

Lecture 13:

Naming

CSE 123: Computer Networks
Stefan Savage





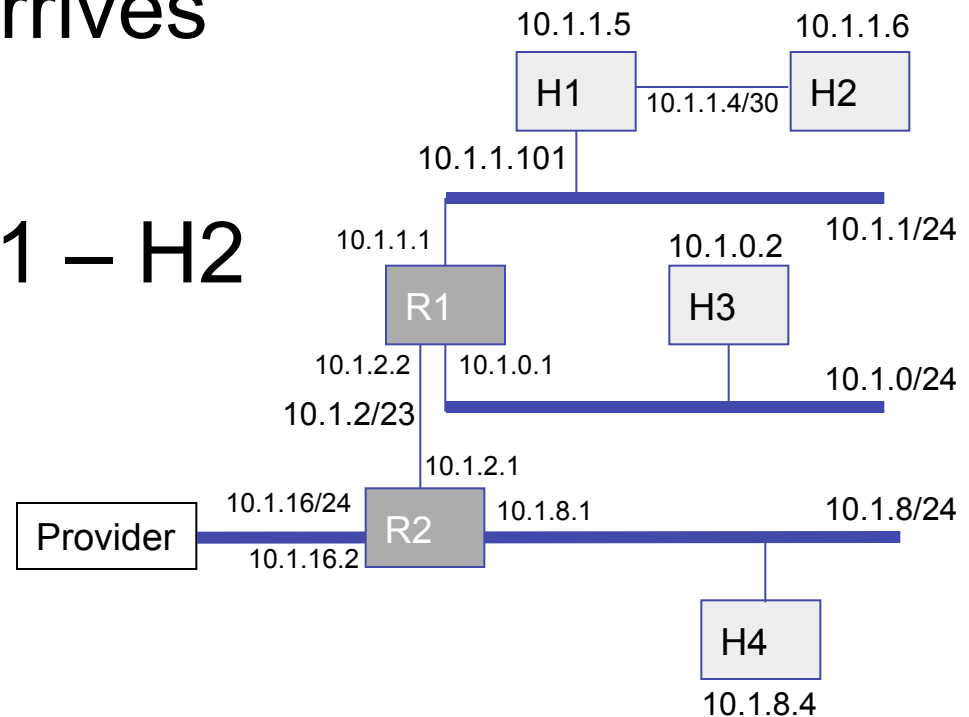
Lecture 13 Overview

- Packet forwarding example
- Discovering addresses (DHCP/ARP)
- User-friendly names (DNS)



Forwarding example

- Packet to 10.1.1.6 arrives
- Path is R2 – R1 – H1 – H2



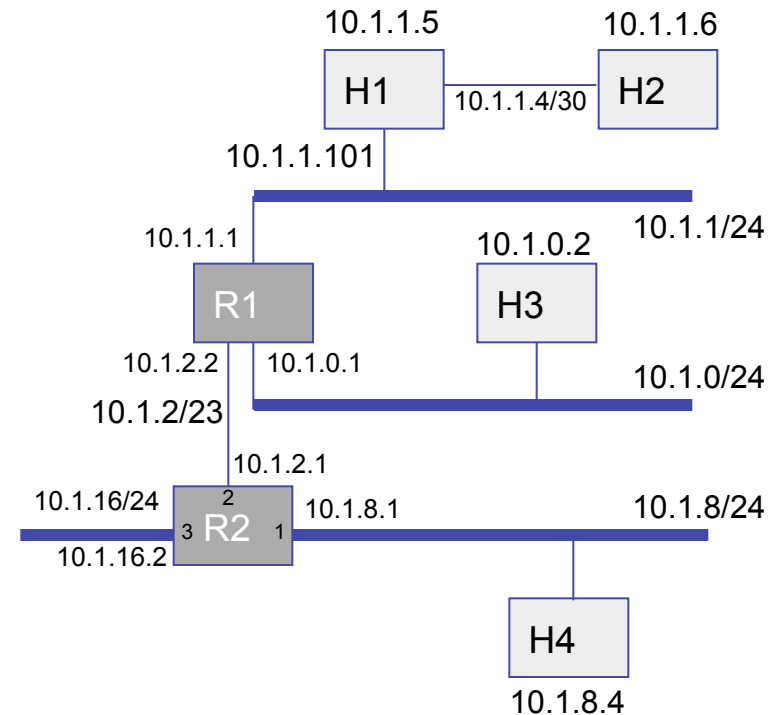


Forwarding example (2)

