Ch 6: Networking Services: NAT, DHCP, DNS, Multicasting

Magda El Zarki
Prof. of CS
Univ. of CA, Irvine
Email: elzarki@uci.edu
http: www.ics.uci.edu/~magda

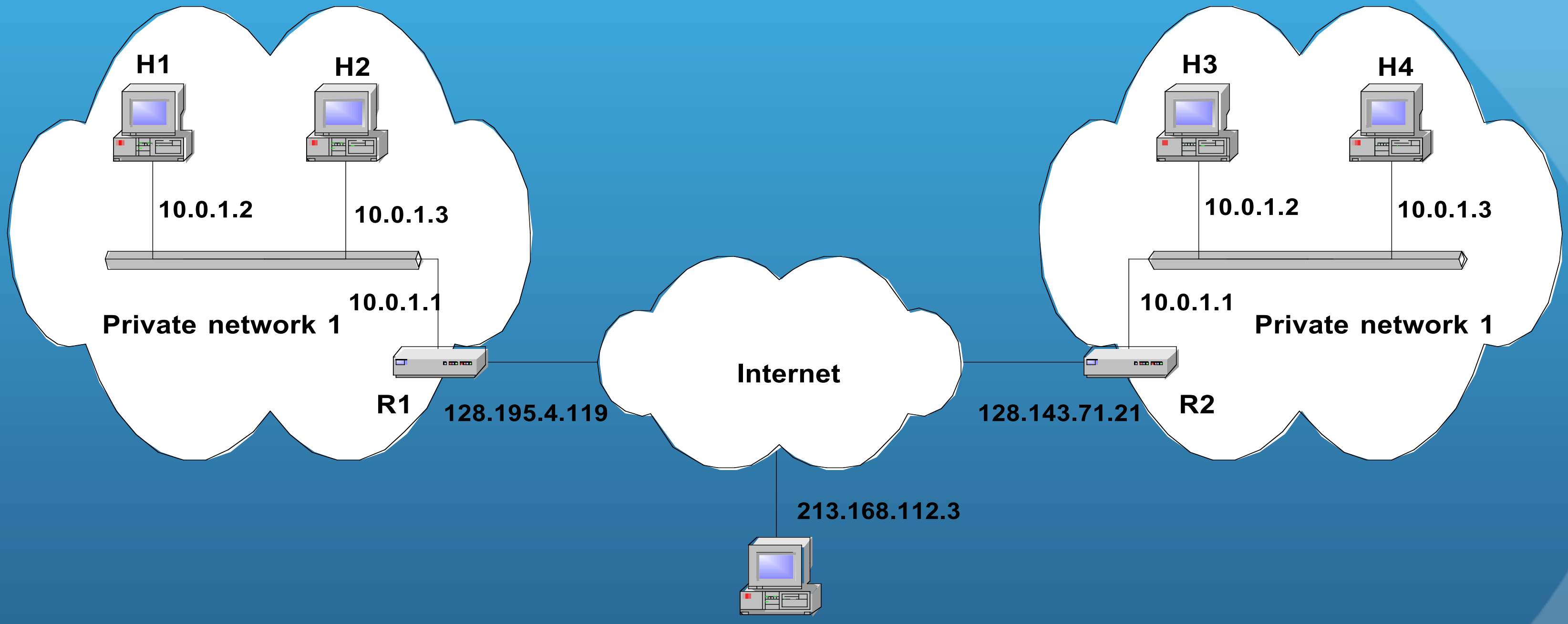
Overivew of NAT

- NAT: Network Address Translation

Private Network

- *Private IP* network is an IP network that is not directly connected to the Internet
- IP addresses in a private network can be assigned arbitrarily.
  - Not registered and not guaranteed to be globally unique
- Generally, private networks use addresses from the following experimental address ranges (*non-routable addresses*):
  - 10.0.0.0 – 10.255.255.255
  - 172.16.0.0 – 172.31.255.255
  - 192.168.0.0 – 192.168.255.255

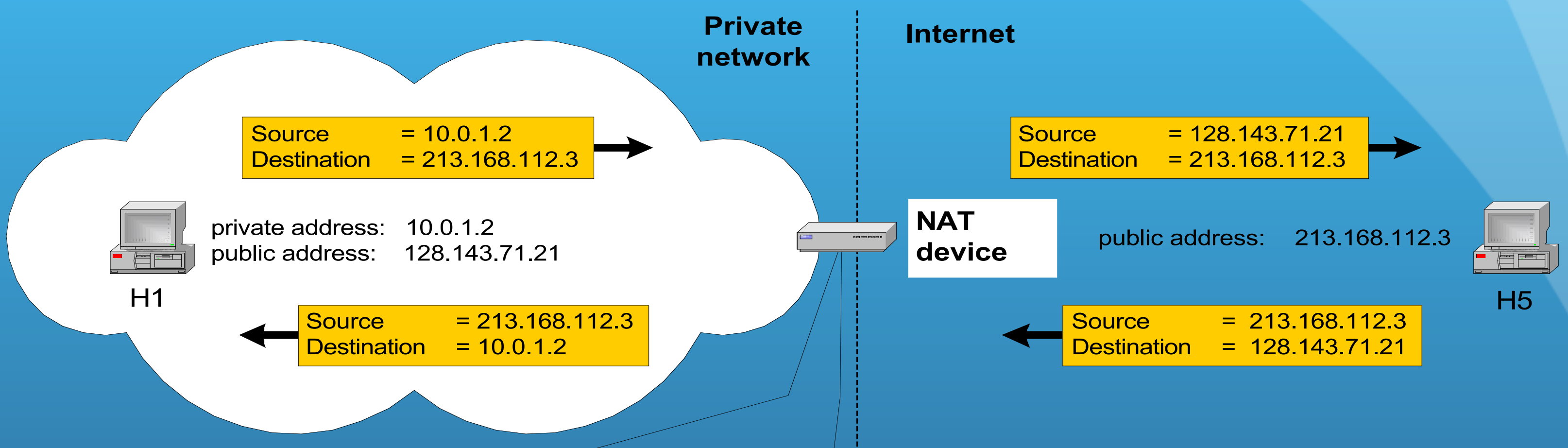
Private Addresses



Network Address Translation (NAT)

- NAT is a router function where **IP addresses** (and possibly port numbers) of IP datagrams are **replaced** at the boundary of a private network
- NAT is a method that enables **hosts** on **private** networks to communicate with hosts on the **Internet**
- NAT is run on routers that connect private networks to the public Internet, to **replace** the **IP address-port pair** of an IP packet with another IP address-port pair.

Basic operation of NAT



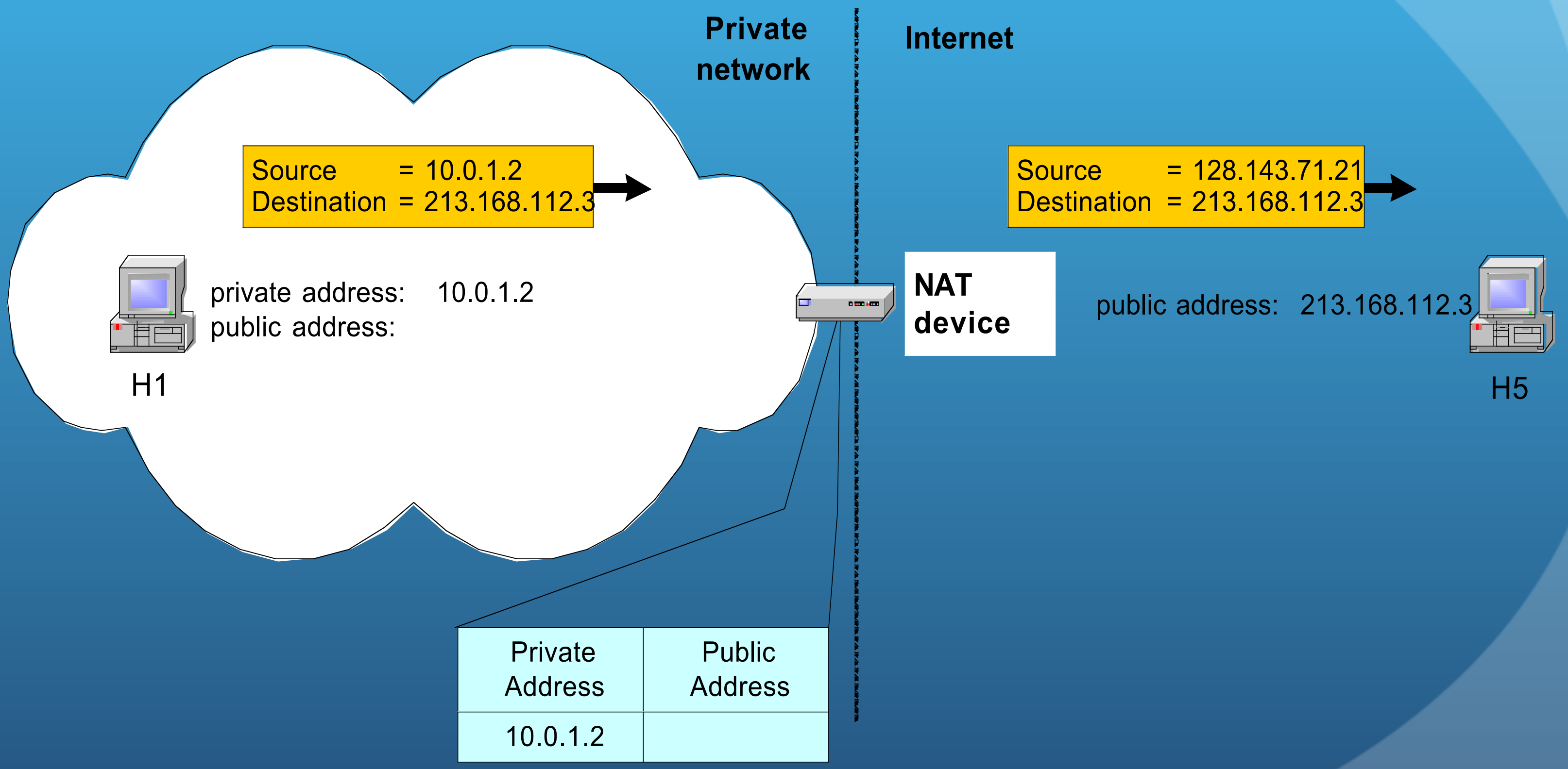
Private Address	Public Address
10.0.1.2	128.143.71.21

