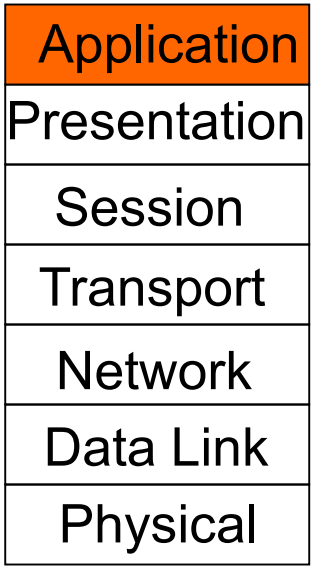
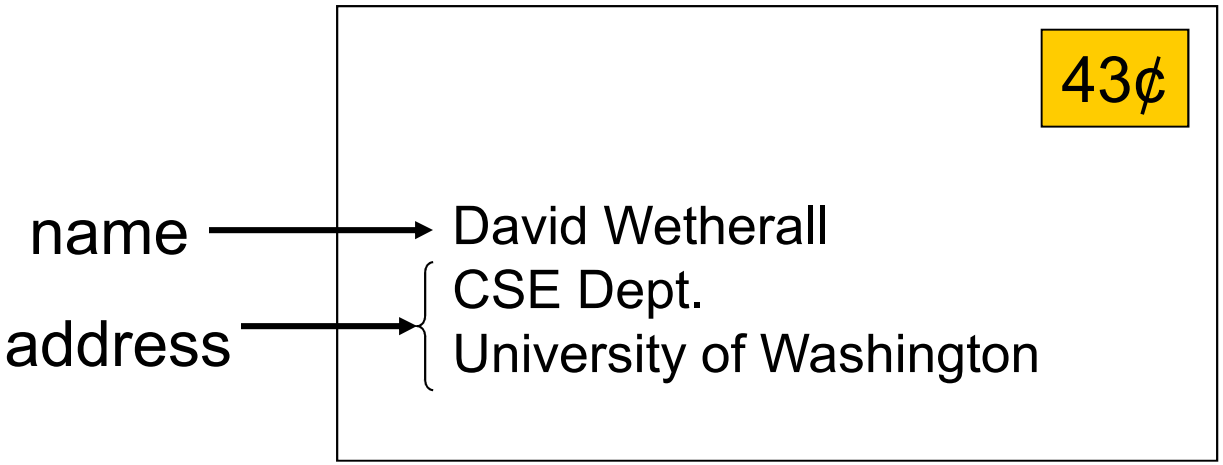
Naming and the DNS

- Focus
  - How do we name hosts etc.?
- Topics
  - Domain Name System (DNS)
  - Email/URLs



Names and Addresses



- Names are identifiers for objects/services (high level)
- Addresses are locators for objects/services (low level)
- Resolution is the process of mapping name to address
- But, addresses are really lower-level names; many levels used

Naming in Systems

- Ubiquitous
  - Files in filesystem, processes in OS, pages on the web, …
- Decouple identifier for object/service from location
  - Hostnames provide a level of indirection for IP addresses
- Naming greatly impacts system capabilities and performance
  - Ethernet addresses are a flat 48 bits
    - flat → any address anywhere but large forwarding tables
  - IP addresses are hierarchical 32/128 bits
    - hierarchy → smaller routing tables but constrained locations

Internet Hostnames

- Hostnames are human-readable identifiers for end-systems based on an administrative hierarchy
  - galah.cs.washington.edu is my desktop machine
- IP addresses are a fixed-length binary encoding for end-systems based on their position in the network
  - 128.95.2.106 is galah's IP address

- Original name resolution: HOSTS.TXT
- Current name resolution: Domain Name System
- Future name resolution: ?

