Pieces of the Puzzle

EE450: Introduction to Computer Networks

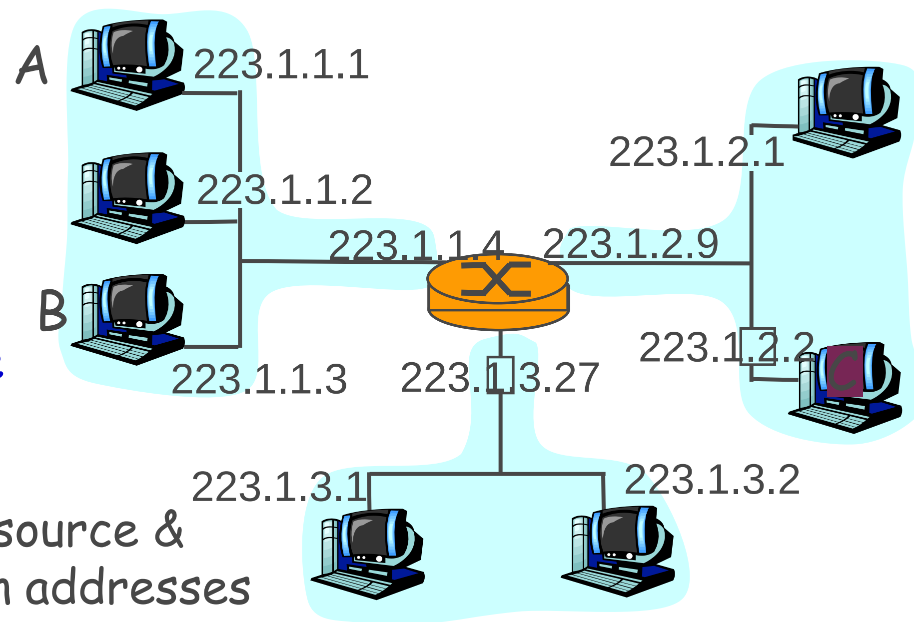
Professor A. Zahid

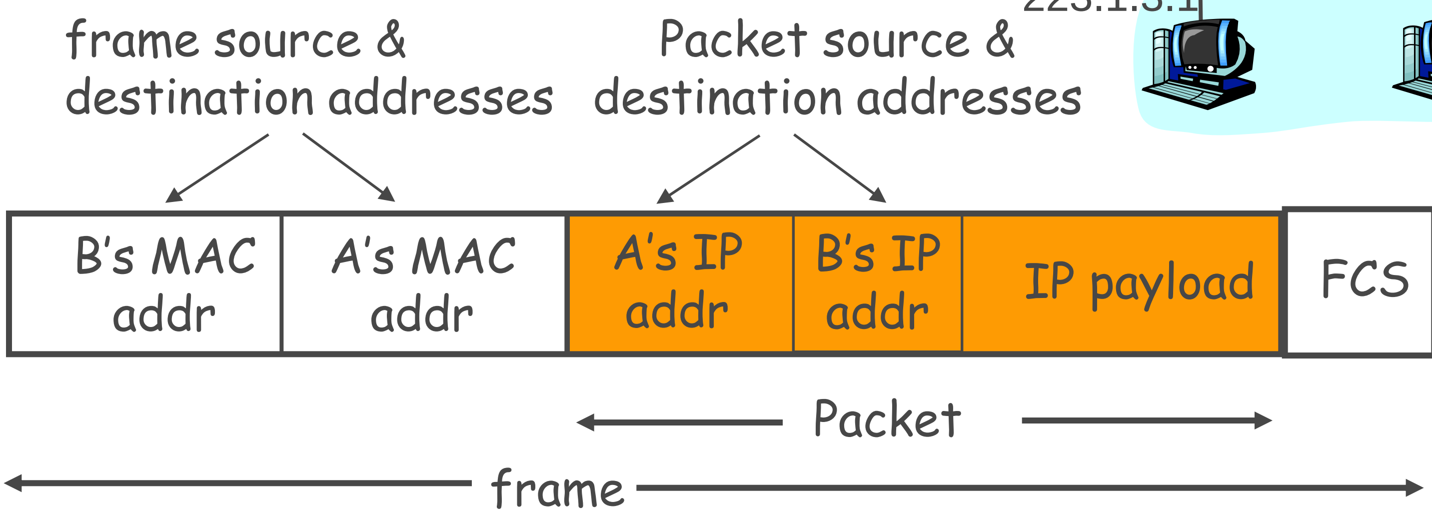
Burning Questions

- How does a host/router get the MAC address of another host/router on the same LAN?
  - Answer: Address Resolution Protocol: ARP
- How does a host get the IP address of another host across the Internet?
  - Answer : Domain Name System: DNS
- How does a host get it's own IP address?
  - Answer: Dynamic Host Configuration Protocol (DHCP)
- How do we distinguish between two or more applications running on the same host?
  - Answer: Port Numbers/Sockets

IP & MAC Addresses

- *A has a Packet to send to B.*
- *A has the IP address of B and it knows that B is located on the same network.*
- *A encapsulate the packet inside a link-layer frame*





IP & MAC Addresses (Cont.)

- MAC address allocation administered by IEEE
- Manufacturer buys portion of MAC address space (to assure uniqueness)
- analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address
- MAC is a flat address ➜ portability
  - can move LAN card from one LAN to another
- IP is a hierarchical address ➜ not portable
  - address depends on IP subnet to which node is attached

IP & MAC Addresses (Cont.)

- **32-bit IP address:**
  - network-layer address for interface
  - used for layer 3 (network layer) forwarding
  - e.g.: 128.125.7.11

Decimal (base 10) notation
(each "numeral" represents 8bits)

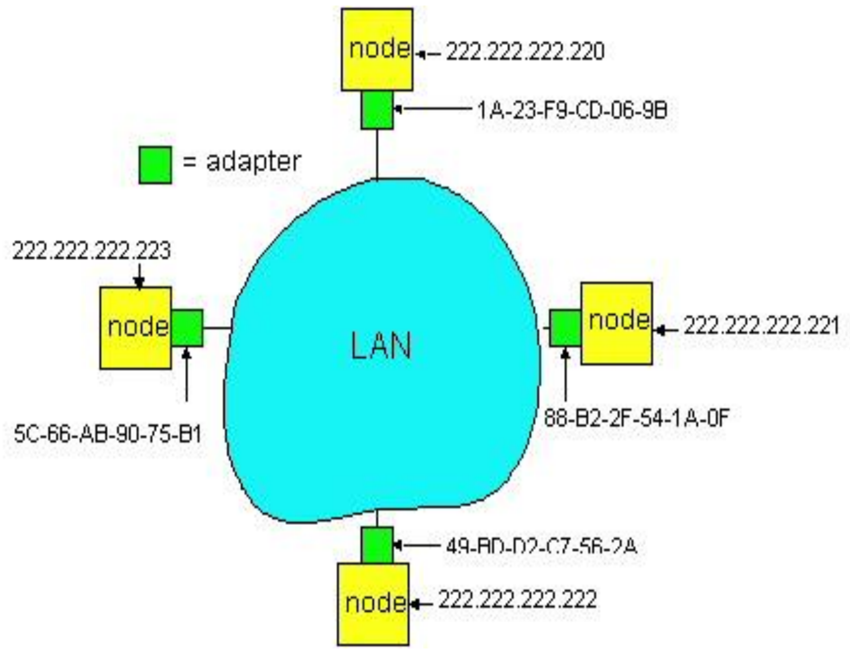
- MAC (or LAN or physical or Ethernet) address:
  - function: used 'locally" to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)
  - 48 bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation
(each "numeral" represents 4 bits)

Question #1:
How does a Host/Router get the MAC address of another Host/Router on the same LAN?

Address Resolution Protocol

Question: how to determine MAC address of B given B's IP address?

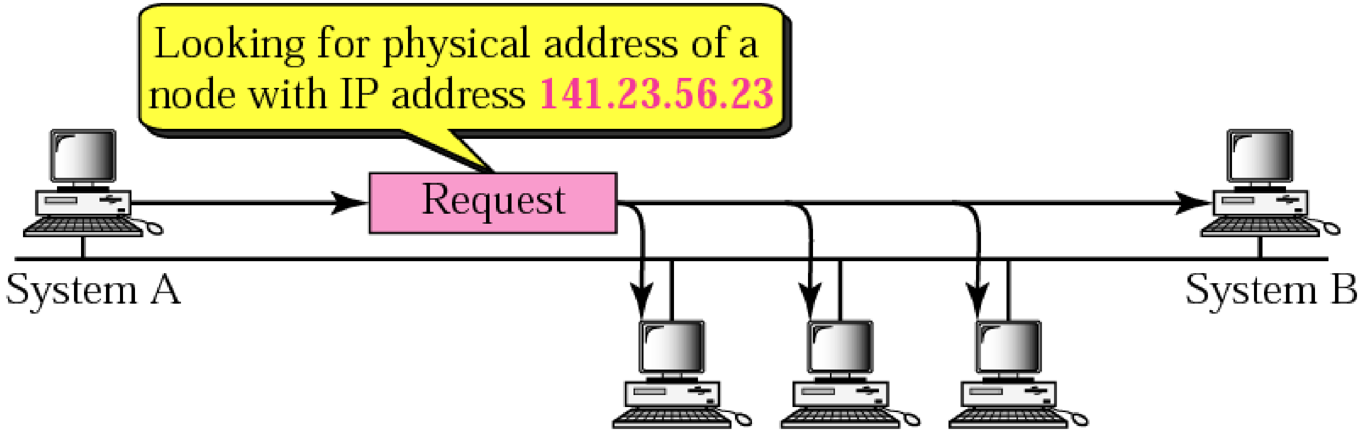


- Each IP node (Host, Router) on LAN has ARP module/table
- ARP Cache Table: IP/MAC address mappings for some LAN nodes
- Cache lifetime ~ 20 min

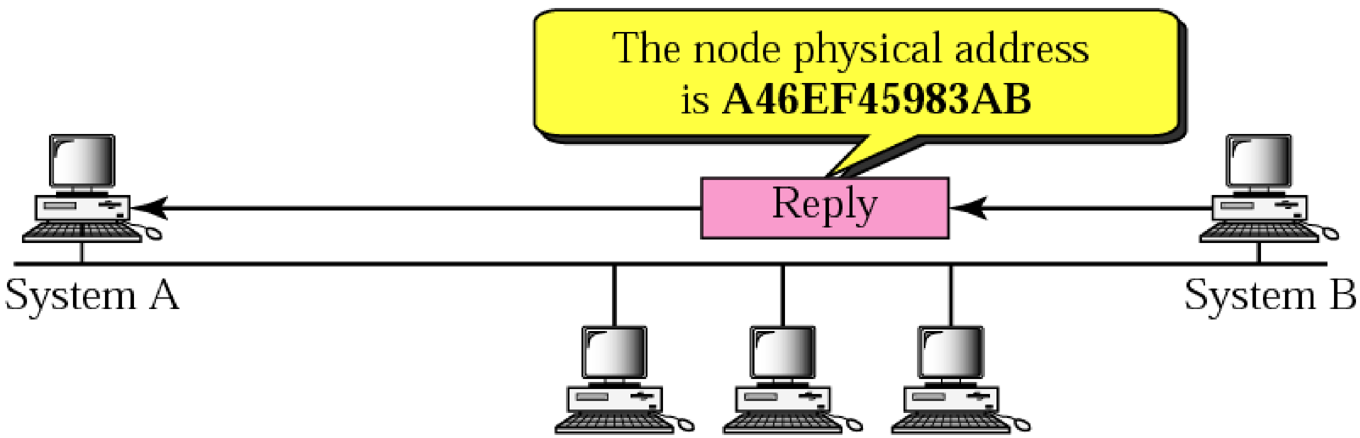
ARP (Continued)