- NAT device has address translation table
- One to one address translation

Pooling of IP addresses

- **Scenario:** Corporate network has many hosts but only a small number of public IP addresses
- **NAT solution:**
  - Corporate network is managed with a private address space
  - NAT device, located at the boundary between the corporate network and the public Internet, manages a pool of public IP addresses
  - When a host from the corporate network sends an IP datagram to a host in the public Internet, the NAT device picks a public IP address from the address pool, and binds this address to the private address of the host

Pooling of IP addresses



Pool of addresses: 128.143.71.0-128.143.71.30

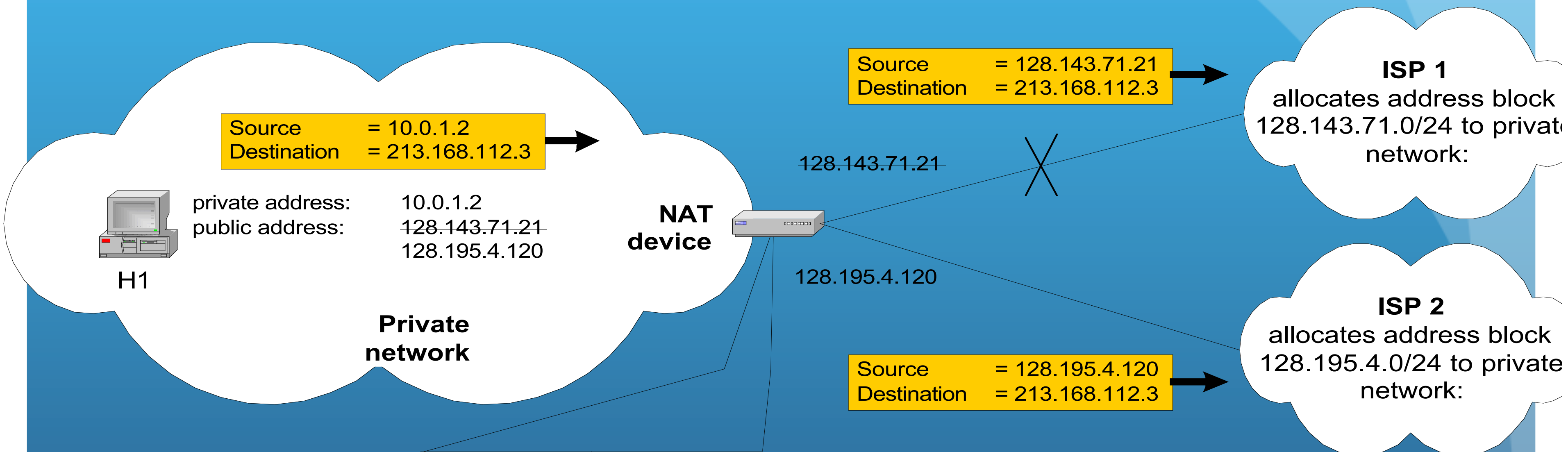
Supporting migration between network service providers

- **Scenario:** In CIDR, the IP addresses in a corporate network are obtained from the service provider. Changing the service provider requires changing all IP addresses in the network.

- **NAT solution:**
  - Assign private addresses to the hosts of the corporate network
  - NAT device has static address translation entries which bind the private address of a host to the public address.
  - Migration to a new network service provider merely requires an update of the NAT device. The migration is not noticeable to the hosts on the network.

  Note:
  - The difference to the use of NAT with IP address pooling is that the mapping of public and private IP addresses is static.

Supporting migration between network service providers

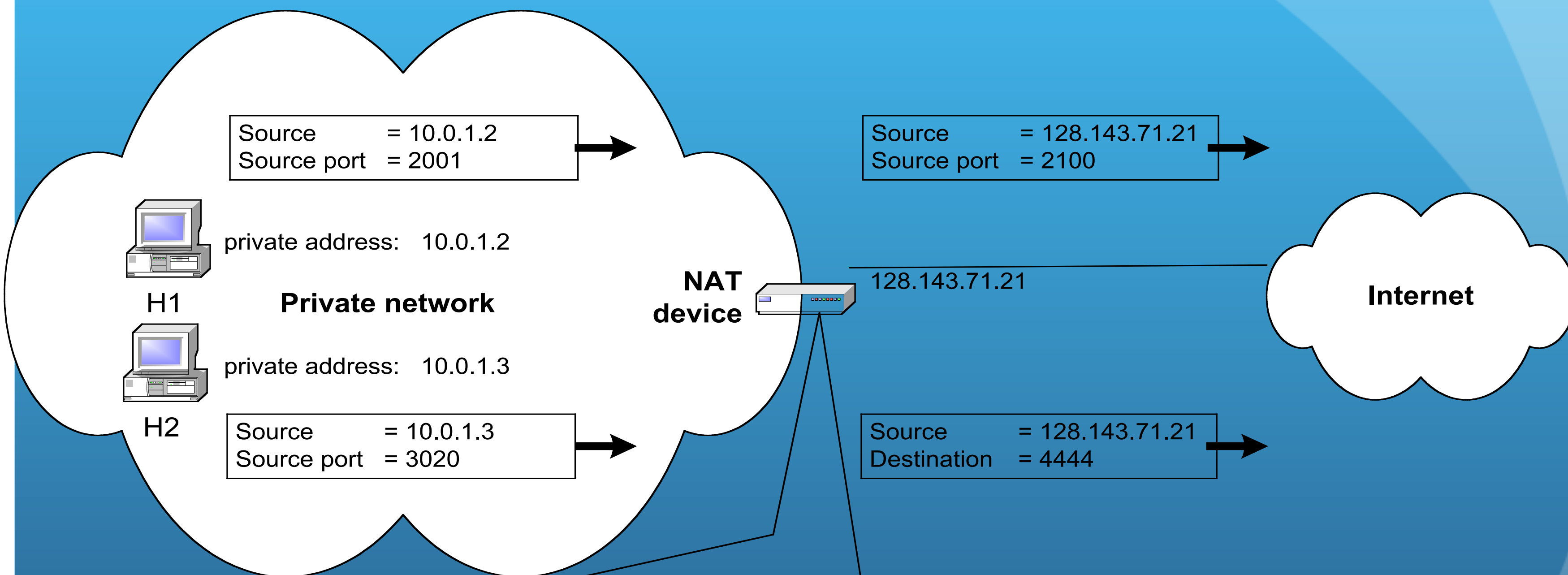


Private Address	Public Address
10.0.1.2	~~128.143.71.21~~ 128.195.4.120

IP masquerading

- **Also called: Network address and port translation (NAPT), port address translation (PAT).**
- **Scenario:** Single public IP address is mapped to multiple hosts in a private network.
- **NAT solution:**
  - Assign private addresses to the hosts of the corporate network
  - NAT device modifies the port numbers for outgoing traffic

IP masquerading

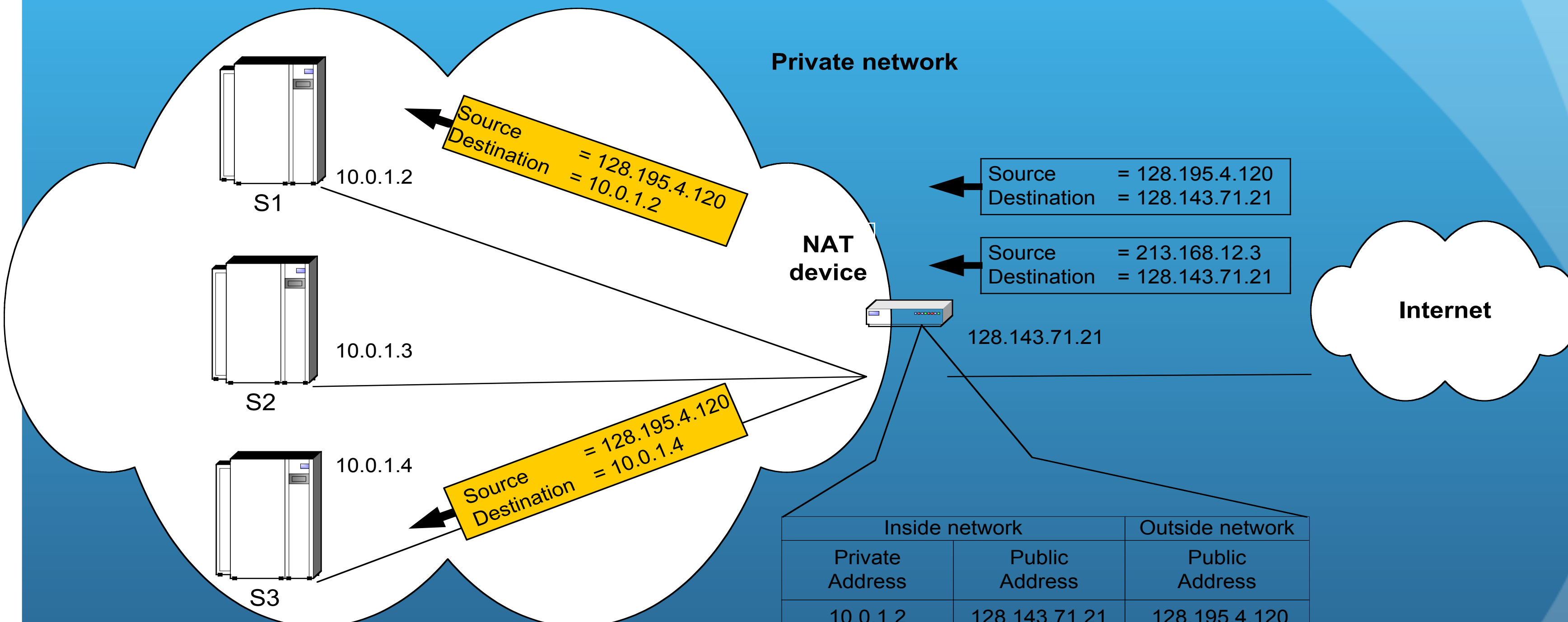


Private Address	Public Address
10.0.1.2/2001	128.143.71.21/2100
10.0.1.3/3020	128.143.71.21/4444

Load balancing of servers

- **Scenario:** Balance the load on a set of identical servers, which are accessible from a single IP address
- **NAT solution:**
  - Here, the servers are assigned private addresses
  - NAT device acts as a proxy for requests to the server from the public network
  - The NAT device changes the destination IP address of arriving packets to one of the private addresses for a server
  - A sensible strategy for balancing the load of the servers is to assign the addresses of the servers in a round-robin fashion.