Original Hostname System

- When the Internet was really young …
- Flat namespace
  - Simple (host, address) pairs
- Centralized management
  - Updates via a single master file called HOSTS.TXT
  - Manually coordinated by the Network Information Center (NIC)
- Resolution process
  - Look up hostname in the HOSTS.TXT file

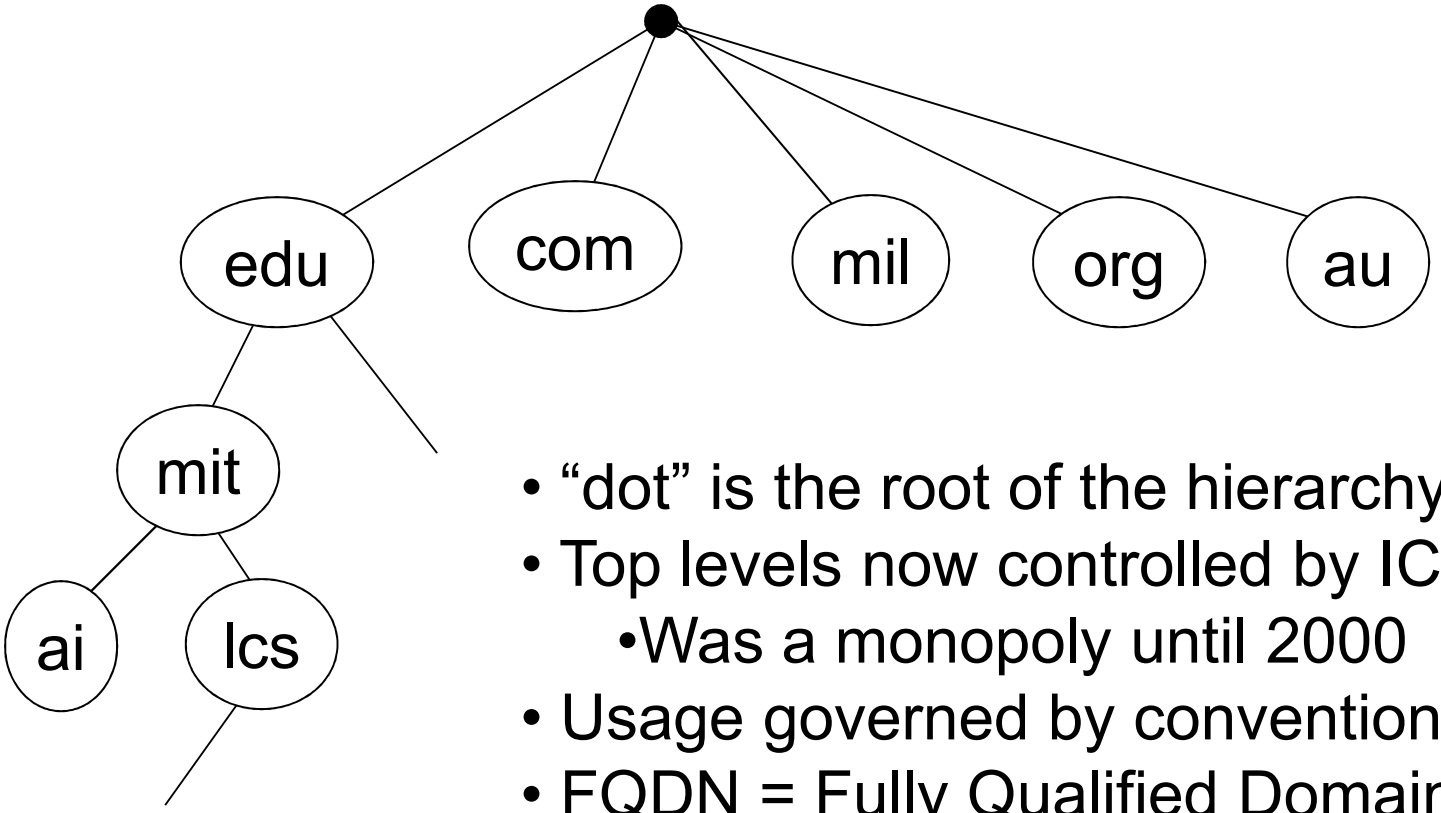
Scaling Problems

- Coordination
  - Between all users to avoid conflicts
- Inconsistencies
  - Between update and distribution of new version
- Reliability
  - Single point of failure
- Performance
  - Competition for centralized resources

Domain Name System (DNS)