- Address Resolution Protocol binds an IP address to a media (link) address
- ARP is a simple <u>request-response</u> protocol
  - Host "A" broadcasts a request packet containing IP address of "B". Broadcast MAC address is FF:FF:FF:FF:FF:FF. All hosts receive the ARP inquiry
  - Host "B" recognizes its IP address
  - Host "B" sends a response (not a broadcast) packet to first host containing its MAC address
  - Host "A" caches address mapping for later use
- ARP is a local, "Plug and Play" Protocol. Nodes create their ARP tables without intervention from net administrator

ARP Operation



a. ARP request is broadcast

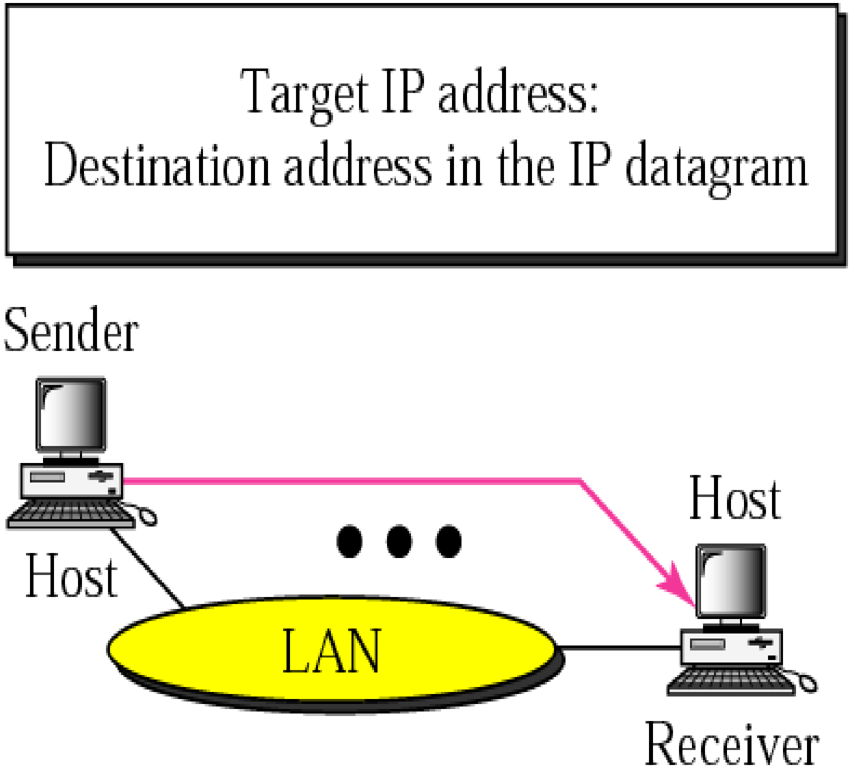


b. ARP reply is unicast

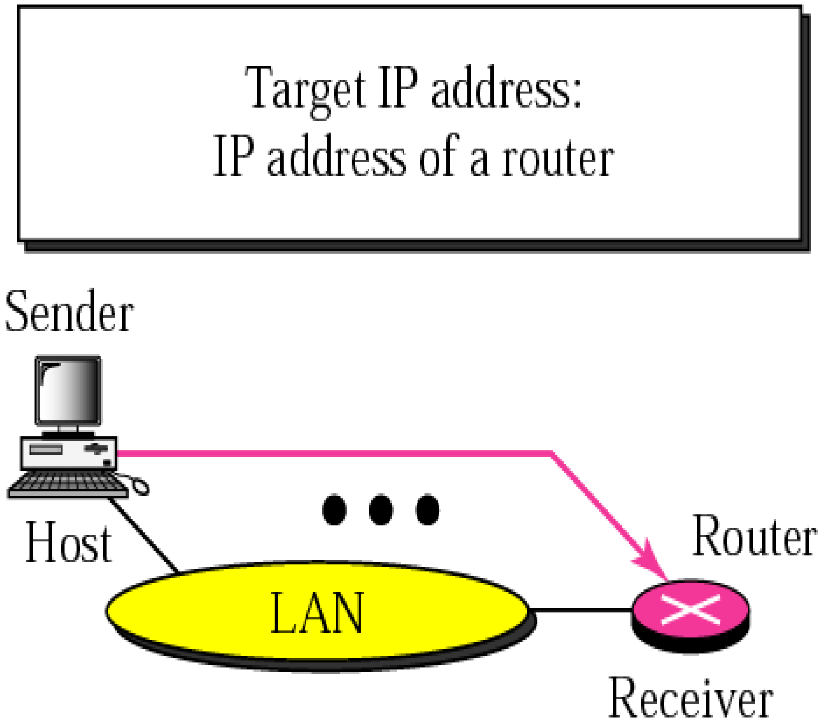
ARP Packet

Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

Four Cases of ARP use



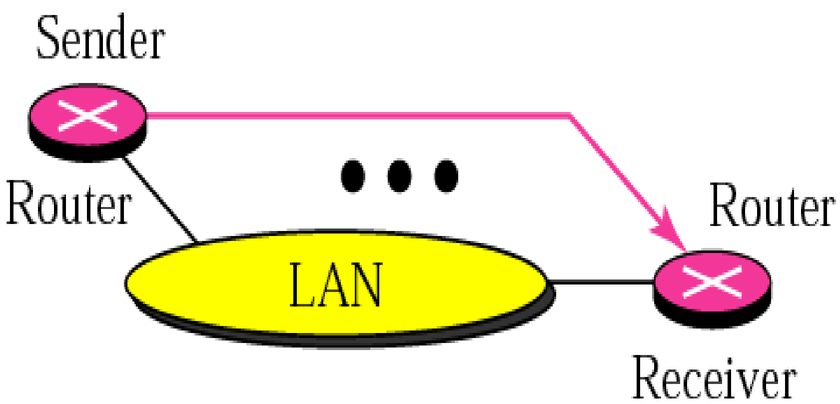
Case 1. A host has a packet to send to another host on the same network.



Case 2. A host wants to send a packet to another host on another network.
It must first be delivered to a router.

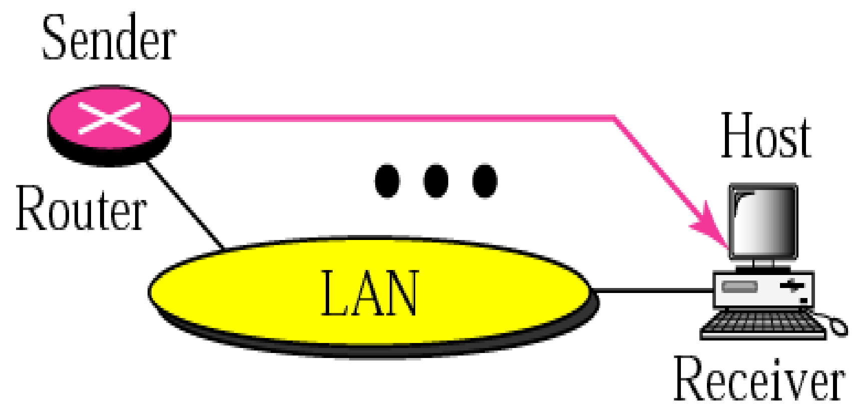
Four Cases of ARP use (Cont.)

Target IP address:
IP address of the appropriate router found in the routing table



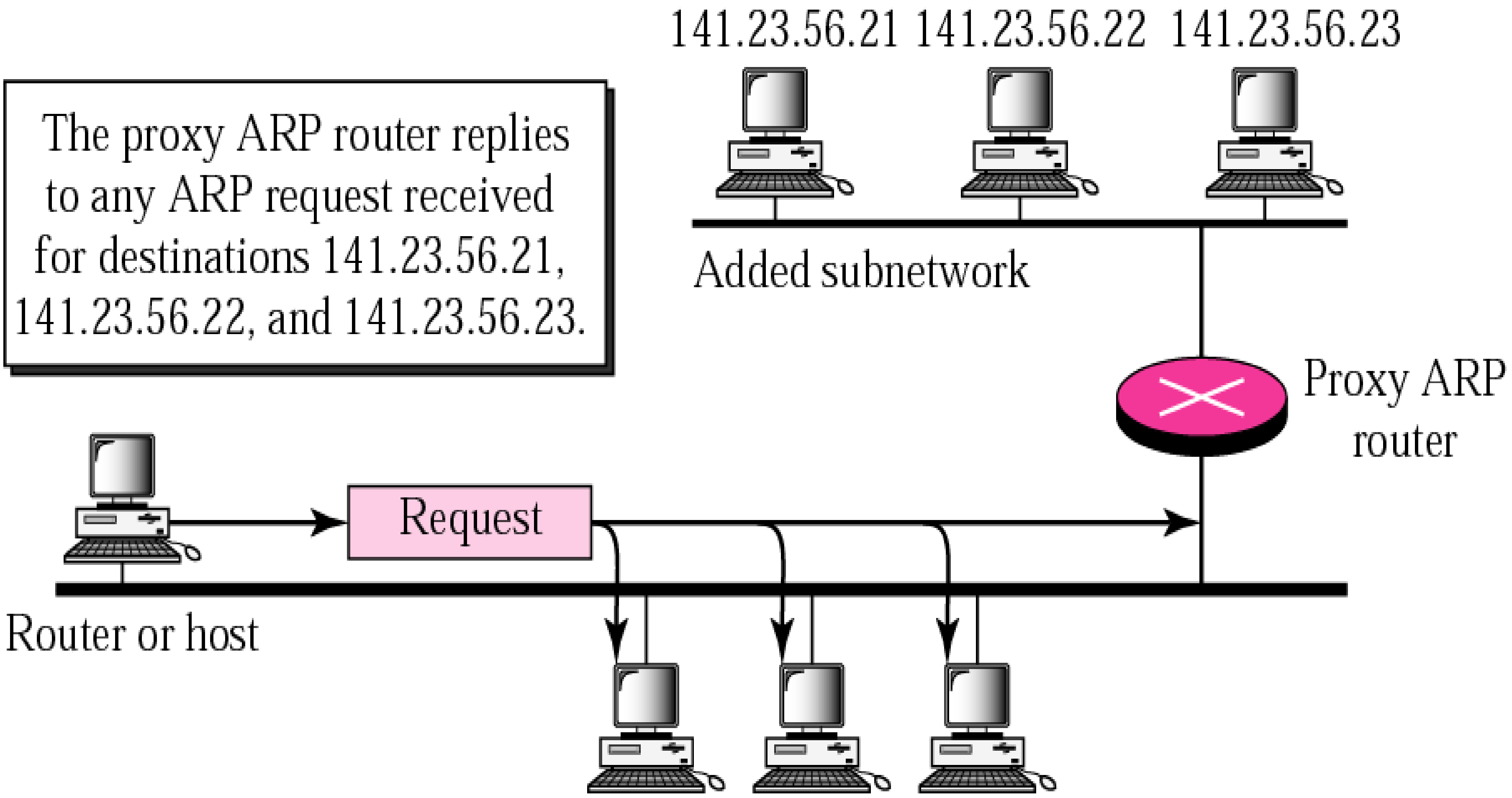
Case 3. A router receives a packet to be sent to a host on another network.
It must first be delivered to the appropriate router.