Load balancing of servers



Inside network		Outside network
Private Address	Public Address	Public Address
10.0.1.2	128.143.71.21	128.195.4.120
10.0.1.4	128.143.71.21	213.168.12.3

Concerns about NAT

- **Performance:**
  - Modifying the IP header by changing the IP address requires that NAT boxes recalculate the IP header checksum
  - Modifying port number and IP address requires that NAT boxes recalculate TCP and UDP checksum (pseudo header)
- **Fragmentation**
  - Care must be taken that a datagram that is fragmented before it reaches the NAT device, is not assigned a different IP address or different port numbers for each of the fragments.

Concerns about NAT

- **End-to-end connectivity:**
  - NAT destroys universal end-to-end reachability of hosts on the Internet.
  - A host in the public Internet often cannot initiate communication to a host in a private network.
  - The problem is worse, when two hosts that are in a private network need to communicate with each other.

Concerns about NAT

- **IP address in application data:**
  - Applications that carry IP addresses in the payload of the application data generally do not work across a private-public network boundary.
  - Some NAT devices inspect the payload of widely used application layer protocols and, if an IP address is detected in the application-layer header or the application payload, translate the address according to the address translation table.

Overview of DHCP

- DHCP – Dynamic Host Control Protocol

Dynamic Assignment of IP addresses

- Dynamic assignment of IP addresses is desirable for several reasons:
  - IP addresses are assigned on-demand
  - Avoid manual IP configuration
  - Support mobility of laptops

DHCP Message Types

Value	Message Type
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNAK
7	DHCPRELEASE
8	DHCPINFORM

Message Types

- DHCPDISCOVER: Broadcast by a client to find available DHCP servers.
- DHCPOFFER: Response from a server to a DHCPDISCOVER and offering IP address and other parameters.
- DHCPREQUEST: Message from a client to servers that does one of the following:
  - Requests the parameters offered by one of the servers and declines all other offers.
  - Verifies a previously allocated address after a system or network change (a reboot for example).
  - Requests the extension of a lease on a particular address.

Contd.

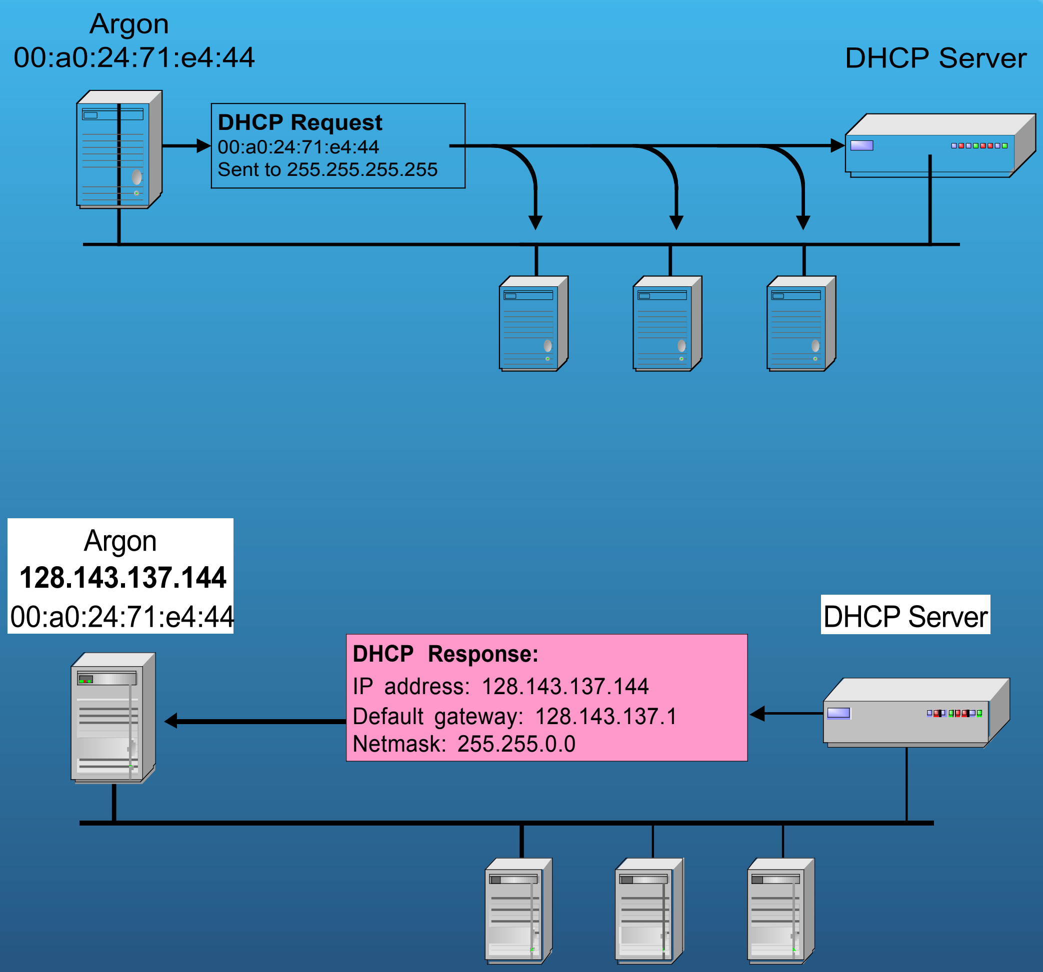
- DHCPACK: Acknowledgement from server to client with parameters, including IP address.
- DHCPNACK: Negative acknowledgement from server to client, indicating that the client's lease has expired or that a requested IP address is incorrect.
- DHCPDECLINE: Message from client to server indicating that the offered address is already in use.
- DHCPRELEASE: Message from client to server canceling remainder of a lease and relinquishing network address.
- DHCPINFORM: Message from a client that already has an IP address (manually configured for example), requesting further configuration parameters from the DHCP server.

Client Server Interactions

- The client broadcasts a DHCPDISCOVER message on its local physical subnet.
  - The DHCPDISCOVER message may include some options such as network address suggestion or lease duration.

- Each server may respond with a DHCPOFFER message that includes an available network address (your IP address) and other configuration options.
  - The servers record the address as offered to the client to prevent the same address being offered to other clients in the event of further DHCPDISCOVER messages being received before the first client has completed its configuration.

DHCP Interaction (simplified)

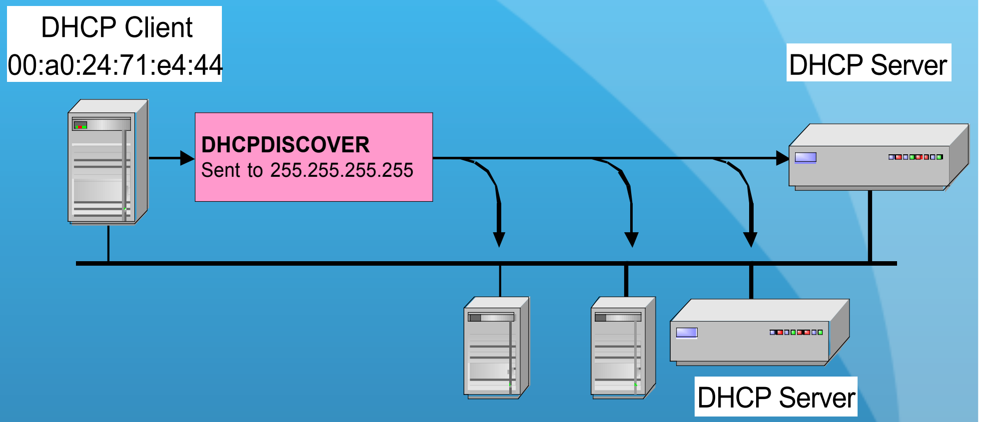


Contd.

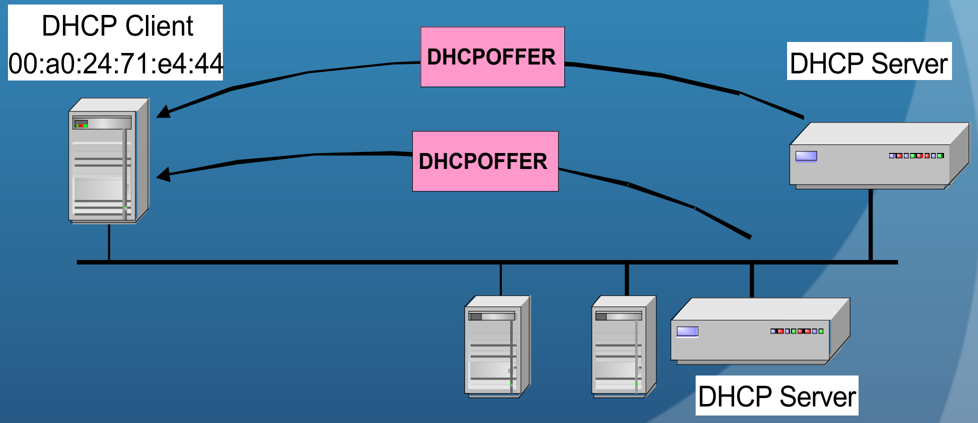
- T\If the client receives one or more DHCPOFFER messages from one or more servers.
  - The client chooses one based on the configuration parameters offered and broadcasts a DHCPREQUEST message that includes the server identifier option to indicate which message it has selected and the requested IP address option, taken from your IP address in the selected offer.
  - In the event that no offers are received, if the client has knowledge of a previous network address, the client may reuse that address if its lease is still valid, until the lease expires.