- Designed by Mockapetris and Dunlap in the mid 80s
- Namespace is hierarchical
  - Allows much better scaling of data structures
  - e.g., galah.cs.washington.edu
- Namespace is distributed
  - Decentralized administration and access
  - e.g., galah managed by CSE
- Resolution is by query/response
  - With replicated servers for redundancy
  - With heavy use of caching for performance

DNS Hierarchy



- "dot" is the root of the hierarchy
- Top levels now controlled by ICANN
  - Was a monopoly until 2000
- Usage governed by conventions
- FQDN = Fully Qualified Domain Name

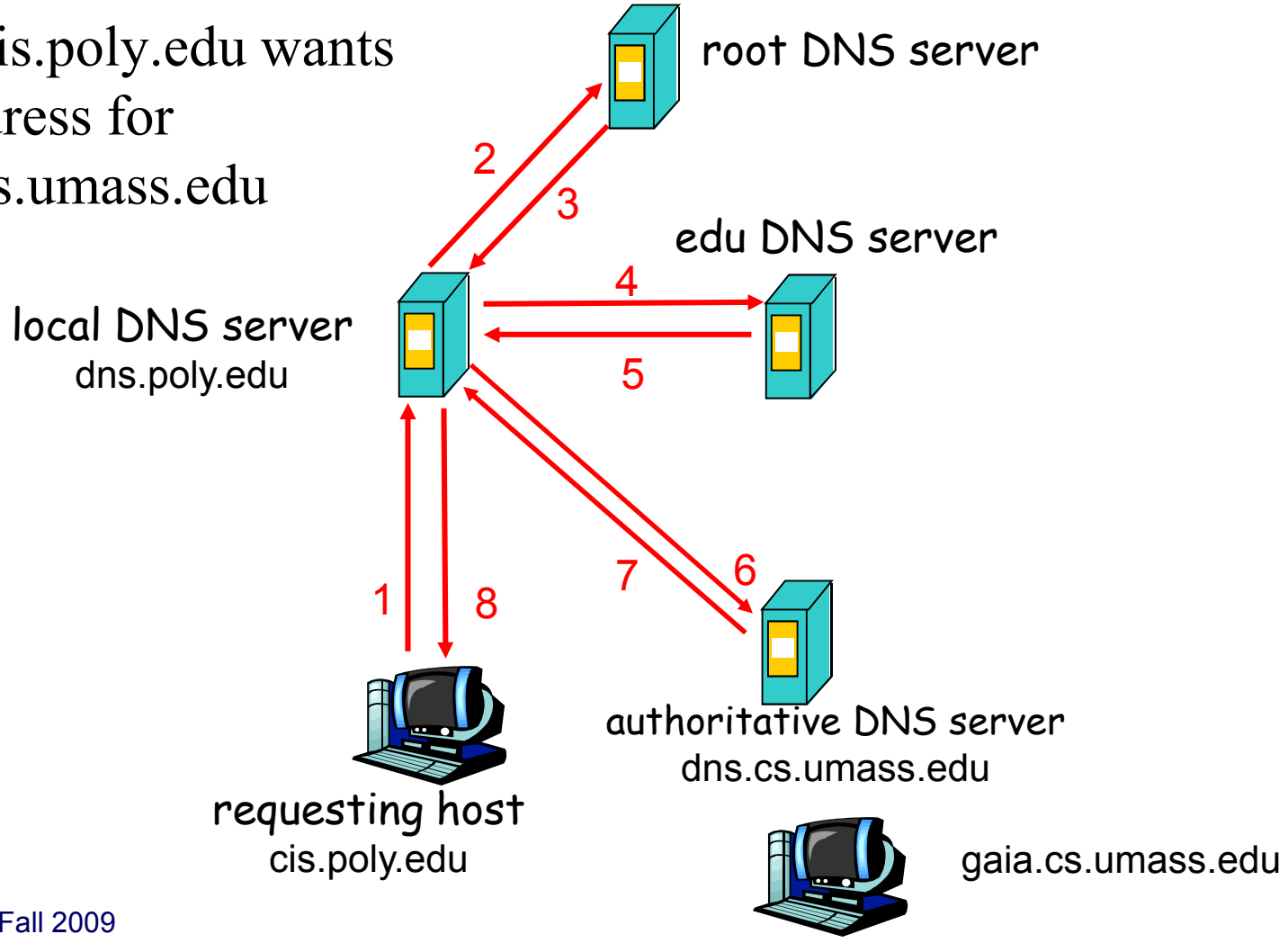
DNS Distribution

- Data managed by zones that contain resource records
  - Zone is a complete description of a portion of the namespace
  - e.g., all hosts and addresses for machines in washington.edu with pointers to subdomains like cs.washington.edu
  -
- One or more nameservers manage each zone
  - Zone transfers performed between nameservers for consistency
  - Multiple nameservers provide redundancy