Target IP address:
Destination address in the IP datagram



Case 4. A router receives a packet to be sent to a host on the same network.

Proxy ARP



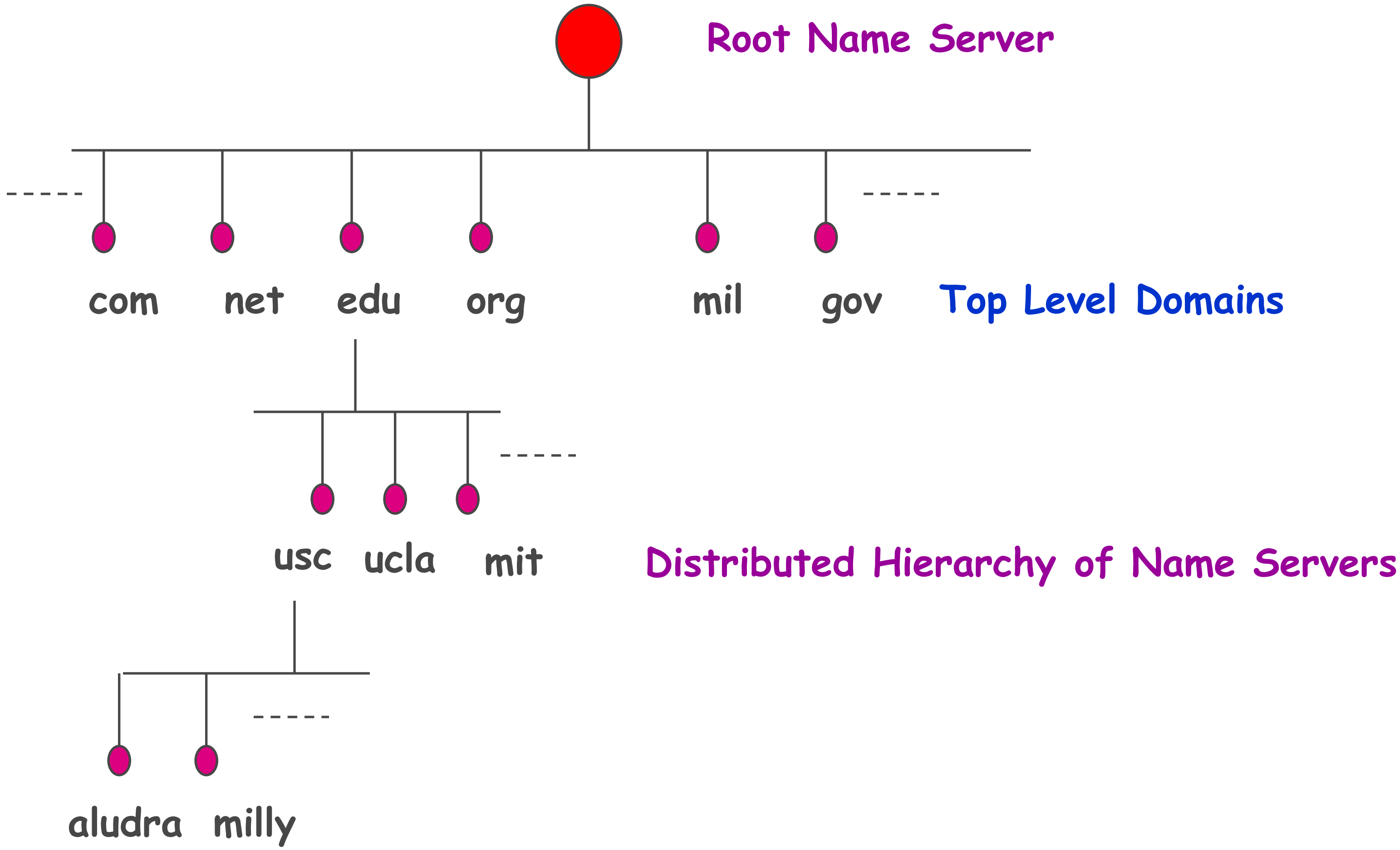
141.23.56.21
141.23.56.22
141.23.56.23
The proxy ARP router replies to any ARP request received for destinations 141.23.56.21, 141.23.56.22, and 141.23.56.23.
Added subnetwork
Proxy ARP router
Request
Router or host

Question #2:
How does a Host get the IP address of another Host across the Internet?

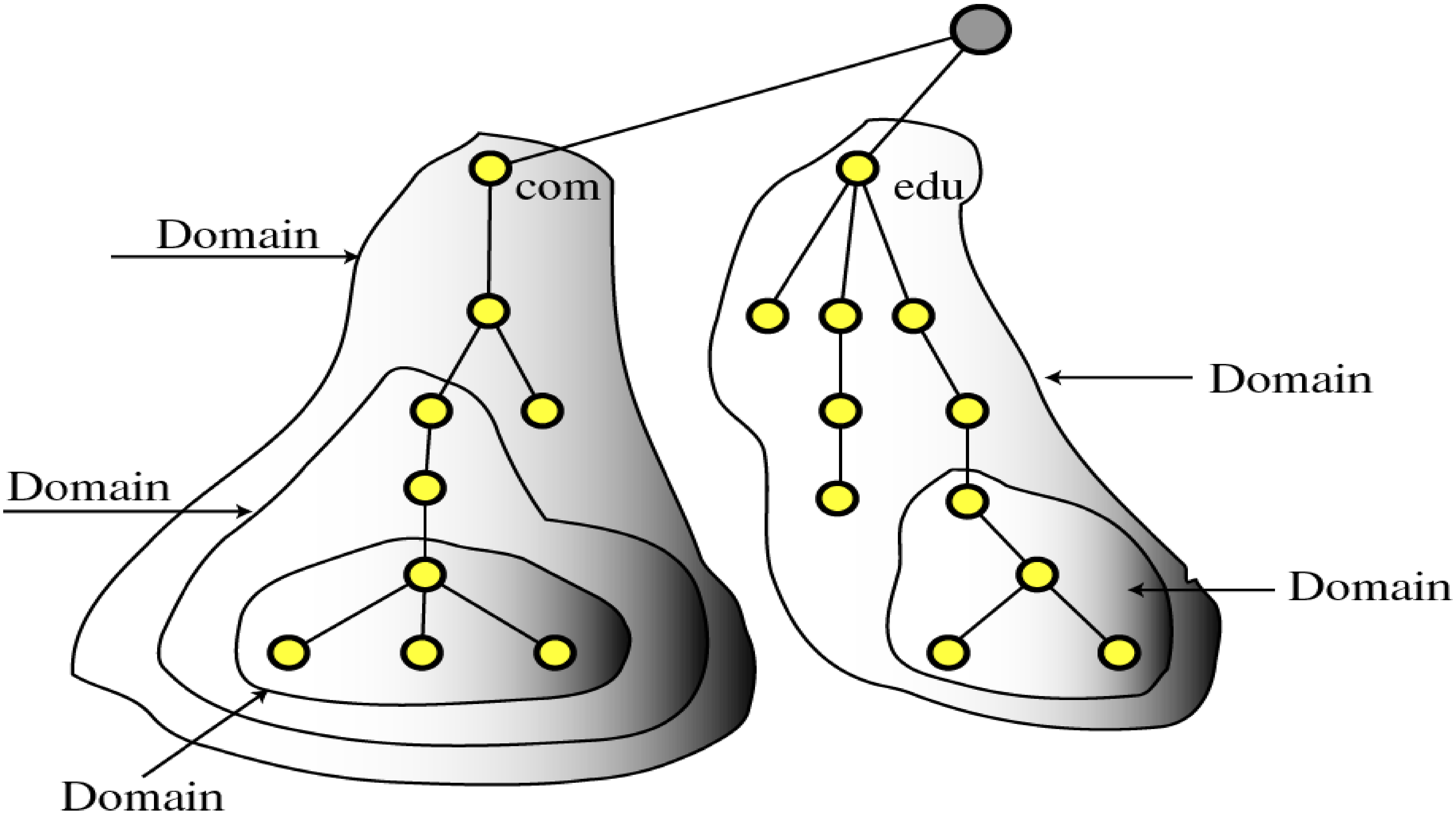
Domain Name Services (DNS)

- DNS is a TCP/IP client server application protocol that allow host and name servers to communicate in order to provide host name to IP address translation
- DNS uses a distributed, hierarchical naming structure by defining several Domains. A domain is a collection of sites that are related in some manner
- DNS use the services of UDP, port # 53
- Application protocols such as HTTP, FTP, SMTP, etc... use the services of DNS