DHCP Operation

- DCHP DISCOVER



- DCHP OFFER

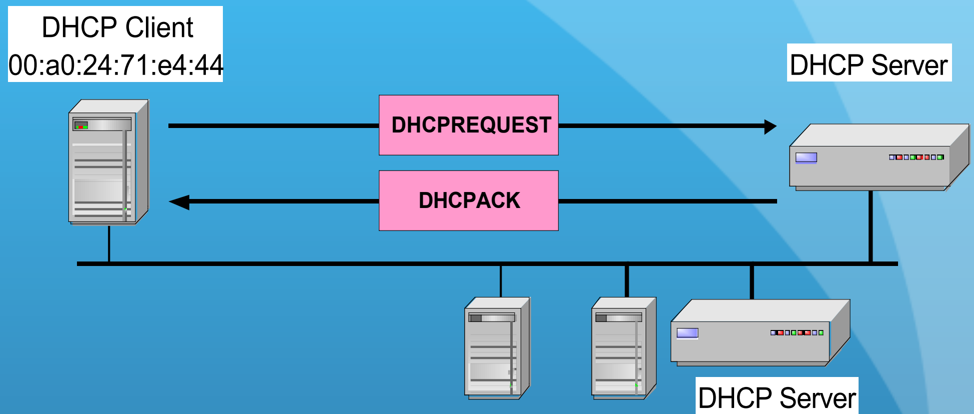


Contd.

- The servers receive the DHCPREQUEST broadcast from the client.
  - Those servers not selected by the DHCPREQUEST message use the message as notification that the client has declined that server's offer.
  - The server selected in the DHCPREQUEST message commits the binding for the client to persistent storage and responds with a DHCPACK message containing the configuration parameters for the requesting client.

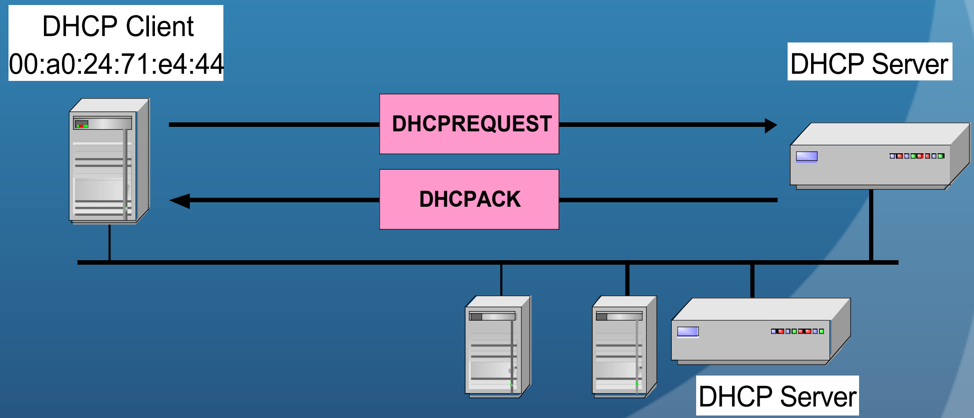
DHCP Operation

- DCHP REQUEST



At this time, the DHCP client can start to use the IP address

- Renewing a Lease (sent when 50% of lease has expired)
- If DHCP server sends DHCPNACK, then address is released when timer expires



Contd.

- The combination of client hardware and assigned network address constitute a unique identifier for the client's lease and are used by both the client and server to identify a lease referred to in any DHCP messages.
- The *your IP* address field in the DHCPACK messages is filled in with the selected network address.

Contd.

- The client receives the DHCPACK message with configuration parameters.
  - The client performs a final check on the parameters, for example with ARP for allocated network address, and notes the duration of the lease and the lease identification cookie specified in the DHCPACK message. At this point, the client is configured.
  - If the client detects a problem with the parameters in the DHCPACK message (the address is already in use on the network, for example), the client sends a DHCPDECLINE message to the server and restarts the configuration process.

Contd.

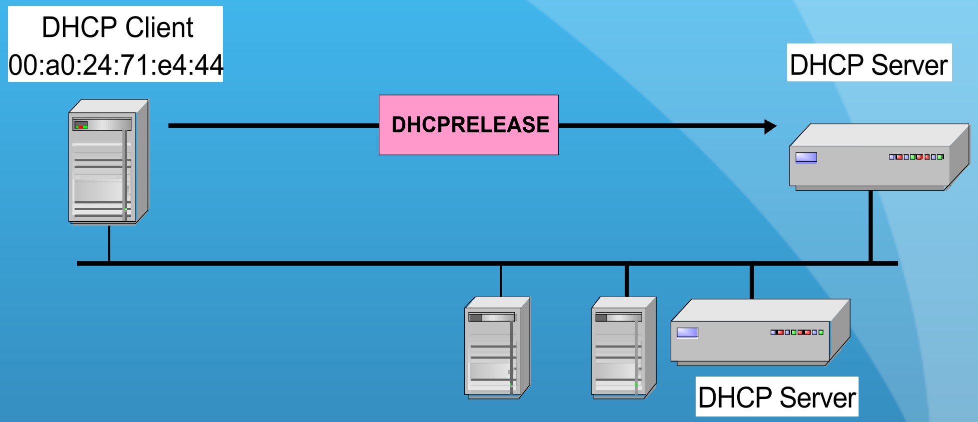
- The client should wait a minimum of ten seconds before restarting the configuration process to avoid excessive network traffic in case of looping.
- On receipt of a DHCPDECLINE, the server must mark the offered address as unavailable (and possibly inform the system administrator that there is a configuration problem).
- If the client receives a DHCPNAK message, the client restarts the configuration process.

Contd.

- The client may choose to relinquish its lease on a network address by sending a DHCPRELEASE message to the server.
- The client identifies the lease to be released by including its network address and its hardware address.

DHCP Operation

- DCHP RELEASE



At this time, the DHCP client has released the IP address

Lease Renewal

- When a server sends the DHCPACK to a client with IP address and configuration parameters, it also registers the start of the lease time for that address.
- This lease time is passed to the client as one of the options in the DHCPACK message, together with two timer values, T1 and T2.
- The client is rightfully entitled to use the given address for the duration of the lease time.

Contd.

- On applying the receive configuration, the client also starts the timers T1 and T2. At this time, the client is in the BOUND state.
- Times T1 and T2 are options configurable by the server but T1 must be less than T2, and T2 must be less than the lease time.
- According to RFC 2132, T1 defaults to (0.5 * lease time) and T2 defaults to (0.875 * lease time).

Contd.

- When timer T1 expires, the client will send a DHCPREQUEST (unicast) to the server that offered the address, asking to extend the lease for the given configuration. The client is now in the RENEWING state

- The server would usually respond with a DHCPACK message indicating the new lease time, and timers T1 and T2 are reset at the client accordingly.

- The server also resets its record of the lease time.

- Under normal circumstances, an active client would continually renew its lease in this way indefinitely, without the lease ever expiring.

Contd.

- If no DHCPACK is received until timer T2 expires, the client enters the REBINDING state.
- Client now broadcasts a DHCPREQUEST message to extend its lease.
- This request can be confirmed by a DHCPACK message from any DHCP server on the network.

Contd.

- If the client does not receive a DHCPACK message after its lease has expired, it has to stop using its current TCP/IP configuration.
- The client may then return to the INIT state, issuing a DHCPDISCOVER broadcast to try and obtain any valid address.

DHCP Pros

- It relieves the network administrator of a great deal of manual configuration work.
- The ability for a device to be moved from network to network and to automatically obtain valid configuration parameters for the current network can be of great benefit to mobile users.
- Because IP addresses are only allocated when clients are actually active, it is possible, by the use of reasonably short lease times and the fact that mobile clients do not need to be allocated more than one address, to reduce the total number of addresses in use in an organization.

DHCP Cons

- Uses UDP, an unreliable and insecure protocol.
- DNS cannot be used for DHCP configured hosts.

Overview of DNS

DNS – Domain Name Service

Outline

- What is DNS?
- What services does it provide?
- How does it operate?
- Message format
- Types of messages

What is DNS?

- DNS is a host name to IP address translation service
- DNS is
  - a distributed database implemented in a hierarchy of name servers
  - an application level protocol for message exchange between clients and servers

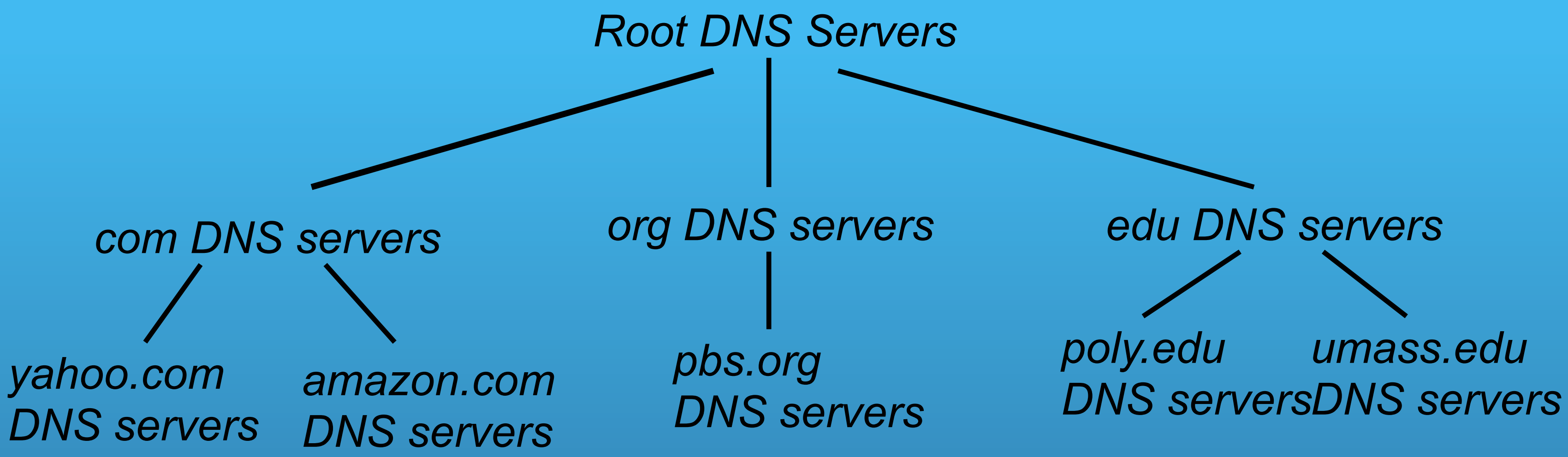
Why DNS?