- Packet to 10.1.1.6
- Matches 10.1.0.0/23

Forwarding table at R2

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.16.1
10.1.8.0/24	interface1
10.1.2.0/23	interface2
10.1.0.0/23	10.1.2.2
10.1.16.0/24	interface3



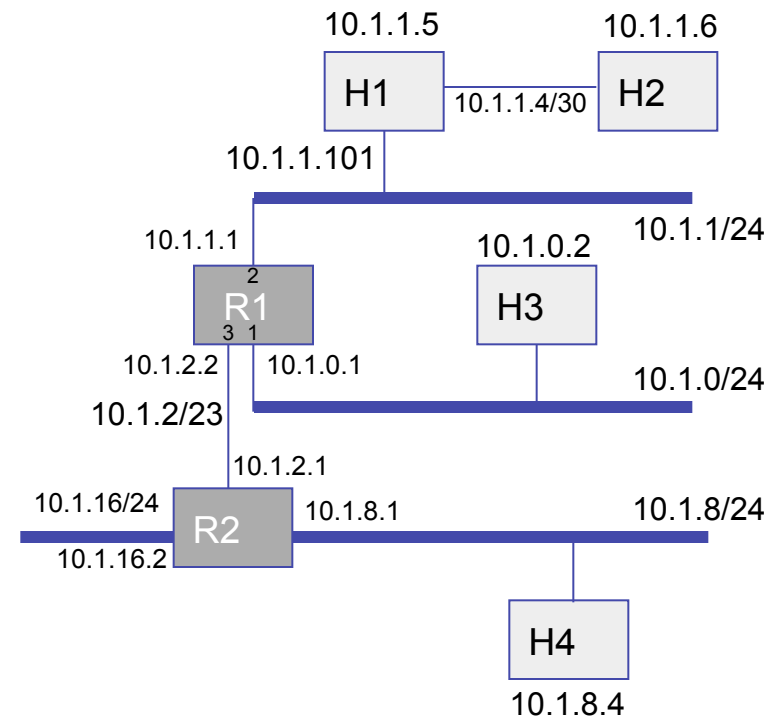


Forwarding example (3)

- Packet to 10.1.1.6
- Matches 10.1.1.4/30
 - Longest prefix match

Routing table at R1

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.2.1
10.1.0.0/24	interface1
10.1.1.0/24	interface2
10.1.2.0/23	interface3
10.1.1.4/30	10.1.1.101



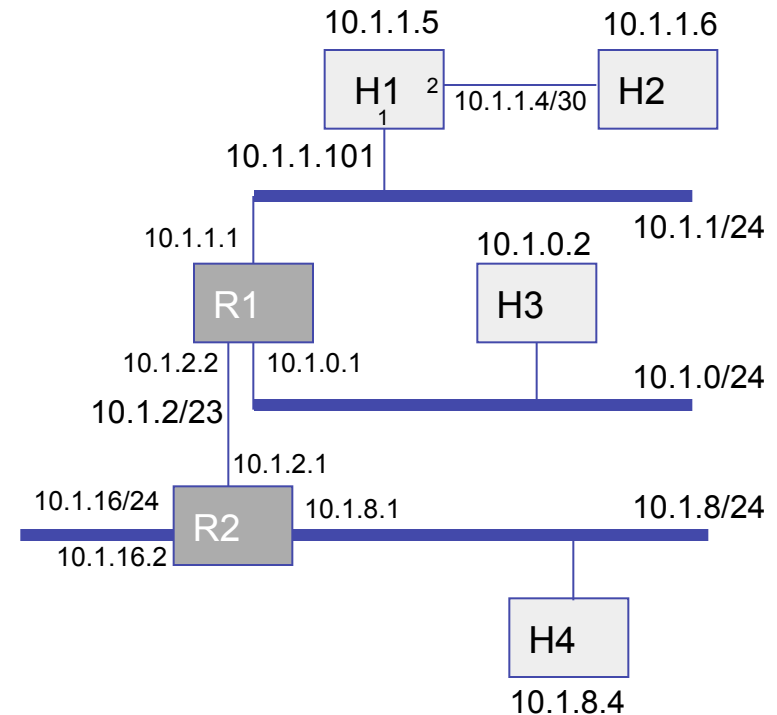


Forwarding example (4)

- Packet to 10.1.1.6
- Direct route
 - Longest prefix match

Routing table at H1

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.1.1
10.1.1.0/24	interface1
10.1.1.4/30	interface2