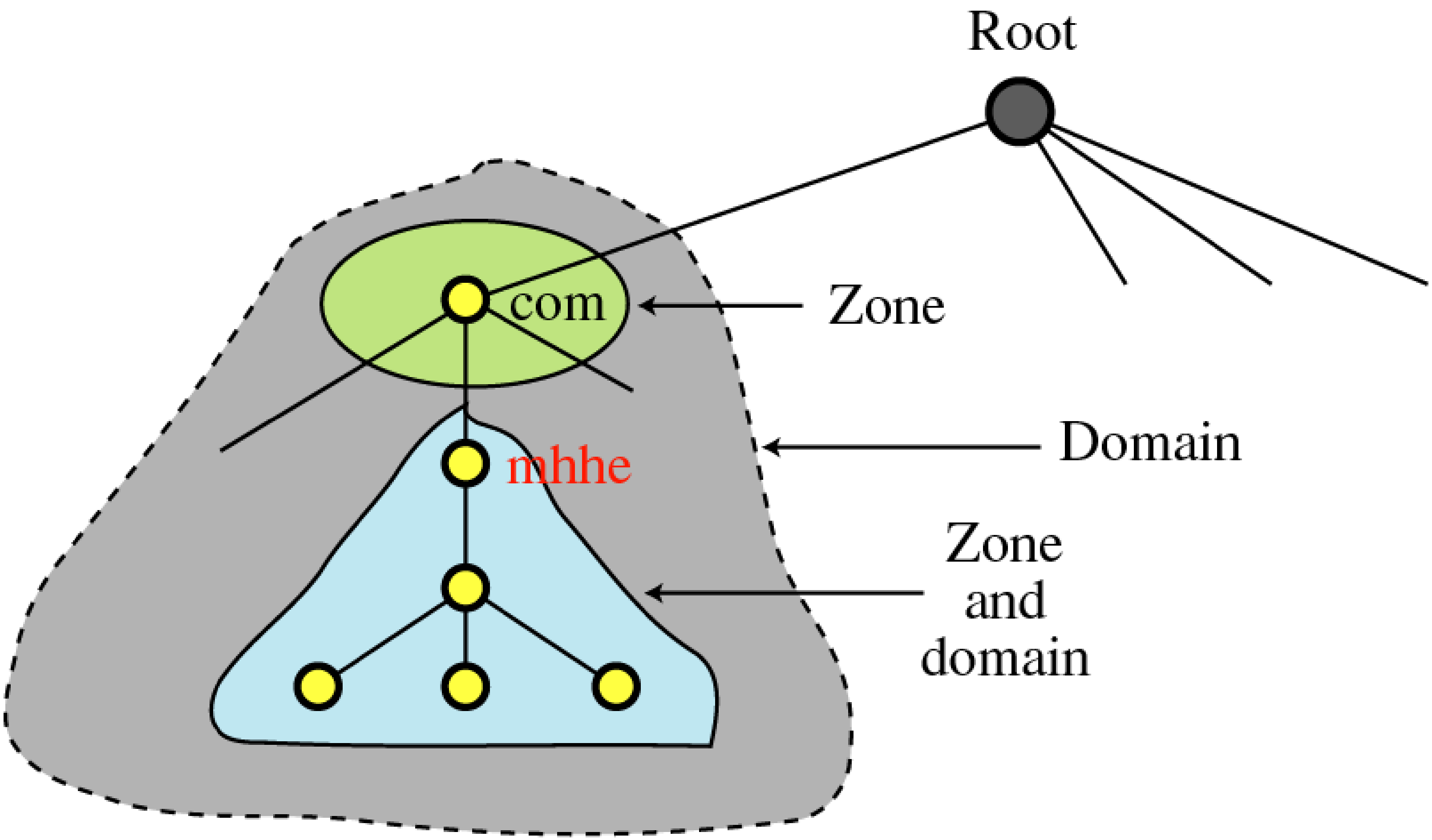
Domain Name Space (Distributed)



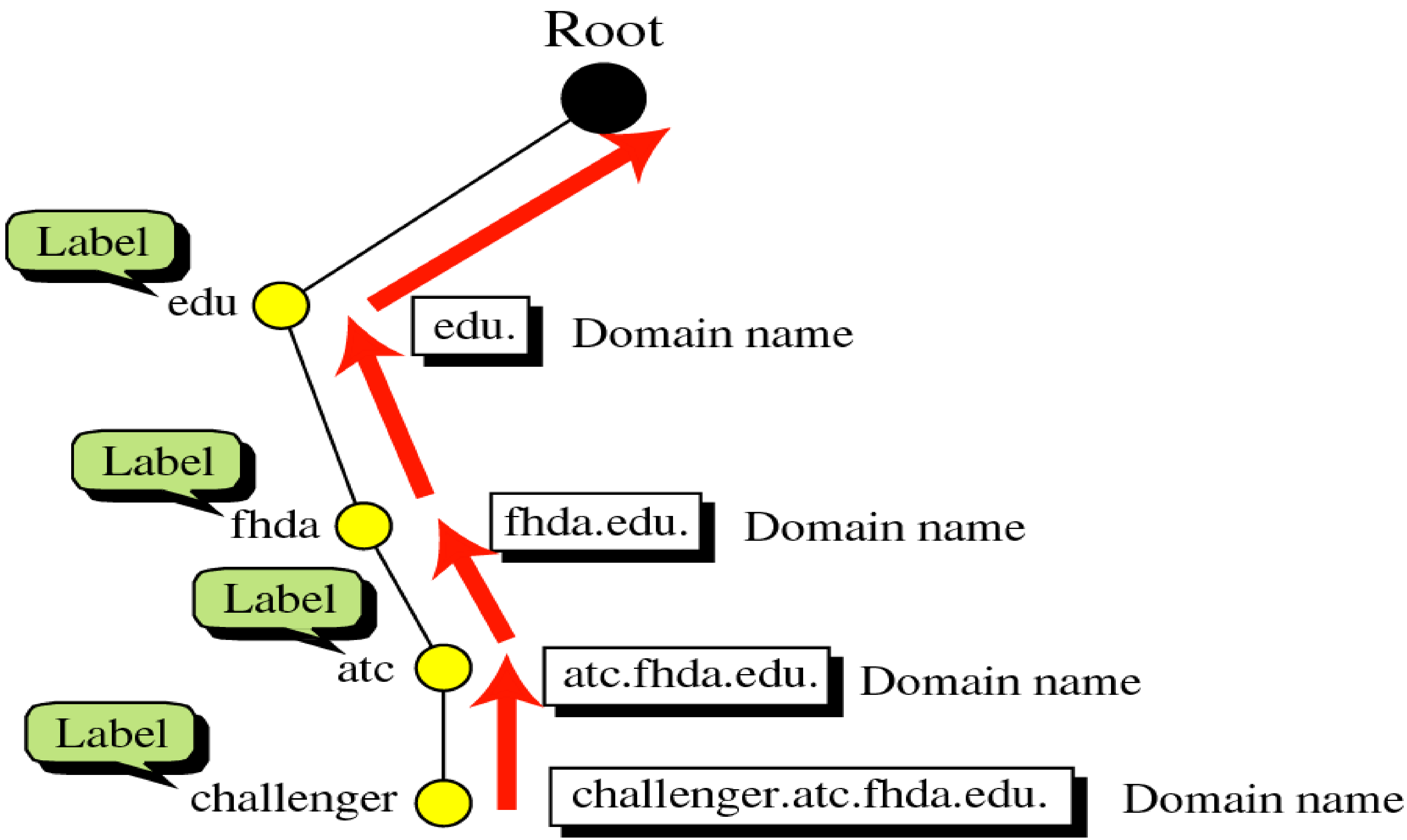
Domains



Domains and Zones



Domain Names & Labels

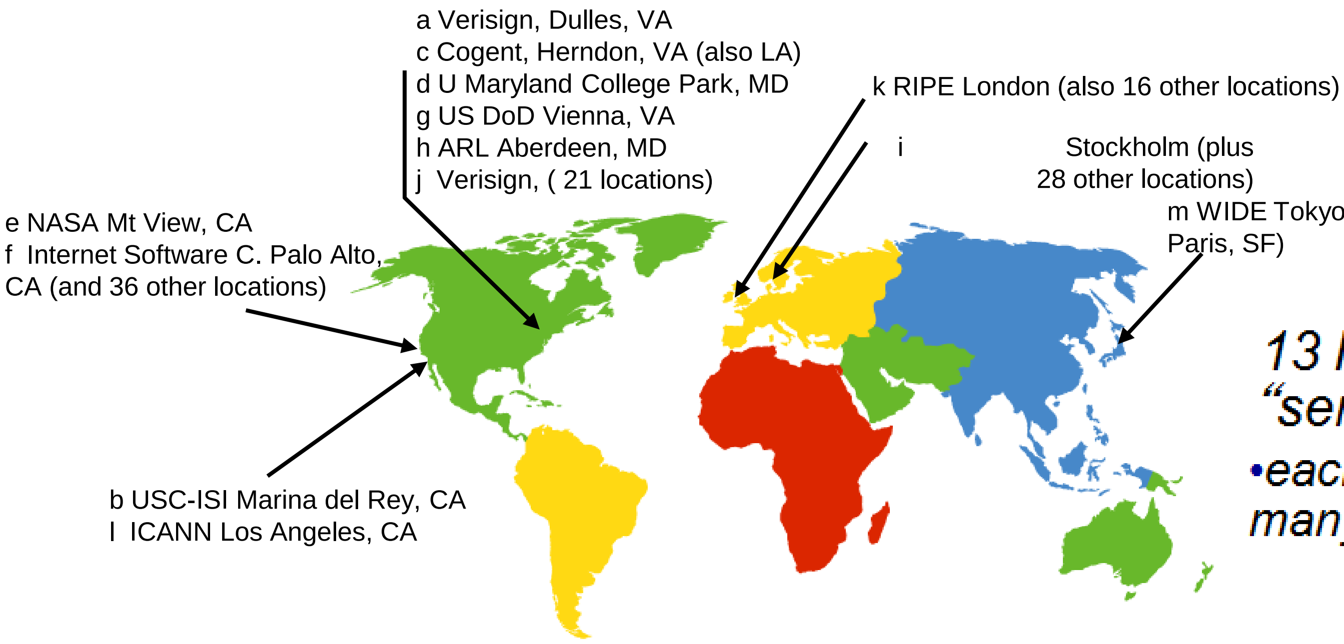


Name Servers

- Local Name Servers: This is the default name server (in department, university, company, residential ISP, etc…) that will receive the DNS query from the host
  - The IP address of the default local name server is configured manually (or Dynamically by DHCP) in the host
- Root Name Servers: There are 13 root name servers most of which are located in US (two of them in Marina Del Rey). When a local name server can't satisfy the query from a host, it will behave as a DNS client and queries one of the root servers. If the root name server can't satisfy the query, it consult with
- Authoritative Name Server: This is where the host register its name/IP address

Root Name Servers

- Contacted by local name server that can not resolve name
- Root name server:
  - Contacts authoritative name server if name mapping not known
  - Gets mapping
  - Returns mapping to local name server