- It is easier to remember a host name than it is to remember an IP address.
- A name has more meaning to a user than a 4 byte number.
- Applications such as FTP, HTTP, email, etc., all require the user to input a destination.
- The user generally enters a host name.
- The application takes the host name supplied by the user and forwards it to DNS for translation to an IP address.

How does it work?

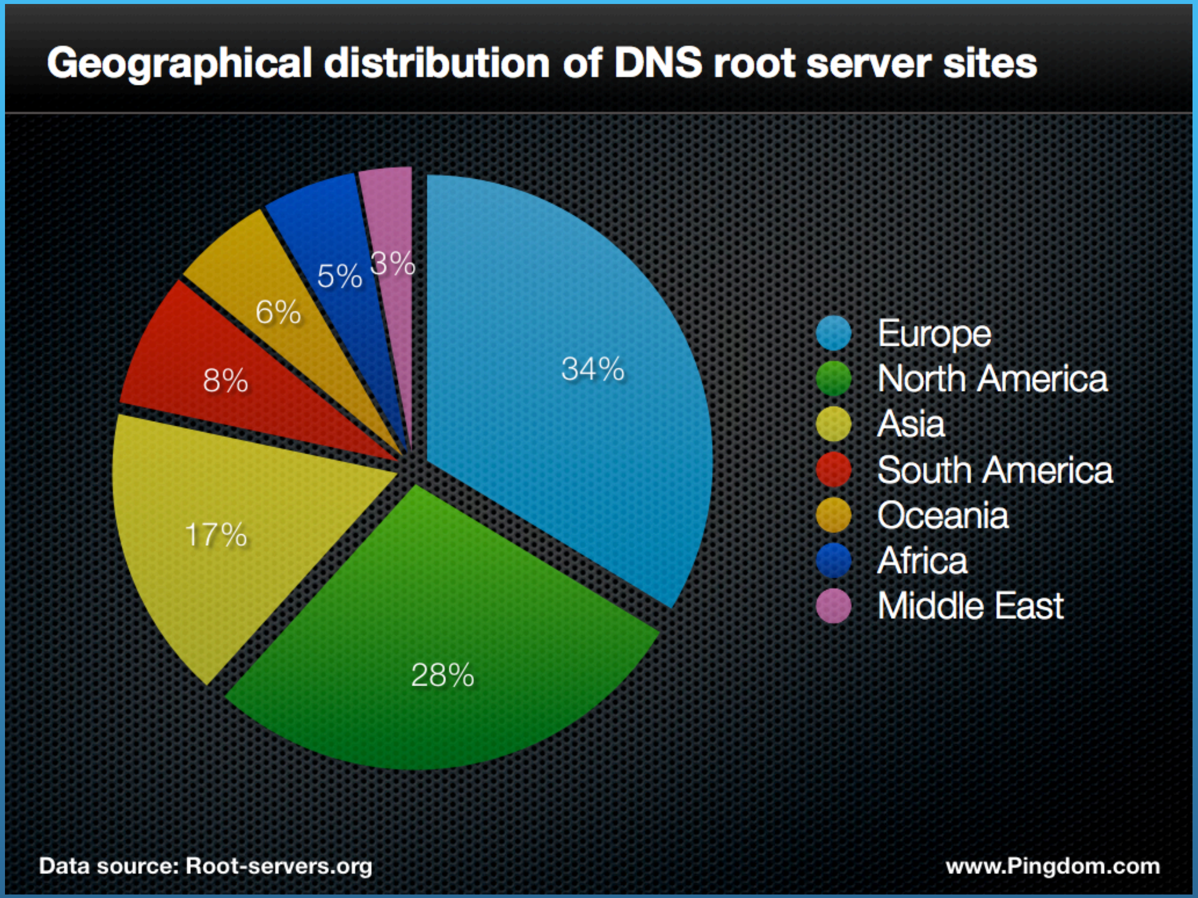
- DNS works by exchanging messages between client and server machines.
- A client application will pass the destination host name to the DNS process (in Unix referred to as the gethostbyname() routine) to get the IP address.
- The application then sits and waits for the response to return.

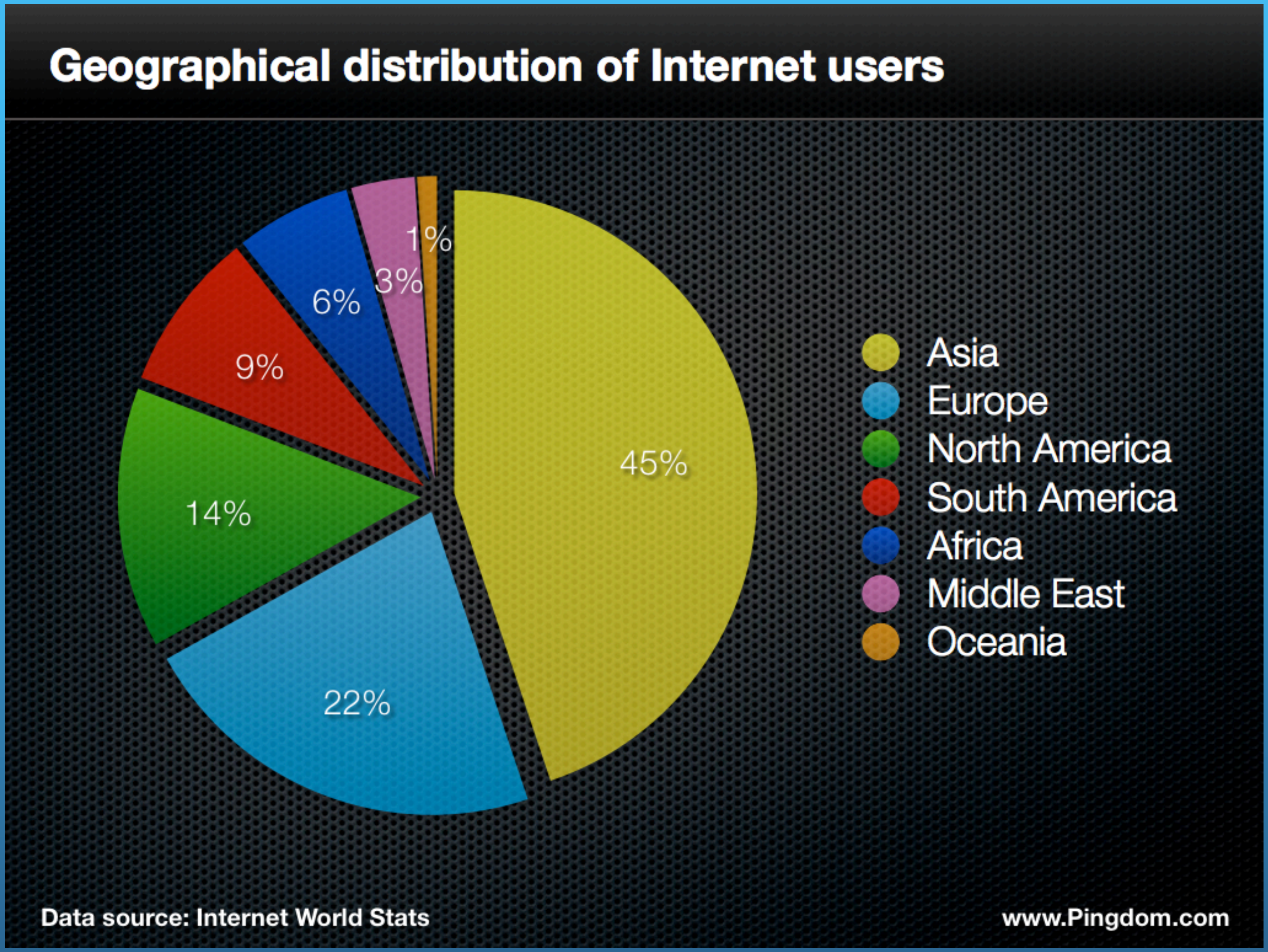
Distributed, Hierarchical Database



Client wants IP for www.amazon.com; 1st approx:

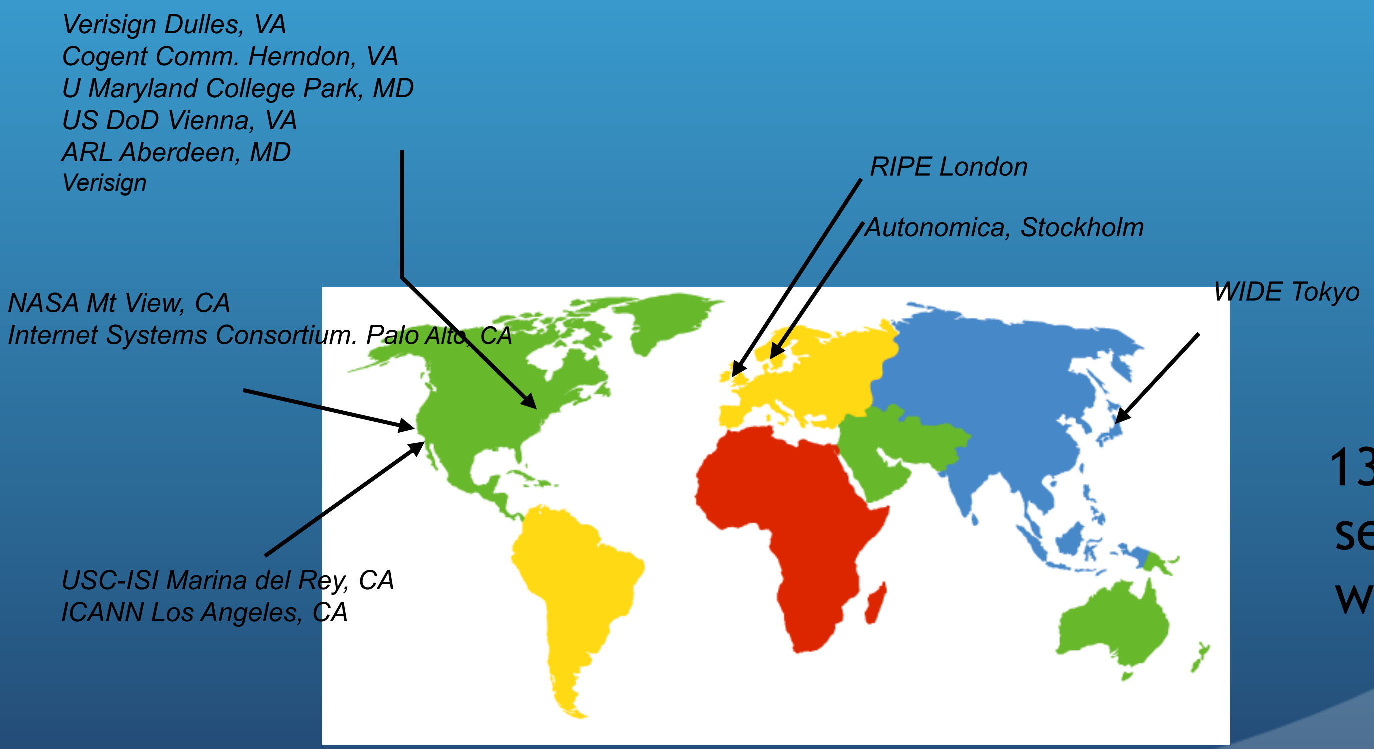
- client queries a root server to find "com" DNS server
- client queries "com" DNS server to get "amazon.com" DNS server
- client queries "amazon.com" DNS server to get IP address for "www.amazon.com"