Layers of Identifiers

- **Host name** (e.g., `www.ucsd.edu`)
 - ◆ Used by *humans* to specify host of interest
 - ◆ Unique, selected by host administrator
 - ◆ Hierarchical, variable-length string of alphanumeric characters
- **IP address** (e.g., `128.54.70.238`)
 - ◆ Used by *routers* to forward packets
 - ◆ Unique, topologically meaningful locator
 - ◆ Hierarchical namespace of 32 bits
- **MAC address** (e.g., `58:B0:35:F2:3C:D9`)
 - ◆ Used by *network adaptors* to identify interesting frames
 - ◆ Unique, hard-coded identifier burned into network adaptor
 - ◆ Flat name space (of 48 bits in Ethernet)



Naming Hierarchy for Scale

- Host name: **www.ucsd.edu**
 - ◆ **Domain**: registrar for each top-level domain (e.g., .edu)
 - ◆ **Host name**: local administrator assigns to each host
- IP addresses: **128.54.70.238**
 - ◆ **Prefixes**: ICANN, regional Internet registries, and ISPs
 - ◆ **Hosts**: static configuration, or dynamic using DHCP
- MAC addresses: **58:B0:35:F2:3C:D9**
 - ◆ **OIDs**: assigned to vendors by the IEEE
 - ◆ **Adapters**: assigned by the vendor from its block

Mapping Between Identifiers



- Domain Name System (**DNS**)
 - ◆ Given a host name, provide the IP address
 - ◆ Given an IP address, provide the host name
- Address Resolution Protocol (**ARP**)
 - ◆ Given an IP address, provide the MAC address
 - ◆ To enable communication within the Local Area Network
- Dynamic Host Configuration Protocol (**DHCP**)
 - ◆ Automates host boot-up process
 - ◆ Given a MAC address, assign a unique IP address
 - ◆ ... and tell host other stuff about the Local Area Network

Address Resolution Protocol



- Every node maintains an ARP table
 - ◆ (IP address, MAC address) pair
- Consult the table when sending a packet
 - ◆ Map destination IP address to MAC address
 - ◆ Encapsulate and transmit the data packet
- What if the IP address is not in the table?
 - ◆ Broadcast: “Who has IP address x.x.x.x?”
 - ◆ Response: “MAC address yy:yy:yy:yy:yy:yy”
 - ◆ Sender caches the result in its ARP table

Recall: Whence come IP Addresses?



- You already have a bunch from the days when you called Jon Postel and asked for them (e.g. BBN)
- You get them from another provider
 - ◆ E.g. buy service from Sprint and get a /24 from one of their address blocks
- You get one directly from a routing registry
 - ◆ ARIN: North America, APNIC (Asia Pacific), RIPE (Europe), LACNIC (Latin America), AFRINIC (Africa)
 - ◆ Registries get address from IANA (Internet Assigned Numbers Authority)

How Do You And I Get One?

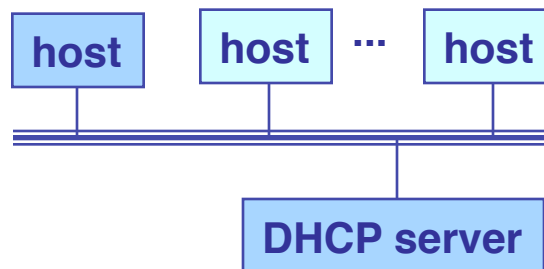


- Well from your provider!
- But how do you know what it is?
- Manual configuration
 - ◆ They tell you and you type that number into your computer (along with the default gateway, DNS server, etc.)
- Automated configuration
 - ◆ Dynamic Host Resolution Protocol (DHCP)



Bootstrapping Problem

- Host doesn't have an IP address yet
 - ◆ So, host doesn't know what source address to use
- Host doesn't know who to ask for an IP address
 - ◆ So, host doesn't know what destination address to use
- Solution: shout to discover a server who can help
 - ◆ Install a special server on the LAN to answer distress calls



DHCP



- Broadcast-based LAN protocol algorithm
 - ◆ Host broadcasts “DHCP discover” on LAN (e.g. Ethernet broadcast)
 - ◆ DHCP server responds with “DHCP offer” message
 - ◆ Host requests IP address: “DHCP request” message
 - ◆ DHCP server sends address: “DHCP ack” message w/IP address
- Easy to have fewer addresses than hosts (e.g. UCSD wireless) and to *renumber* network (use new addresses)
- What if host goes away (how to get address back?)
 - ◆ Address is a “lease” not a “grant”, has a timeout
 - ◆ Host may have different IP addresses at different times?