13 logical root name "servers" worldwide

- *each "server" replicated many times*

TLD and Authoritative Name Servers

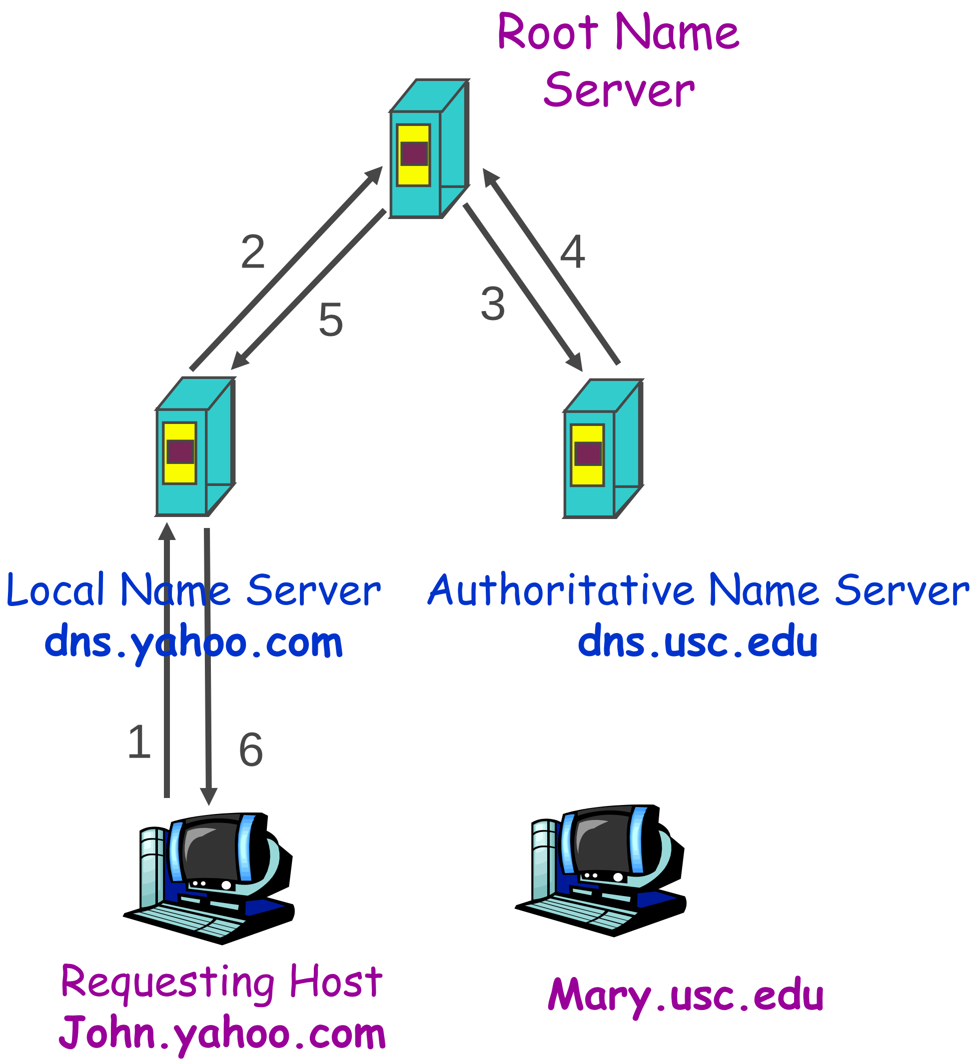
- Top-level domain (TLD) servers:
  - Responsible for com, org, net, edu, etc, and all top-level country domains uk, fr, ca, jp, in, cn
  - Network Solutions maintains servers for com TLD
  - Educause for edu TLD
- Authoritative DNS servers:
  - Organization's DNS servers, providing authoritative hostname to IP mappings for organization's servers (e.g., Web, mail).
  - Can be maintained by organization or service provider

Local Name Server

- Does not strictly belong to hierarchy
- Each ISP (residential ISP, company, university) has one.
  - also called "default name server"
- when host makes DNS query, query is sent to its local DNS server
  - acts as proxy, forwards query into hierarchy

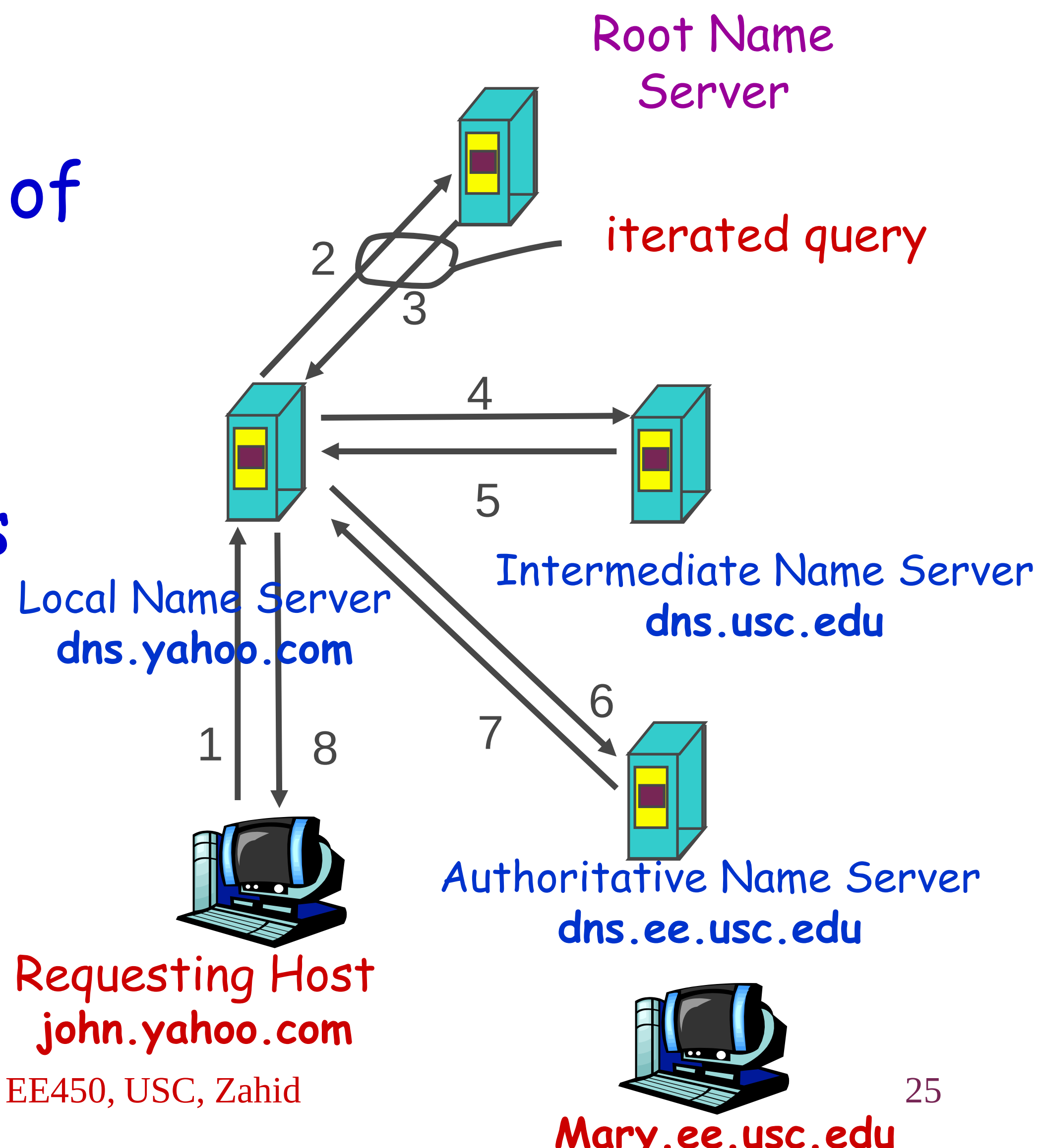
Recursive DNS

- Host "A" whose name is John.yahoo.com wants the IP address of another host "B" whose name is Mary.usc.edu
- Host "A" Contacts its local DNS server, dns.yahoo.com
- dns.yahoo.com contacts root name server, if necessary
- Root name server contacts authoritative name server, dns.usc.com, if necessary

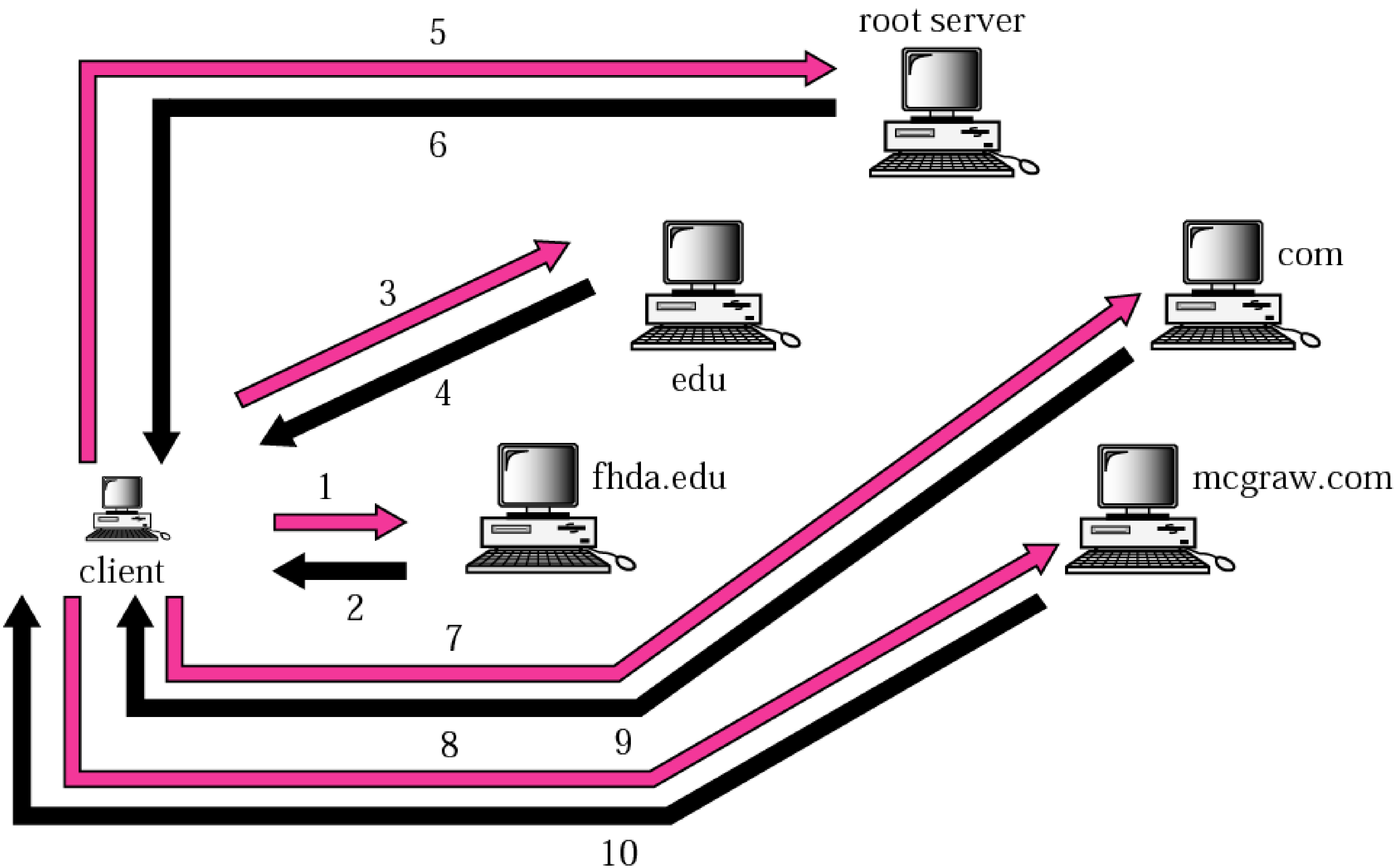


Iterative DNS

- Contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



Pure Iterative Resolution



DNS Caching

- Once (any) name server learns mapping, it caches mapping
  - Cache entries timeout (disappear) after some time
  - TLD servers typically cached in local name servers
    - Thus root name servers not often visited

Why not Centralized DNS?