Geographical distribution of DNS root server sites
34%
28%
17%
8%
6%
5%
3%
Europe
North America
Asia
South America
Oceania
Africa
Middle East
Data source: Root-servers.org
www.Pingdom.com


Geographical distribution of Internet users
1%
3%
6%
9%
45%
14%
22%
Asia
Europe
North America
South America
Africa
Middle East
Oceania
Data source: Internet World Stats
www.Pingdom.com

DNS: Root name servers

- contacted by local name server that cannot resolve name
- root name server:
  - contacts authoritative name server if name mapping not known
  - gets mapping
  - returns mapping to local name server



13 root name server operators worldwide

TLD and Authoritative Servers

- Top-level domain (TLD) servers:
  - responsible for com, org, net, edu, etc, and all top-level country domains uk, fr, ca, jp.
  - 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

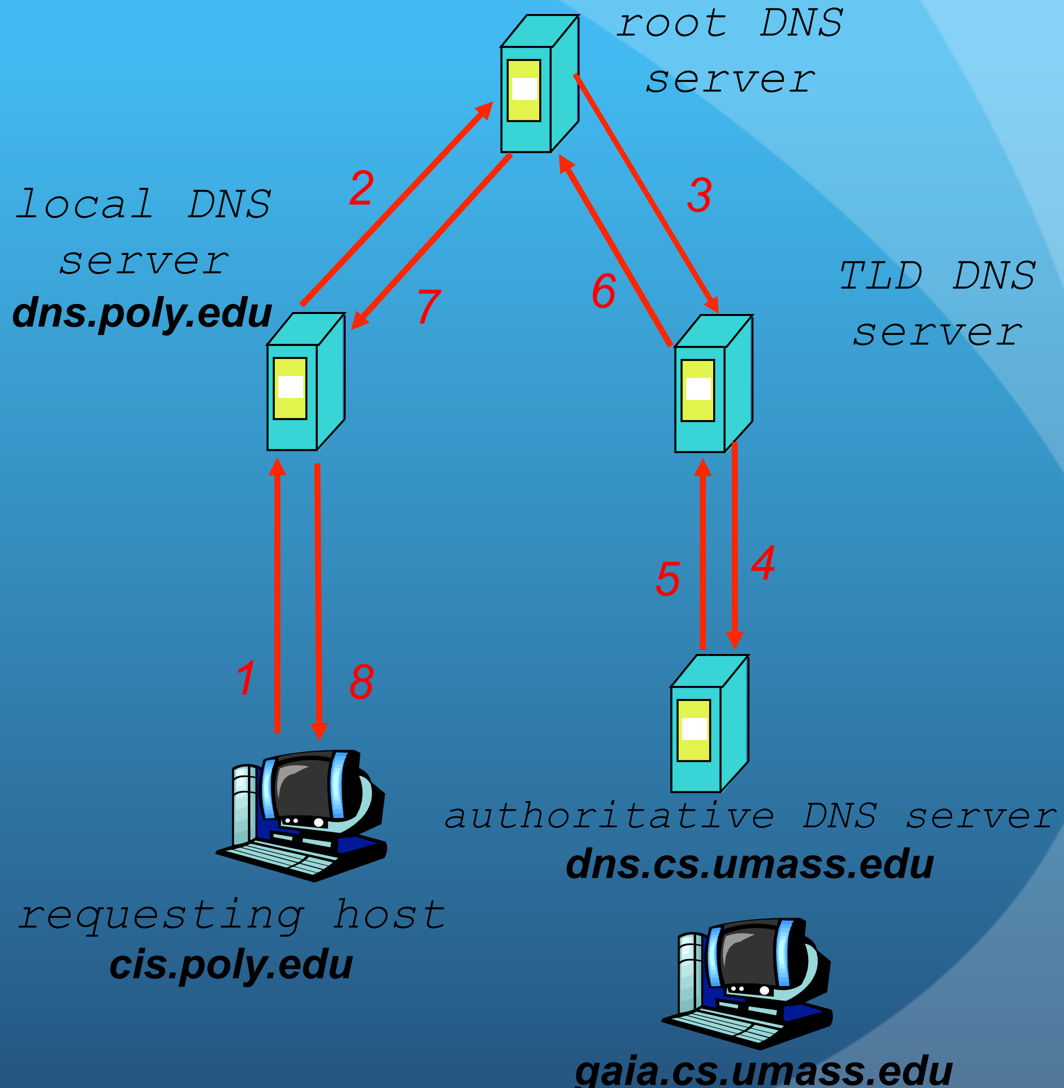
- does not strictly belong to hierarchy
- each ISP (residential ISP, company, university) has one.
  - also called “default name server”
- when host makes DNS query, query is sent to its local DNS server
  - acts as proxy, forwards query into hierarchy

DNS Queries

- Recursive:
  - The client machine sends a request to the local name server, which, if it does not find the address in its database, sends a request to the root name server, which, in turn, will route the query to an intermediate or authoritative name server. Note that the root name server can contain some hostname to IP address mappings. The intermediate name server always knows who the authoritative name server is.

DNS name resolution example

- Host at cis.poly.edu wants IP address for: gaia.cs.umass.edu



recursive query:

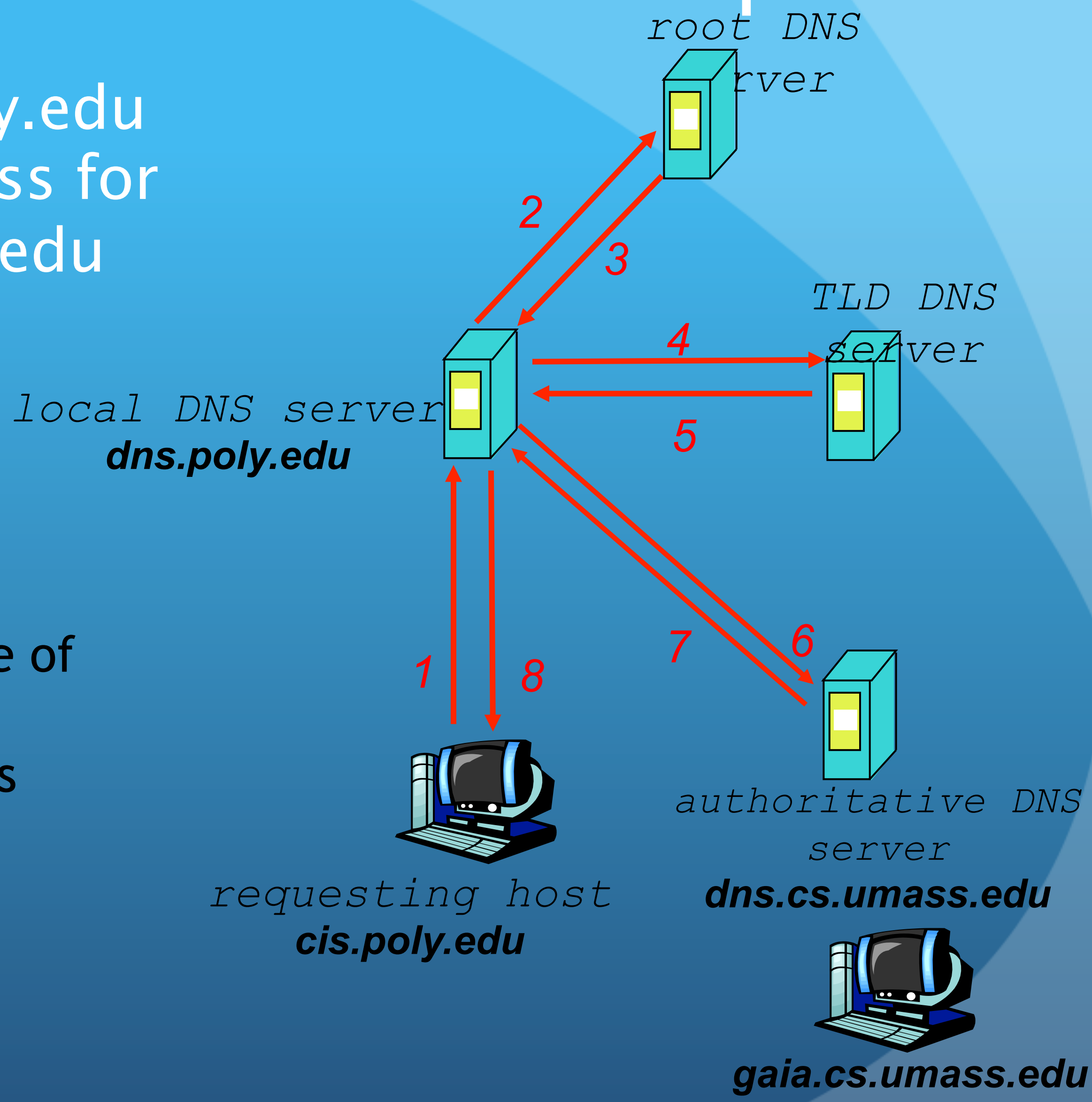
- ❐ puts burden of name resolution on contacted root name server
- ❐ heavy load

DNS Queries (cont’d)

- Iterative:
  - The local server queries the root server. If address not in its database, will have the name/address of an intermediate or authoritative name server and forward that information to the local name server so that it can directly communicate with the intermediate or authoritative name server. This is to prevent the overloading of the root servers that handle millions of requests.

