broadcast network
- Simple idea: broadcast request if an IP address not in table
  - Supported by link level technology
  - Determine host B's physical address P_B from it IP address I_B
    1. Host A broadcasts an ARP request containing I_B to all hosts on LAN
    2. Host B responds with an ARP reply containing the pair (I_B, P_B)

ARP Implementation

- ARP Packet Details
  - HardwareType: type of physical network (e.g., Ethernet)
  - ProtocolType: type of higher layer protocol (e.g., IP)
  - HLEN & PLEN: length of physical and protocol addresses
    - Provides for flexibility to handle a variety of network technologies
  - Operation: request or response
  - Source/Target-Physical/Protocol addresses
- Notes
  - Table entries timeout in about 10 minutes (caching is important)
  - Update table with source when you are the target
  - Update table even if there is already an entry
  - Do not refresh table entries upon reference
  - IP addresses are assigned independently of a systems HW addresses

ARP Packet Format



http://slideplayer.com/slide/6343460/22/images/8/ARP+%E2%80%93+Packet+structure+The+length+of+the+address+fields+is+determined+by+the+corresponding+address+length+fields..jpg

Determining an IP Address at Startup

- How does a machine without permanent storage determine its IP address?
  - OS images with specific IP's cannot be used on multiple machines
  - Critical for network appliances or embedded systems
- Use the network to obtain an IP from a remote server
  - System must use its physical address to to communicate
  - Requests address from server which maintains table of IP's
  - System doesn't know the server - sends broadcast request for address

Dynamic Configuration

- BOOTP was designed for relatively static environment where each host has a permanent network connection
  - Net manager creates a BOOTP config file with parameters for each host – file is typically stable for long periods
- Wireless networking enables environments much more dynamic
  - BOOTP does not provide for dynamic address assignment
- Dynamic configuration is the primary method for IP address allocation used today
  - Not only facilitates mobility but also efficient use of IPs

Dynamic Host Configuration Protocol

- DHCP extends BOOTP
  - Still supports static allocation
  - Supports automatic configuration where addresses are permanent but assigned by DHCP
  - Supports temporary allocation
- Relies on existence of a DHCP server
  - Repository for host configuration information
  - Maintains a pool of available IP's for use on demand
  - Considerably reduces administration overhead
    - Autoconfiguration of course depends on administrative policy
  - Uses UDP to send messages
    - Uses a *relay agent* to communicate with servers off LAN (same as BOOTP)
      - Relay agent is statically configured with DHCP server address

DHCP Implementation

- State machine (6 states) which determines DHCP operation
  - Host boots into *INITIALIZE* state
- To contact the DHCP server(s) a client sends DHCPDISCOVER message to IP broadcast address and moves to *SELECT* state
  - Unique header format with variable length options field
  - UDP packet sent to well known BOOTP port 67
- Server(s) respond with DHCPOFFER message
  - Client can receive 0 or more responses and responds to one
- Client moves to *REQUEST* state to negotiate IP lease with 1 server
  - Sends DHCPREQUEST message to server which responds with DHCPACK
- Client is then in *BOUND* (normal) state

DHCP Implementation contd.

- From *BOUND*, client can issue DHCPRELEASE and return to *INITIALIZE* state
  - This is simply client deciding it no longer needs the IP
- When lease reaches 50% of lease expiration time, it issues DHCPREQUEST to extend lease of current IP with server and moves to *RENEW* state
  - Receipt of DHCPACK moves client back to *BOUND* state
  - Receipt of DHCPNACK moves client back to *INITIALIZE* state
- If no response is received by 87.5% of lease expiration time, the client resends the DHCPREQUEST and moves to *REBIND* state
  - Receipt of DHCPACK moves client back to *BOUND* state
  - Receipt of DHCPNACK or timeout moves client back to *INITIALIZE* state

DHCP Details

- Without relay agent, DHCP would not scale since it would require large number of servers (one per LAN)
- Addresses which are leased over a given period of time and must be updated
  - This means that DHCP requests might have to be made multiple times by the same system (RENEW requests)
- DHCP does not interact with DNS
  - Binding between IP assigned by DHCP and host name must be made independently
    - Possible result 1: No host name given
    - Possible result 2: Host is automatically assigned a preallocated domain name with its IP
    - Possible results 3: Hosts are assigned permanent names
      - Requires additional mechanisms which do not yet exist

NAT - Network Address Translation

- Little over 4 Billion IP Addresses
  - Not enough IP Addresses to go around!
- How do you manage IP Addresses?
  - What if you’re only assigned 1?
- DHCP doesn’t know all IPs on the internet
- How do you deal with duplicates?
  - ie.) 192.168.1.100

NAT - Basic Idea

- Use one IP address for multiple devices
- Devices IP addresses are only seen in local network
- External network (ie Internet) sees one IP for all the devices
- Solves “too many devices” problem
- Allows DHCP to work without coordination
- Easy to manage

NAT - Some Details

- NAT Gateway translates internal addresses to external ones
  - External IP Address + port => Internal Address + Port
- Keeps *NAT Translation Table*
  - Cache of mappings
- Uses cache to forward incoming packets
- Saves new entries for outgoing packets

NAT - Problems

- Port numbers for processes not NAT
- Slows adoption of IPv6 (longer IP Addresses)
- Interferes with P2P
  - How to connect from outside?
  - No NAT Translation Table entry

NAT Traversal

- Establishing a P2P connection through NAT
- Easiest Way: use an intermediary with Public IP Address
  - Game servers, etc.
- UPnP (Universal Plug and Play)
  - Add ports to NAT Translation Table manually, for specific applications