- A centralized DNS represent a single point of failure. If the name server crashes so would the entire internet
- All traffic volume would have to be handled by this name server
- A single name server can't be close to all query clients $\Rightarrow$ increased delays $\Rightarrow$ World Wide Wait !!!!!
- Maintaining and updating a single name server is a huge task. Just dealing wit authentication/authorization is a nightmare

$\Rightarrow$ A single Name Server doesn't scale !

DNS Records

DNS: distributed database storing resource records (RR)

RR format: **(name, value, type, ttl)**

type=A

- **name** is hostname
- **value** is IP address

type=NS

- **name** is domain (e.g., foo.com)
- **value** is hostname of authoritative name server for this domain

type=CNAME

- **name** is alias name for some "canonical" (the real) name
- **www.ibm.com** is really **servereast.backup2.ibm.com**
- **value** is canonical name

type=MX

- **value** is name of mailserver associated with **name**

Question #3:
How does a Host get an IP address?

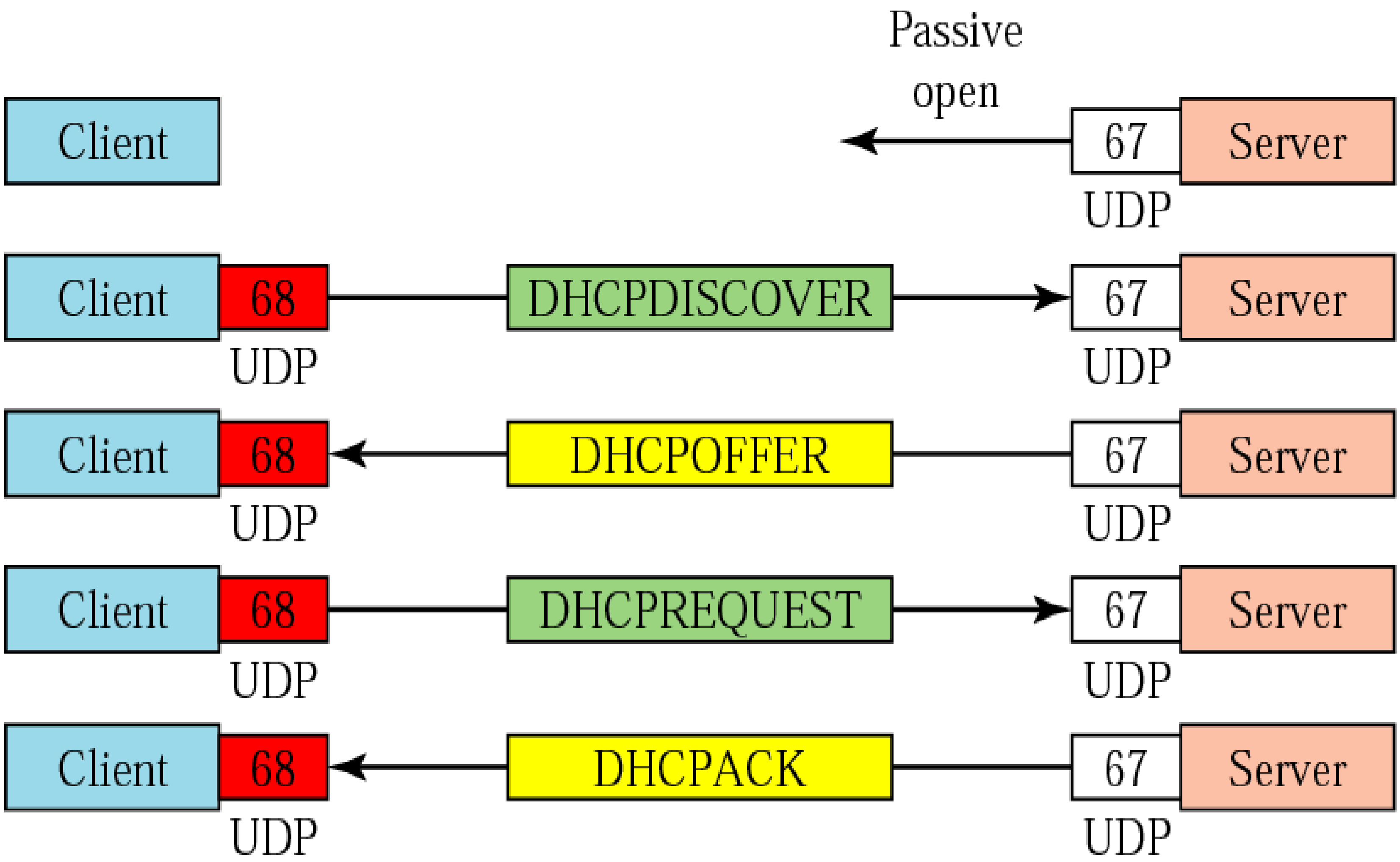
Host Configuration

- Problem of managing IP addresses within a customer network
  - Assigning IP address
  - Reclaiming IP addresses
- Manual management of IP addresses is difficult
  - Error-prone
  - Mobility of hosts

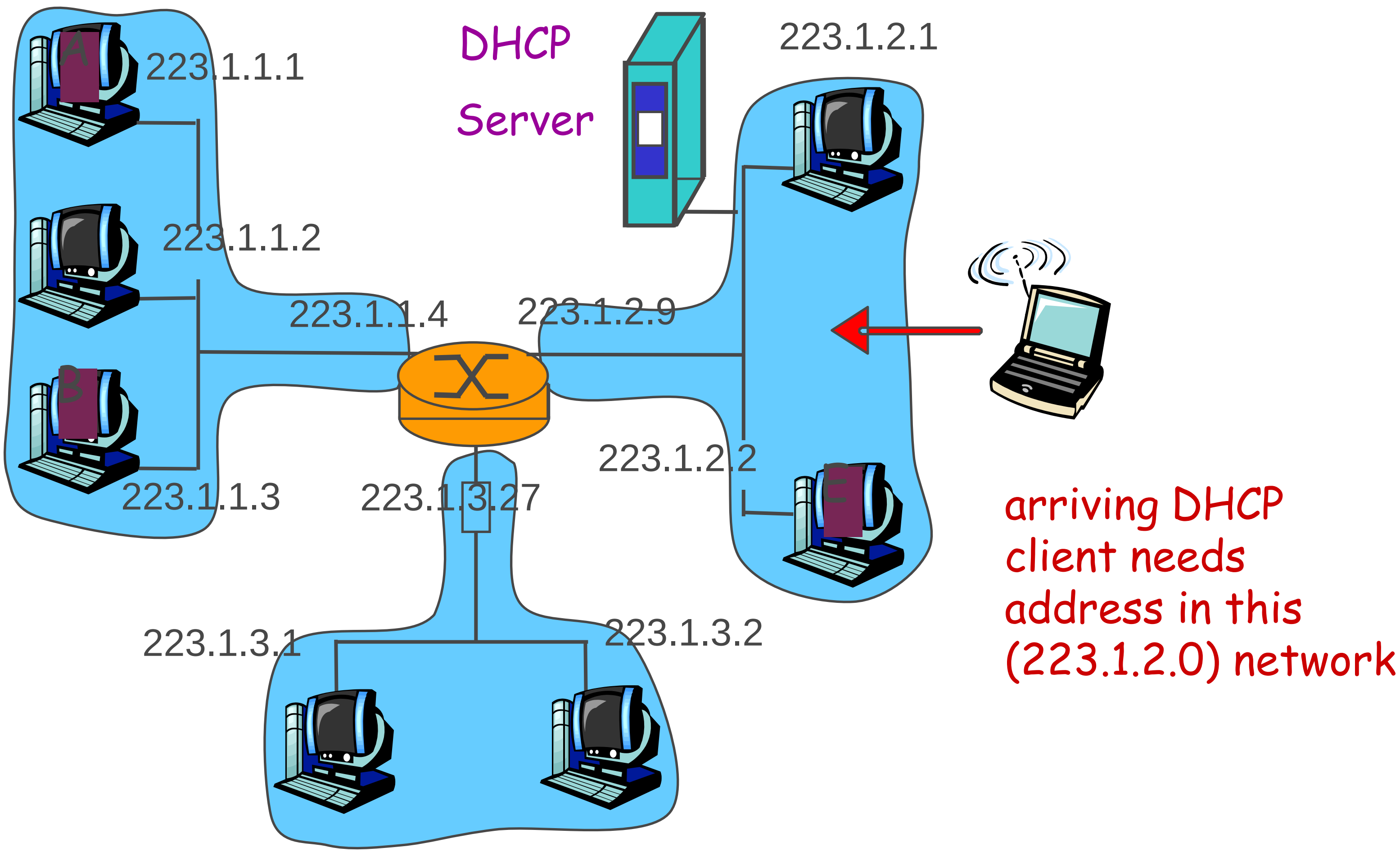
Dynamic Host Configuration Protocol (DHCP)

- DHCP is a client/server application designed to provide a centralized approach to configuring and maintaining IP addresses
- Four basic steps involved in obtaining an IP address:
  - Discovery Phase
  - Offer Phase
  - Request Phase
  - Acknowledgement Phase