DNS name resolution example

- Host at cis.poly.edu wants IP address for gaia.cs.umass.edu



<u>iterated query:</u>

❒ contacted server replies with name of server to contact

-> "I don't know this name, but ask this server"

DNS: caching and updating records

- 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
- update/notify mechanisms under design by IETF
  - RFC 2136
  - http://www.ietf.org/html.charters/dnsind-charter.html

Operation of DNS

- The DNS data is stored in the database in the form of resource records (RR). The RRs are directly inserted in the DNS messages.
- The RRs are a 4 tuple that consist of: {name, value, type, TTL}.

RRs

- TTL: time to live, used to indicate when an RR can be removed from the DNS cache.
- Type =
  - A – then NAME is a hostname and Value its IP address
  - NS – then NAME is a domain name and Value is the IP address of an authoritative name server
  - CNAME – then NAME is an alias for a host and Value is the canonical name for the host
  - MX – then NAME is an alias for an email host and Value is the the canonical name for the email server

DNS records

DNS: distributed db storing resource records (RR)

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

- Type=A
  - **name** is hostname
  - **value** is IP address
- Type=NS
  - **name** is domain (eg., foo.com)
  - **value** is hostname of authoritative name server for this domain
- Type=CNAME
  - **name** is alias name for some "canonical" (the real) name, eg., www.ibm.com is really servereast.backup2.ibm.com
  - **value** is canonical name
- Type=MX
  - **value** is name of mailserver associated with **name**

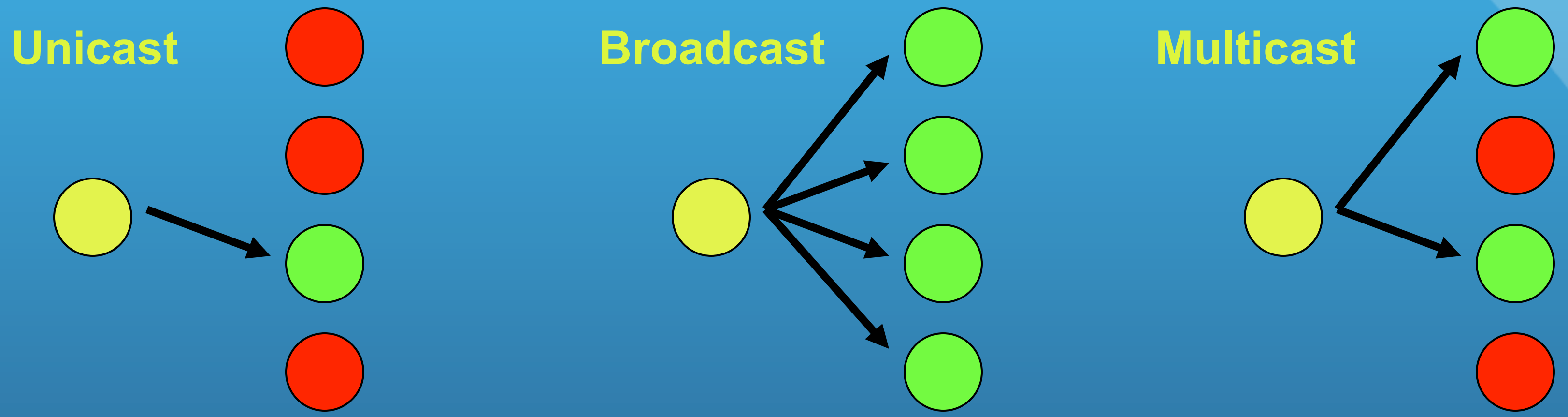
Summary

- DNS provides a mechanism for maintaining the user friendliness of the Internet by hiding some of the operational details.
- DNS servers have to be created manually. Recently an update protocol was introduced that allows DNS to exchange data for additions and deletions.

IP Multicasting

Multicasting

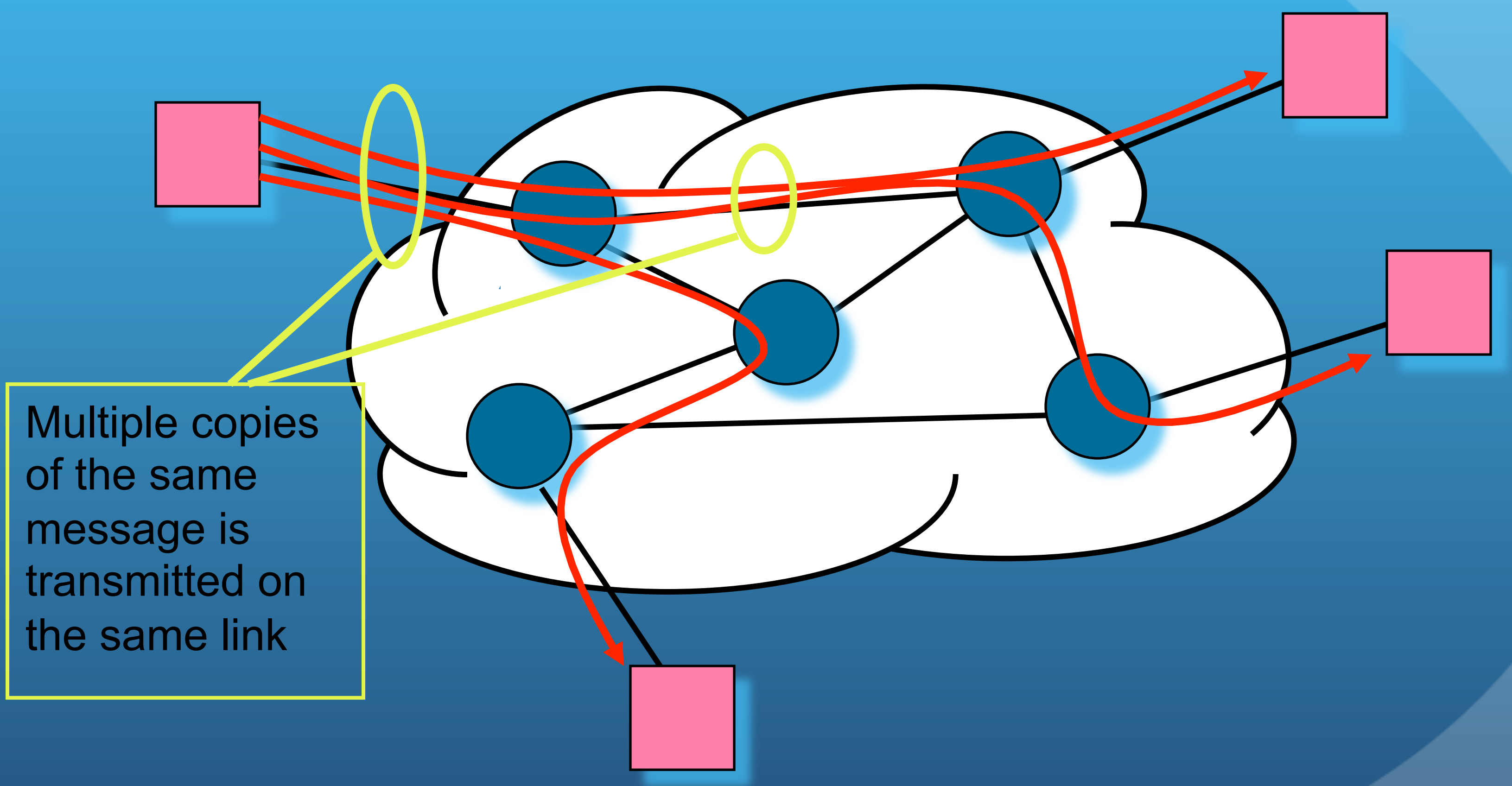
- Multicast communications refers to one-to-many or many-to-many communications.



IP Multicasting refers to the implementation of multicast communication in the Internet

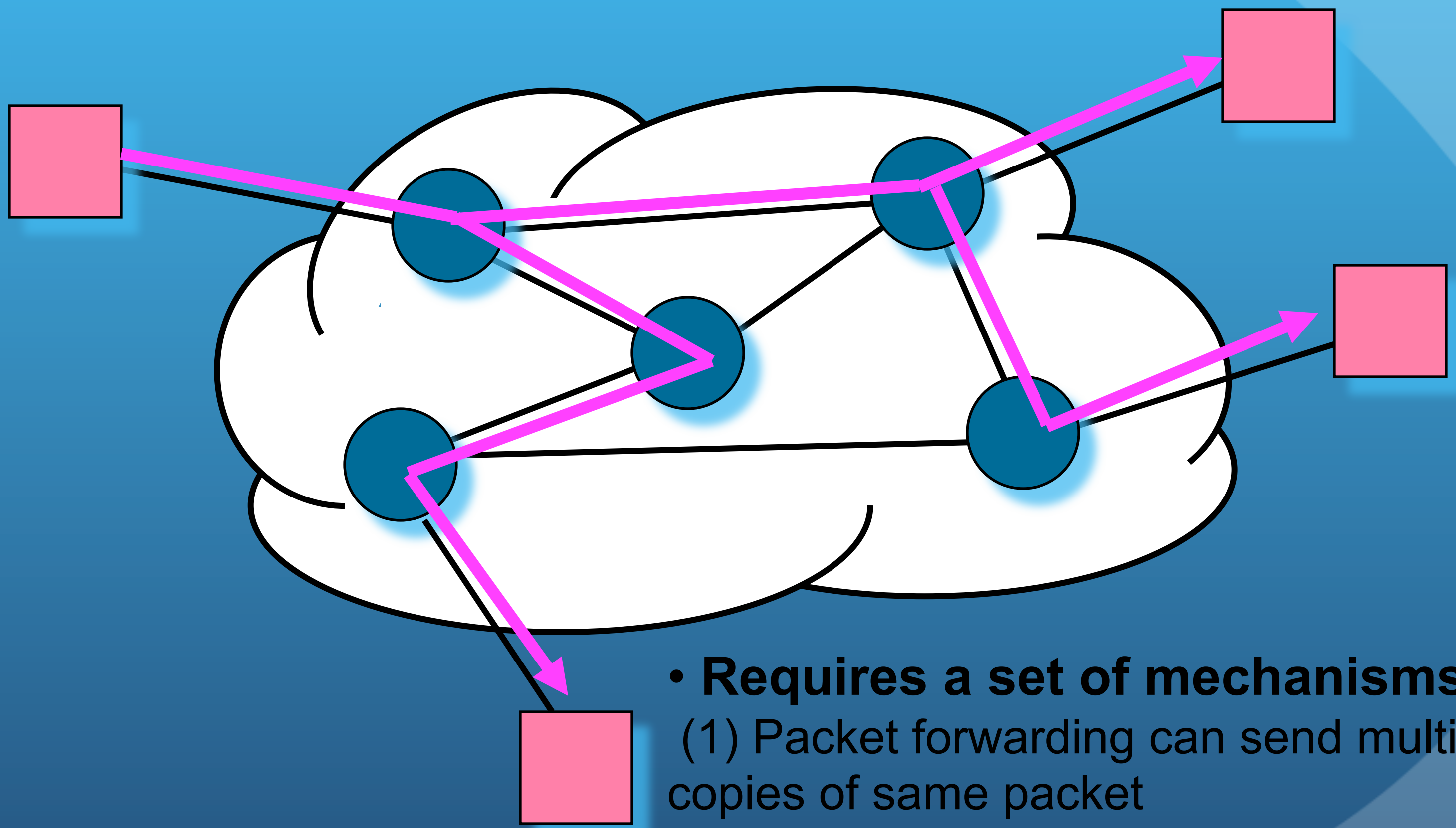
Multicasting over a Packet Network

- Without support for multicast at the network layer:



Multicasting over a Packet Network

- With support for multicast at the network layer:



- **Requires a set of mechanisms:**
 (1) Packet forwarding can send multiple copies of same packet
 (2) Multicast routing algorithm which builds a spanning tree (dynamically)

Semantics of IP Multicast

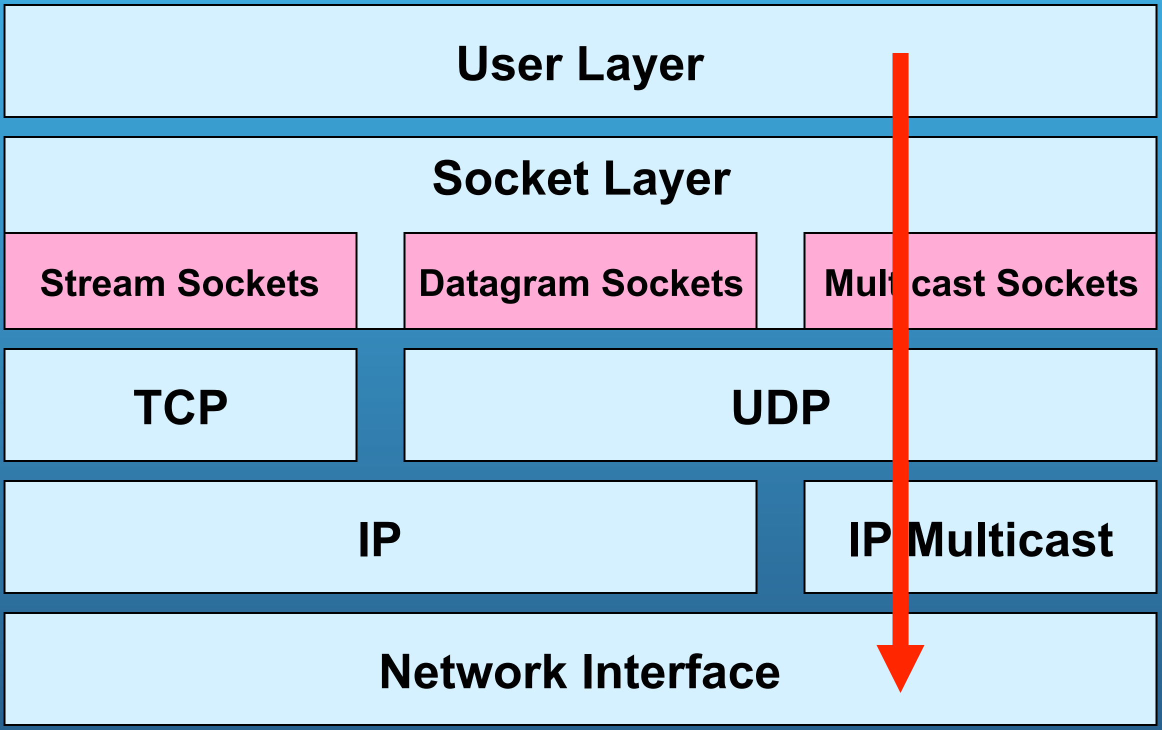
IP multicast works as follows:

- Multicast groups are identified by IP addresses in the range 224.0.0.0 - 239.255.255.255 (class D address)
- Every host (*more precisely*: interface) can join and leave a multicast group dynamically
    - no access control
- Every IP datagram sent to a multicast group is transmitted to all members of the group

- The IP Multicast service is unreliable

The IP Protocol Stack

- IP Multicasting only supports UDP as higher layer
- There is no multicast TCP !



Multicast Addressing

- All Class D addresses are multicast addresses:

Class D

1	1	1	0	multicast group id

28 bits

Class	From	To
D	224.0.0.0	239.255.255.255

- Multicast addresses are dynamically assigned.
- An IP datagram sent to a multicast address is forwarded to everyone who has joined the multicast group
- If an application is terminated, the multicast address is (implicitly) released.

Types of Multicast addresses

- The range of addresses between 224.0.0.0 and 224.0.0.255, inclusive, is reserved for the use of routing protocols and other low-level topology discovery or maintenance protocols
- Multicast routers should not forward any multicast datagram with destination addresses in this range.
- Examples of special and reserved Class D addresses:

224.0.0.1	All systems on this subnet
224.0.0.2	All routers on this subnet
224.0.1.1	NTP (Network Time Protocol)
224.0.0.9	RIP-2 (a routing protocol)

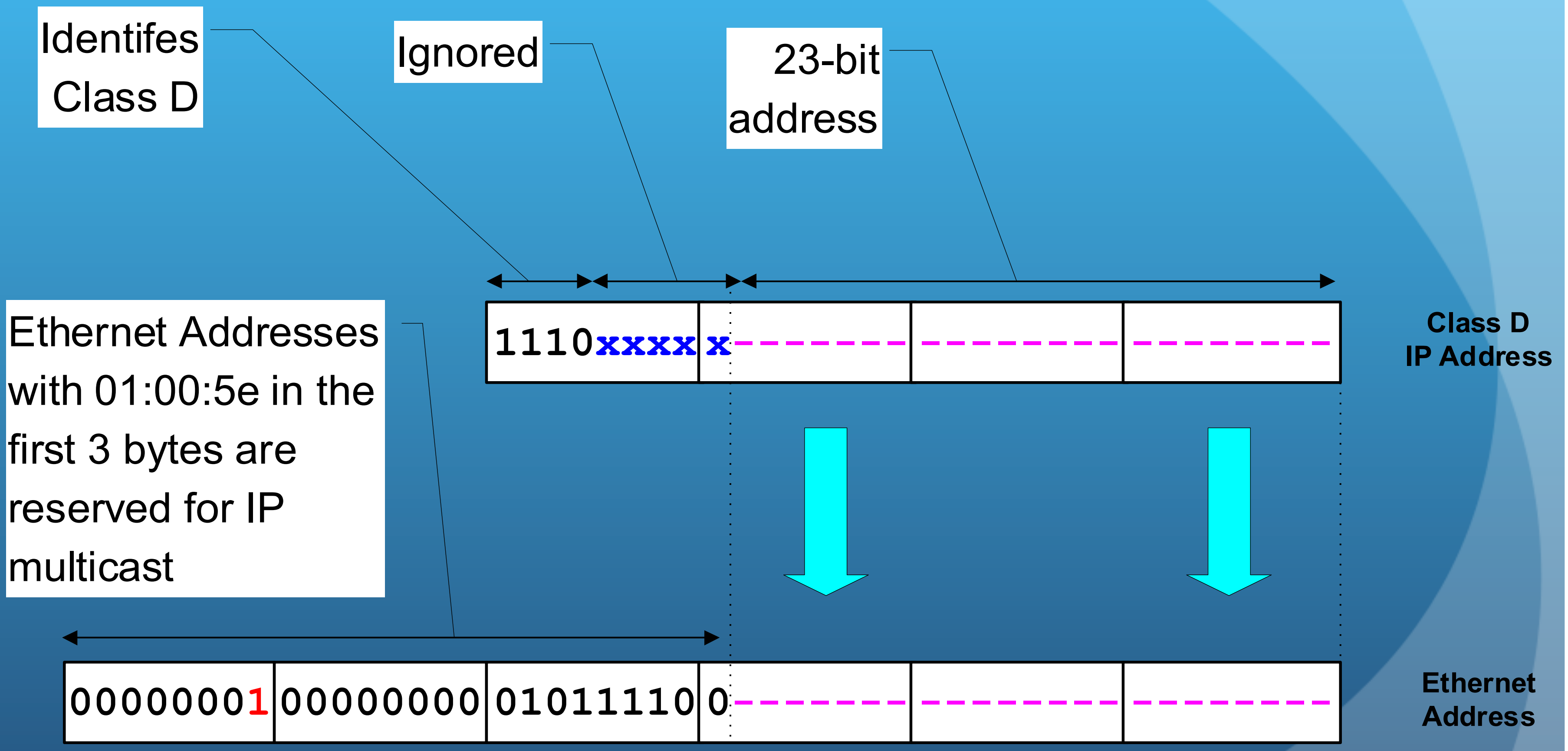
Multicast Address Translation

- In Ethernet MAC addresses, a multicast address is identified by setting the lowest bit of the "most left byte"

-------1	--------	--------	--------	--------	--------

Not all Ethernet cards can filter multicast addresses in hardware
Then: Filtering is done in software by device driver.

Multicast Address Mapping



IGMP

- The **Internet Group Management Protocol (IGMP)** is a simple protocol for the support of IP multicast.
- IGMP is defined in RFC 1112.
- IGMP operates on a physical network (e.g., single Ethernet Segment.
- IGMP is used by multicast routers to keep track of membership in a multicast group.
- Support for:
  - Joining a multicast group
  - Query membership
  - Send membership reports