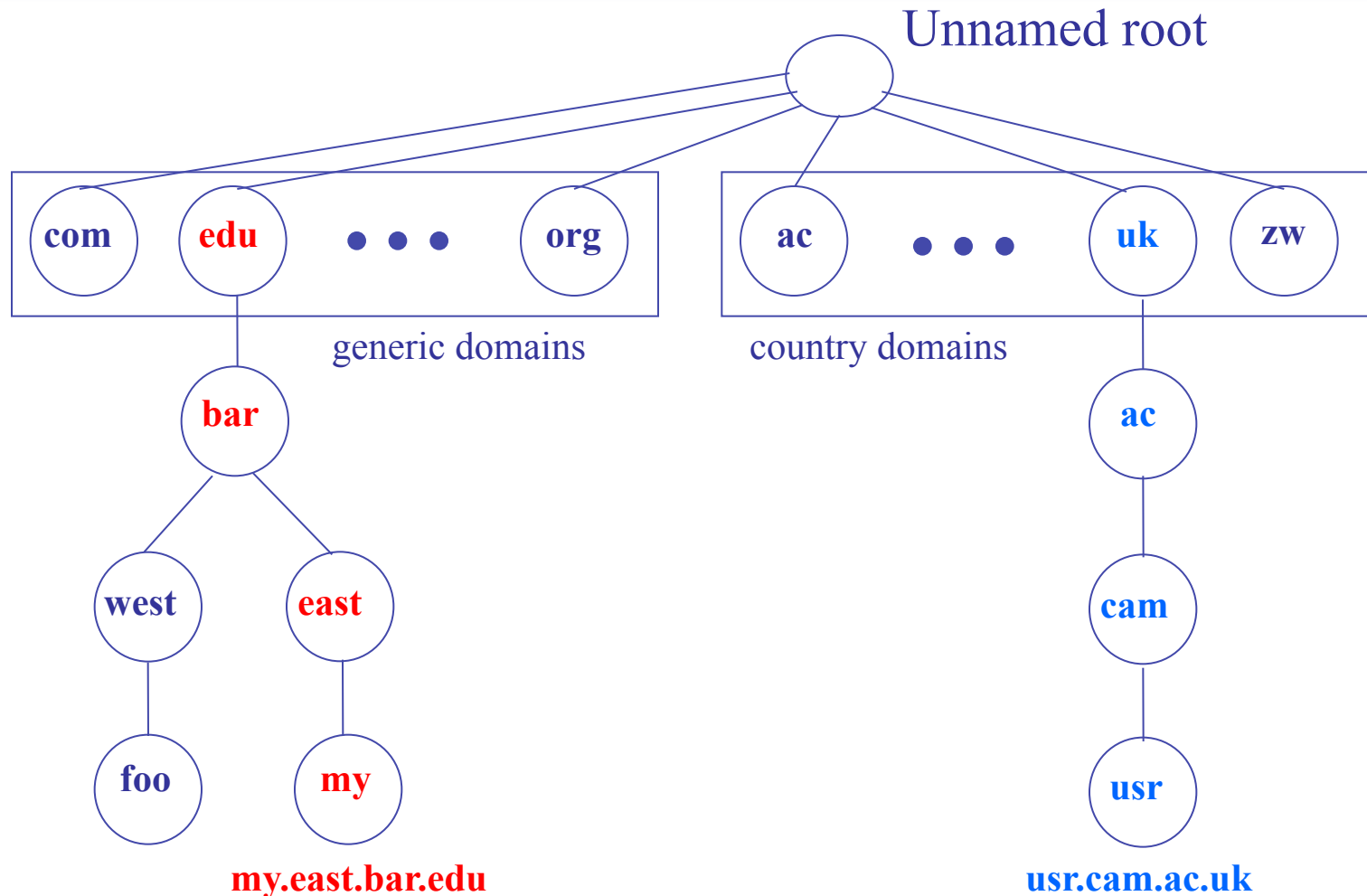
Domain Name System (DNS)



- Distributed administrative control
 - ◆ Hierarchical name space divided into zones
 - ◆ Distributed over a collection of DNS servers
- Hierarchy of DNS servers
 - ◆ Root servers
 - ◆ Top-level domain (TLD) servers
 - ◆ Authoritative DNS servers
- Performing the translations
 - ◆ Local DNS servers
 - ◆ Resolver software