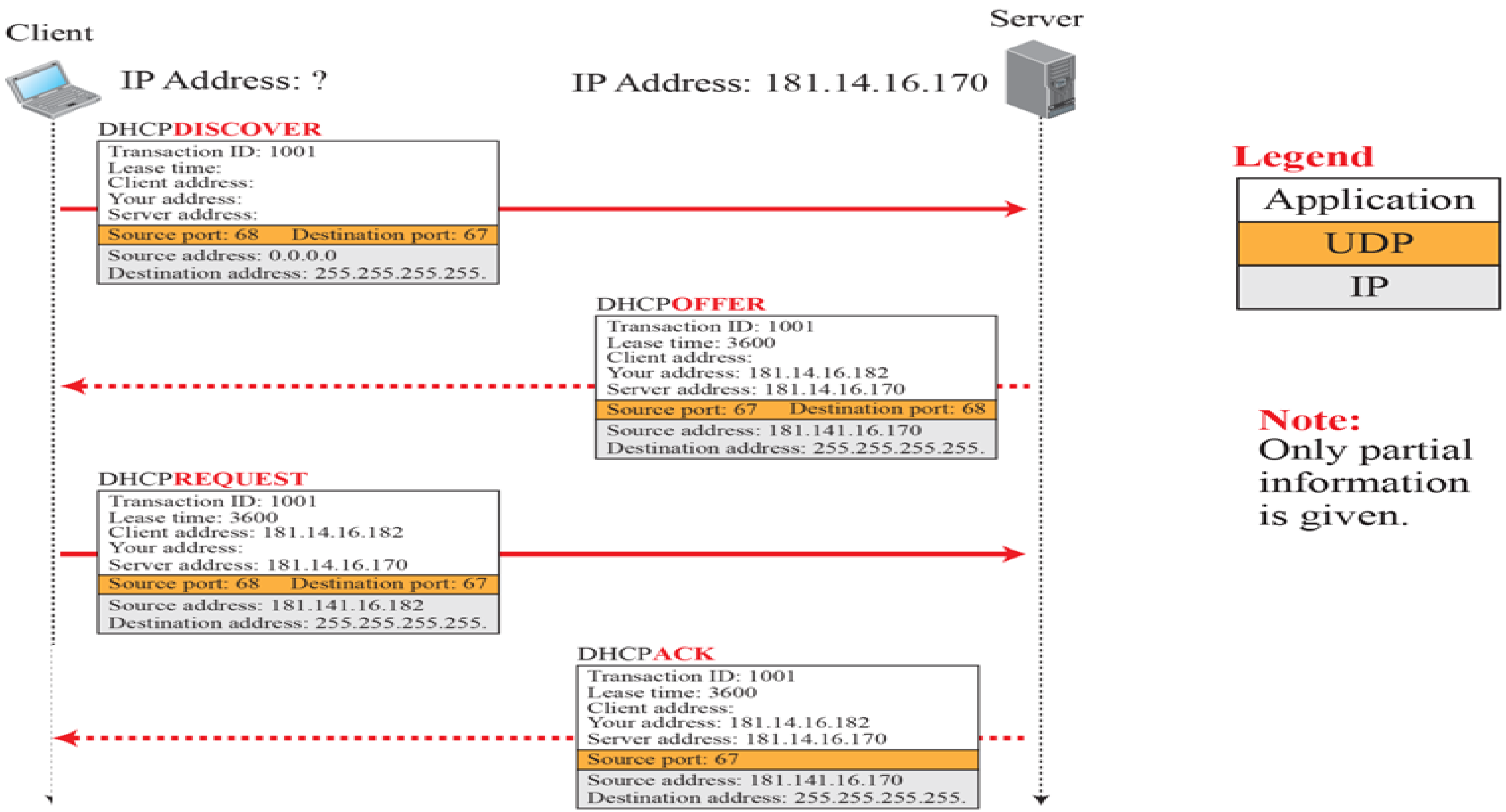
DHCP Exchange Messages



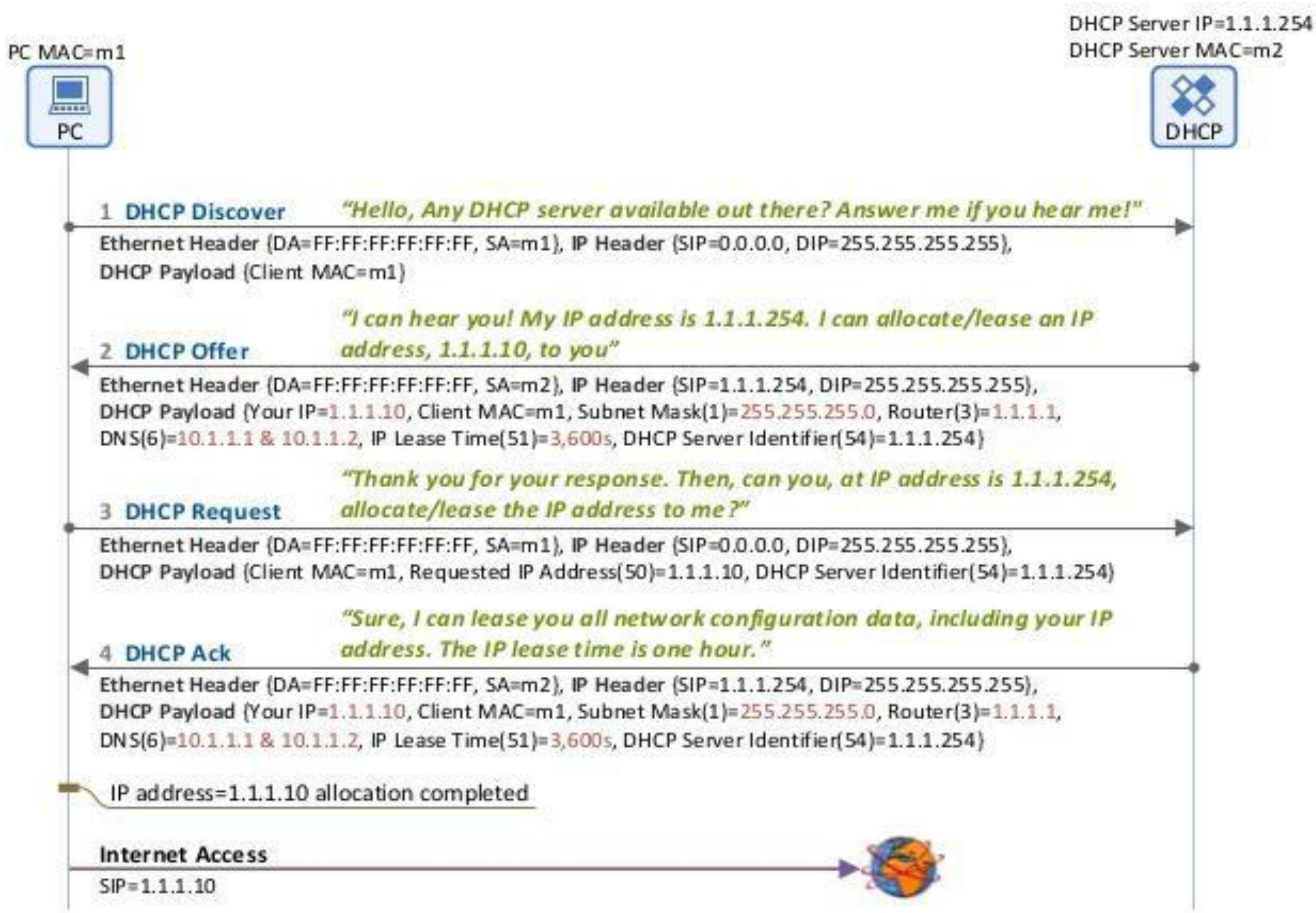
DHCP Client/Server Scenario



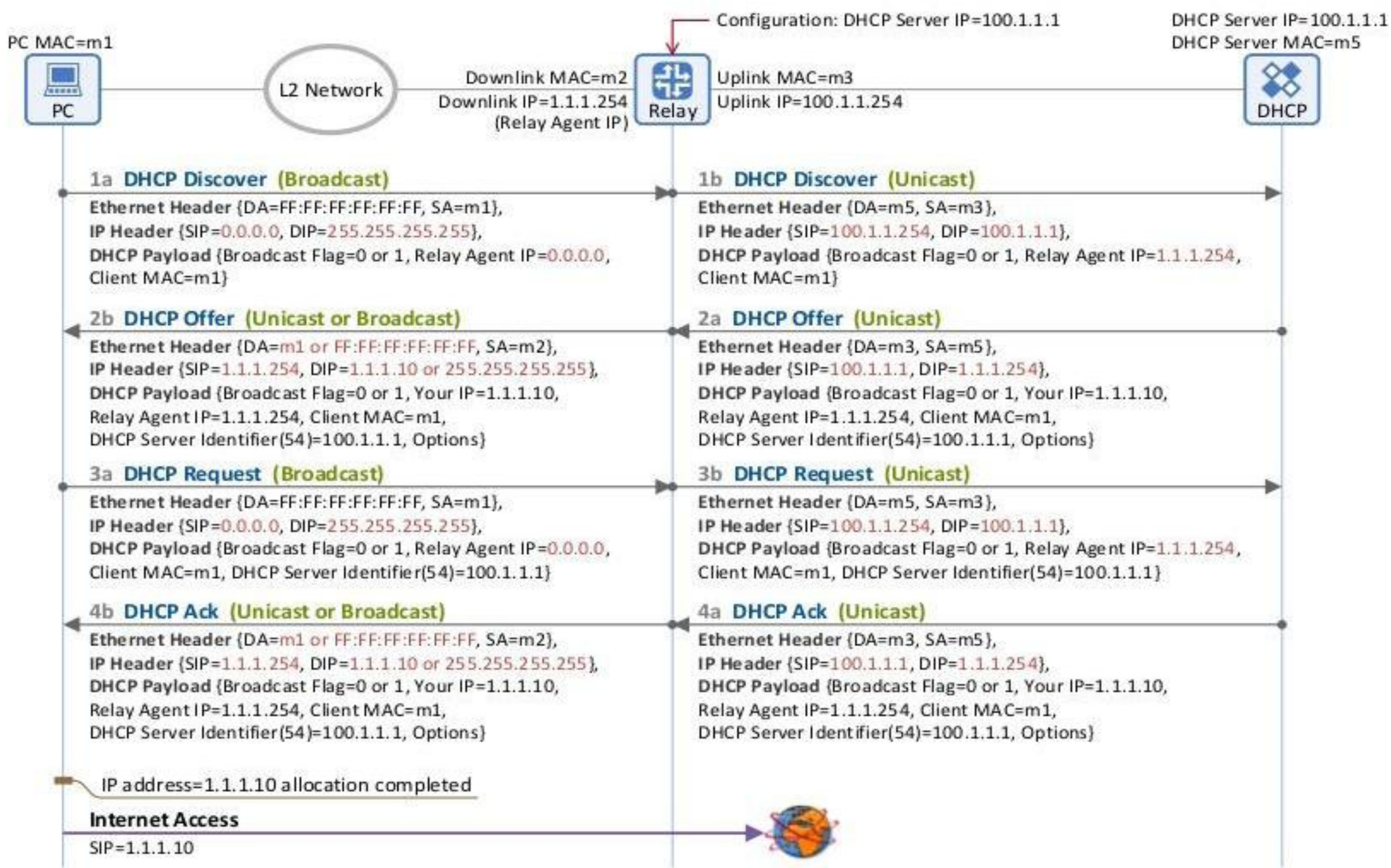
DHCP Client/Server Scenario



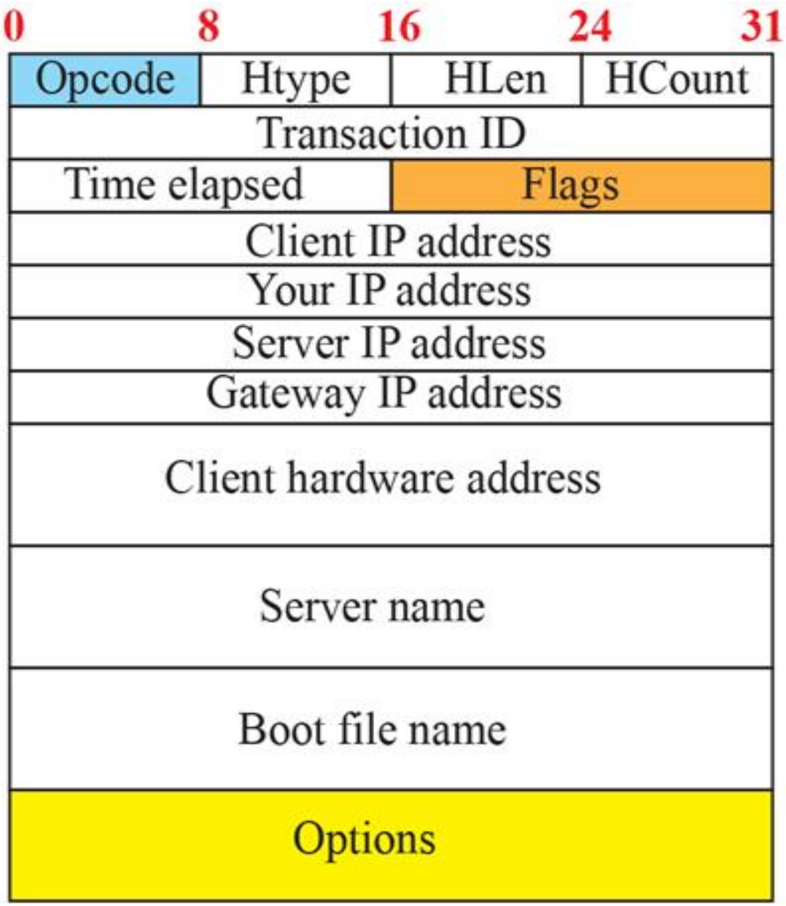
DHCP Basic Operations (When Both the Client and the Server are on same Network