- Client resolvers query nameservers for specified records
  - Multiple messages may be exchanged per DNS lookup to navigate the name hierarchy (coming soon)

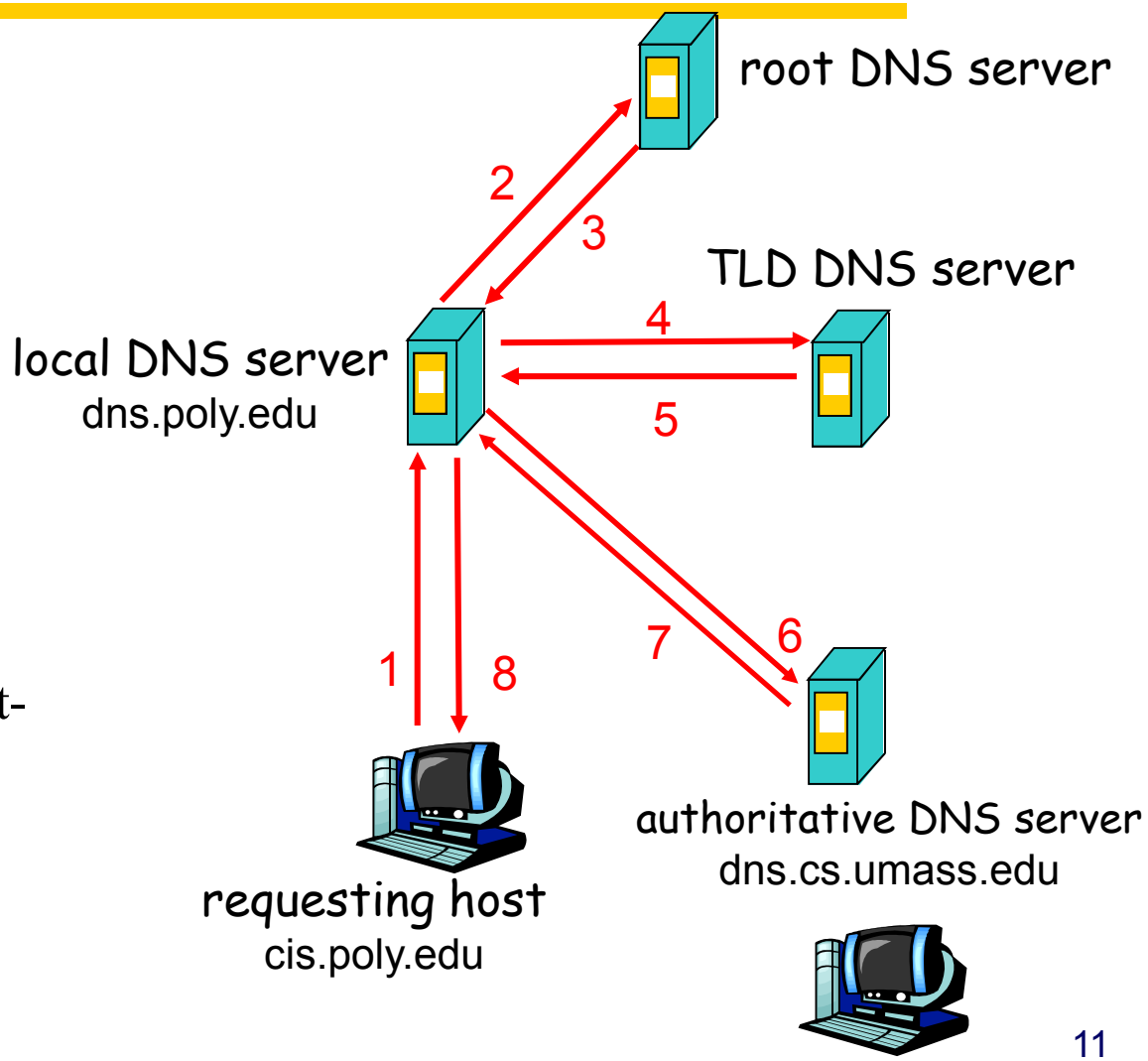
DNS Lookups / Resolution

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



Recursive vs. Iterative Queries

- Recursive query
  - Ask server to get answer for you
  - E.g., request 1 and response 8
- Iterative query
  - Ask server who to ask next
  - E.g., all other request-response pairs



DNS Resource Records

DNS: distributed db storing resource records (RR)

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

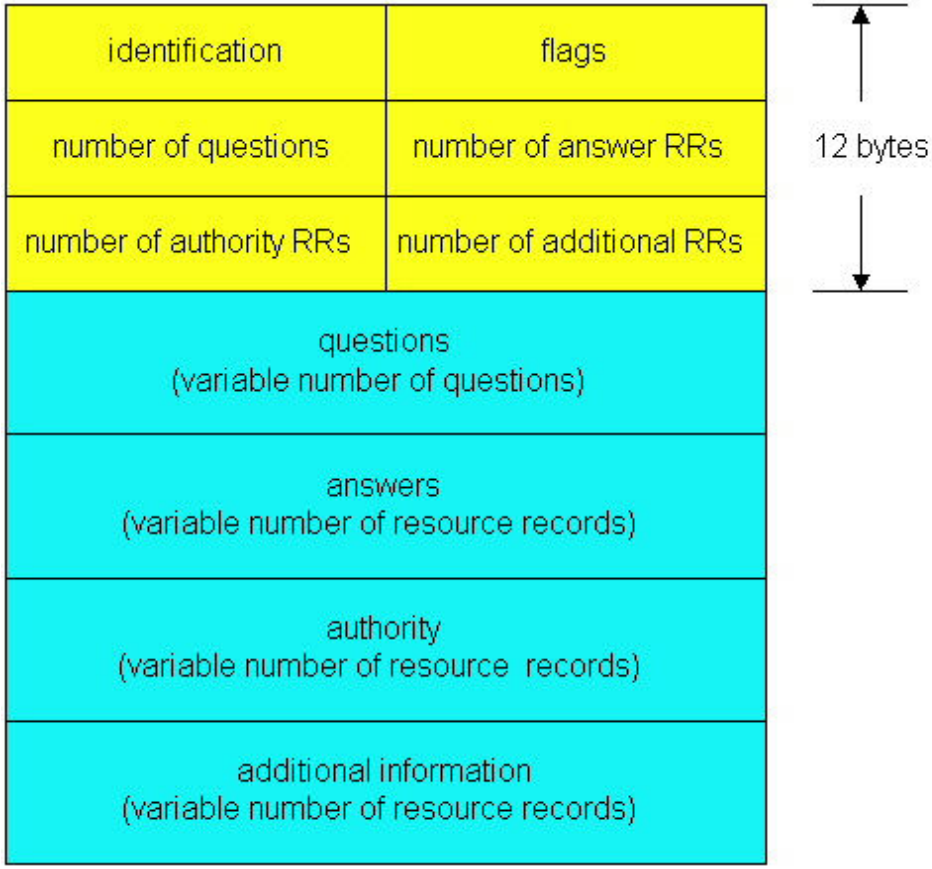
- Type=A
  - name is hostname
  - value is IP address
- Type=NS
  - **name** is domain (e.g. foo.com)
  - **value** is hostname of authoritative name server for this domain
- Type=CNAME
  - name is alias name for some "canonical" (the real) name
    www.ibm.com is really
    servereast.backup2.ibm.com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

DNS Protocol

DNS protocol : *query* and *reply* messages, both with same *message format*

Message header

- Identification: 16 bit # for query, reply to query uses same #
- Flags:
  - Query or reply
  - Recursion desired
  - Recursion available
  - Reply is authoritative



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
- Same identifier for all queries
  - Don't care which server responds