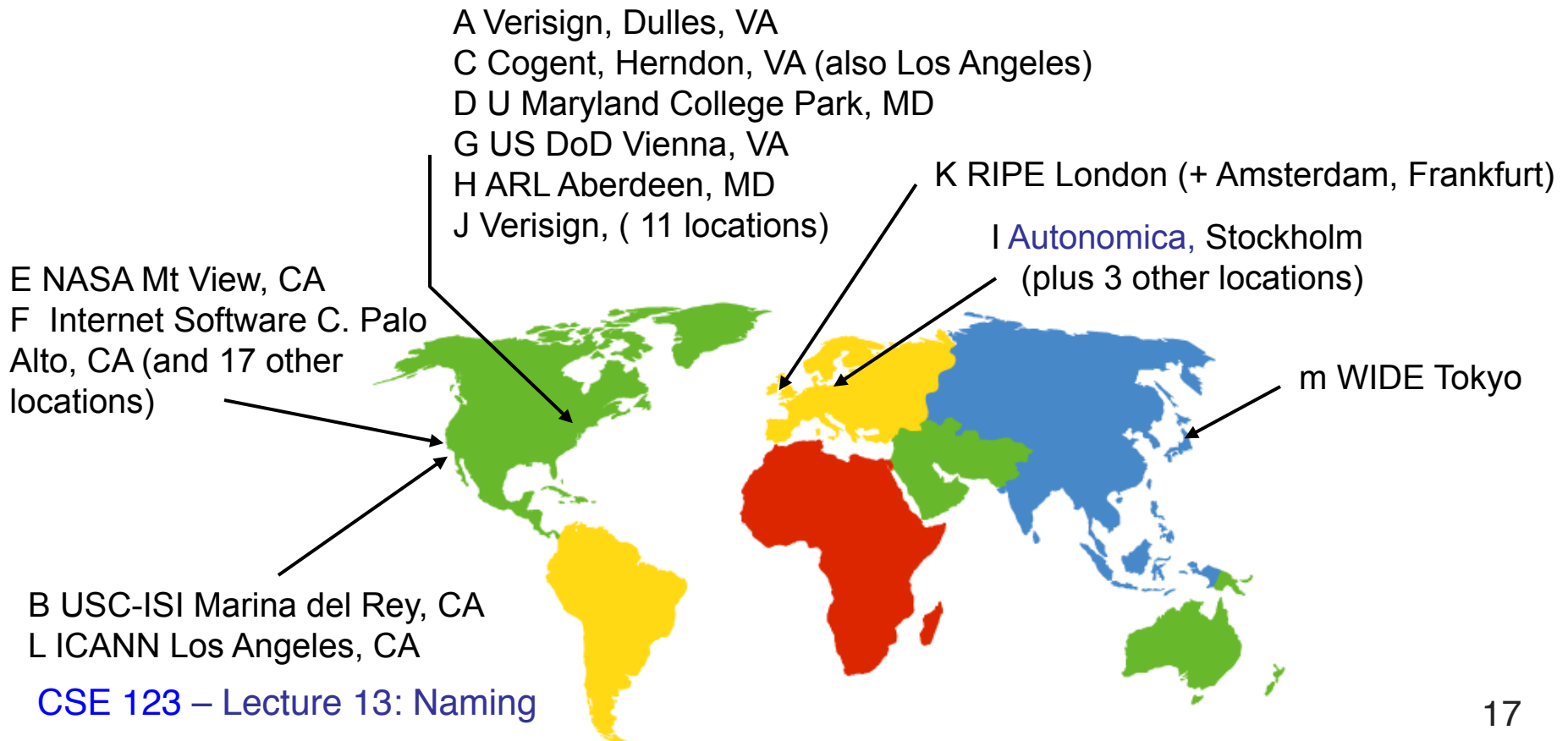
DNS: Distributed Database



DNS Root Servers



- 13 root servers (see <http://www.root-servers.org/>)
 - ♦ Labeled A through M





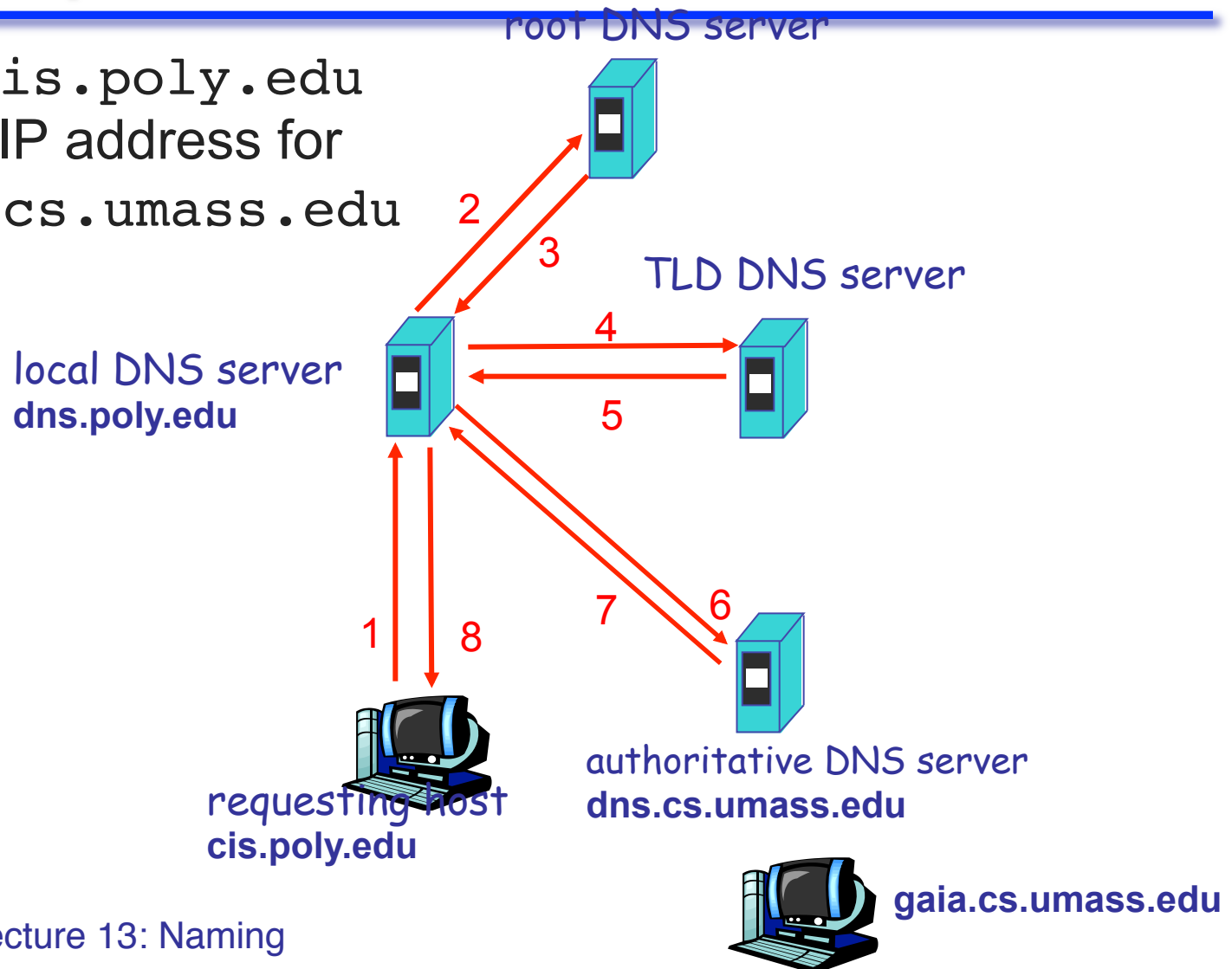
Using DNS

- Local DNS server (“default name server”)
 - ◆ Usually near the end hosts who use it
 - ◆ Local hosts configured with local server (e.g., `/etc/resolv.conf`) or learn the server via DHCP
- Client application
 - ◆ Extract server name (e.g., from the URL)
 - ◆ Do *gethostbyname()* to trigger resolver code
- Server application
 - ◆ Extract client IP address from socket
 - ◆ Optional *gethostbyaddr()* to translate into name



Example

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





Reliability

- DNS servers are replicated
 - ◆ Name service available if at least one replica is up
 - ◆ Queries can be load balanced between replicas
- UDP used for queries
 - ◆ Need reliability: must implement this on top of UDP
 - ◆ Try alternate servers on timeout
 - ◆ Exponential backoff when retrying same server
- Cache responses to decrease load
 - ◆ Both at end hosts and local servers



Summary

- IP to MAC Address mapping
 - ◆ Dynamic Host Configuration Protocol (DHCP)
 - ◆ Address Resolution Protocol (ARP)
- Domain Name System
 - ◆ Distributed, hierarchical database
 - ◆ Distributed collection of servers
 - ◆ Caching to improve performance



For Next Time

- Read 3-3.1, 3.3.3
- Start studying for the Midterm