DHCP Relay Agent (Needed when the Client and the Server are located on different Networks



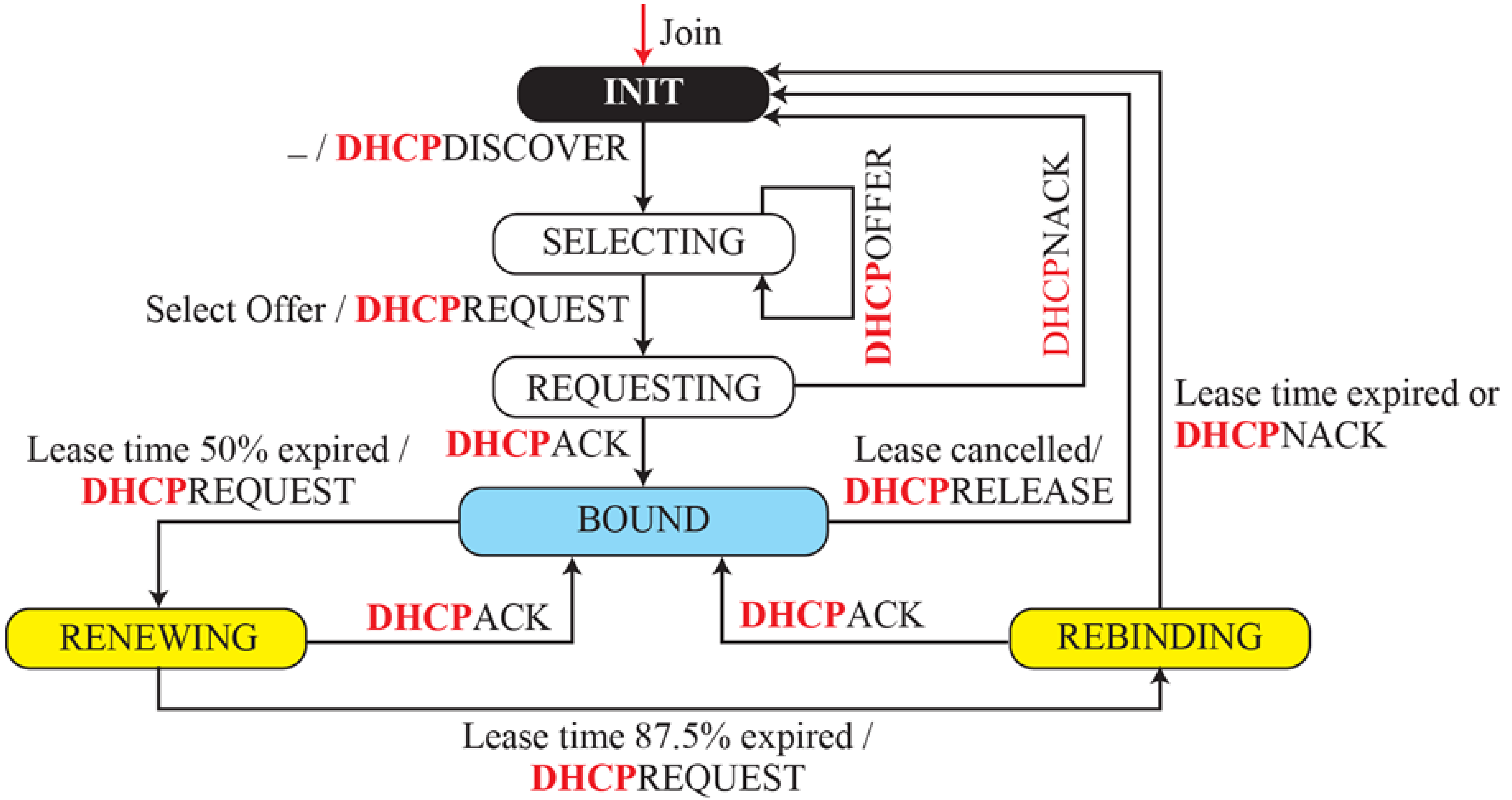
DHCP Message Format



Fields:
Opcode: Operation code, request (1) or reply (2)
Htype: Hardware type (Ethernet, ...)
HLen: Lengh of hardware address
HCount: Maximum number of hops the packet can travel
Transaction ID: An integer set by client and repeated by the server
Time elapsed: The number of seconds since the client started to boot
Flags: First bit defines unicast (0) or multicast (1); other 15 bits not used
Client IP address: Set to 0 if the client does not know it
Your IP address: The client IP address sent by the server
Server IP address: A broadcast IP address if client does not know it
Gateway IP address: The address of default router
Server name: A 64-byte domain name of the server
Boot file name: A 128-byte file name holding extra information
Options: A 64-byte field with dual purpose described in text

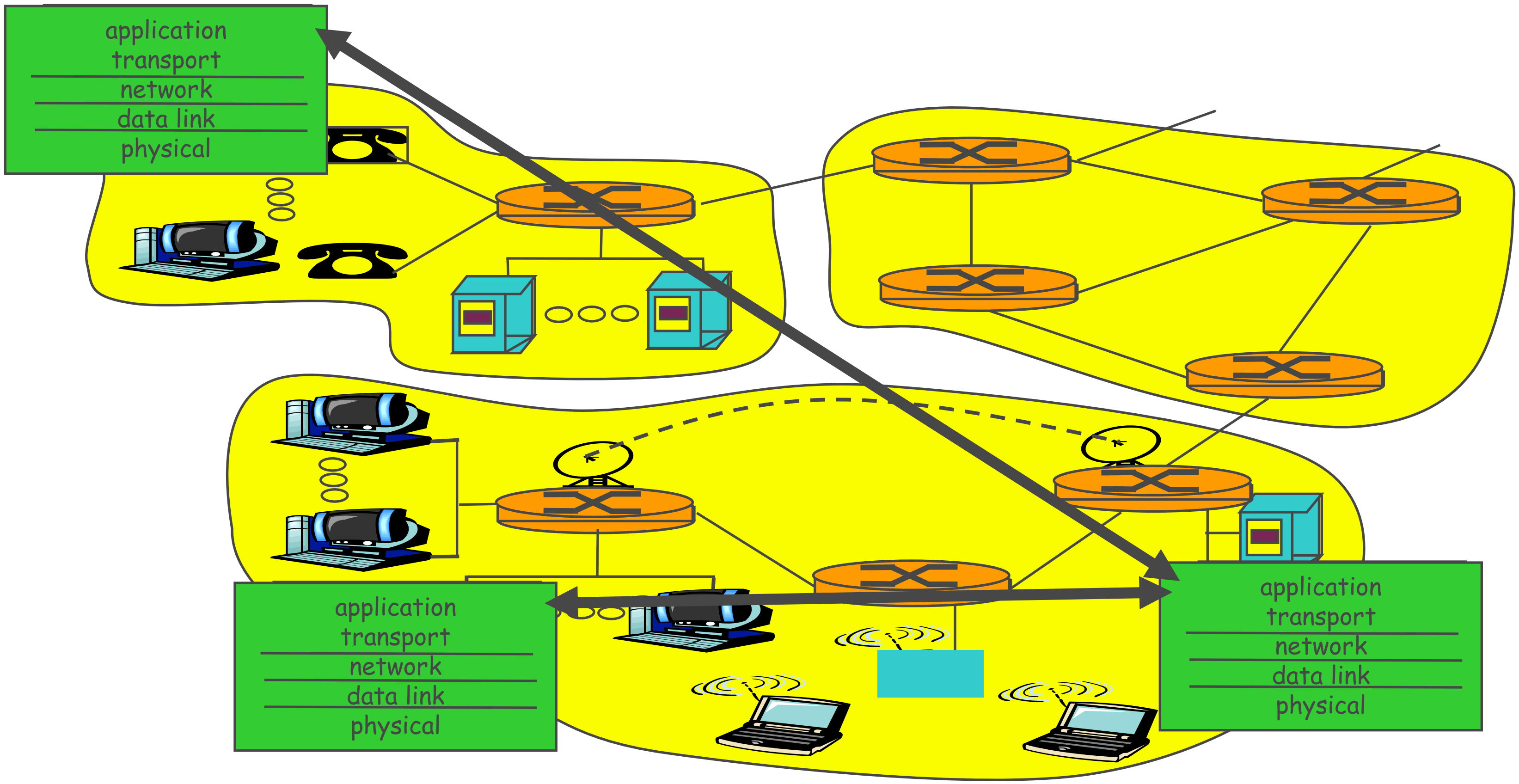
1 **DHCP**DISCOVER	5 **DHCP**ACK
2 **DHCP**OFFER	6 **DHCP**NACK
3 **DHCP**REQUEST	7 **DHCP**RELEASE
4 **DHCP**DECLINE	8 **DHCP**INFORM

DHCP State Transition



Question #4:
How does a Process (Application) "identify" the other process with which it wants to communicate?

Client/Server Paradigm



Process-to-Process Communications

- Host-to-Host Communications: IP
- IP delivery is incomplete. Message must be delivered to the correct process running in destination host
- Both local and remote hosts can be running several processes at the same time ⇒ we need to be able to distinguish between these processes
- For communications to take place we need to define
    - Local host/Local process
    - Remote host/remote process
    - Transport layer protocol providing transport services

Port Numbers

- Port Numbers allow receiving host to determine to which local process the message be delivered
- Port numbers are integers between 0 and 65,535
- Client process defines itself with a port number chosen **randomly** by the underlying transport layer protocol.
- Server process defines itself by a **well-known port number**. The ports ranging from 0~1023 are well-known port numbers and are assigned and controlled by **ICANN**.

Socket Address

- The combination of the IP address, the port number and the transport layer protocol defines the "Socket Address" which uniquely defines the communications between the client process and the server process