DNS Bootstrapping

- Need to know IP addresses of root servers before we can make any queries
- Addresses for 13 root servers ([a-m].root-servers.net) handled via initial configuration (named.ca file)

DNS Caching

- Performing all these queries take time
  - And all this before the actual communication takes place
  - E.g., 1-second latency before starting Web download
- Caching can substantially reduce overhead
  - The top-level servers very rarely change
  - Popular sites (e.g., www.cnn.com) visited often
  - Local DNS server often has the information cached
- How DNS caching works
  - DNS servers cache responses to queries
  - Responses include a "time to live" (TTL) field
  - Server deletes the cached entry after TTL expires

Negative Caching

- Remember things that don't work
  - Misspellings like www.cnn.comm and www.cnnn.com
  - These can take a long time to fail the first time
  - Good to remember that they don't work
  - … so the failure takes less time the next time around

Building on the DNS

- Other naming designs leverage the DNS
- Email:
  - e.g., djw@cs.washington.edu is djw in the domain cs.washington.edu
- Uniform Resource Locators (URLs) name for Web pages
  - e.g., www.cs.washington.edu/homes/djw
  - Use domain name to identify a Web server
  - Use "/" separated string to name path to page (like files)

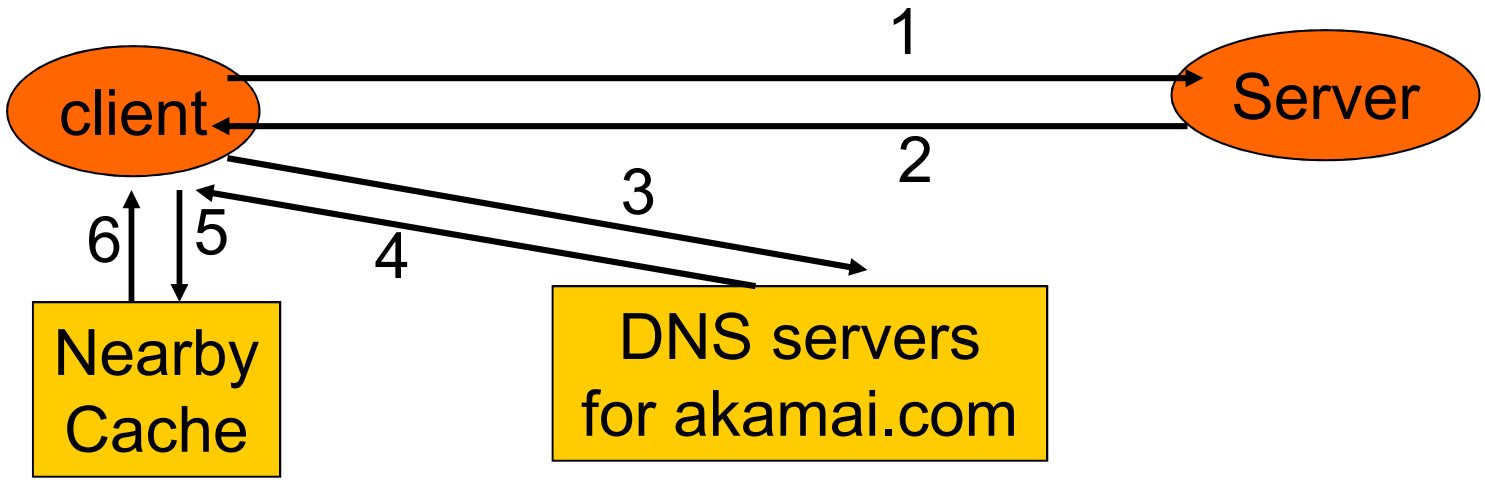
Future Evolution of the DNS

- Design constrains us in two major ways that are increasingly less appropriate

- Static host to IP mapping
  - What about mobility (Mobile IP) and dynamic address assignment (DHCP)
- Location-insensitive queries
  - What if I don't care what server a Web page comes from, as long as it's the right page?
  - e.g., a yahoo page might be replicated

Akamai

- Use the DNS to effect selection of a nearby Web cache



- Leverage separation of static/dynamic content
- Beware DNS caching