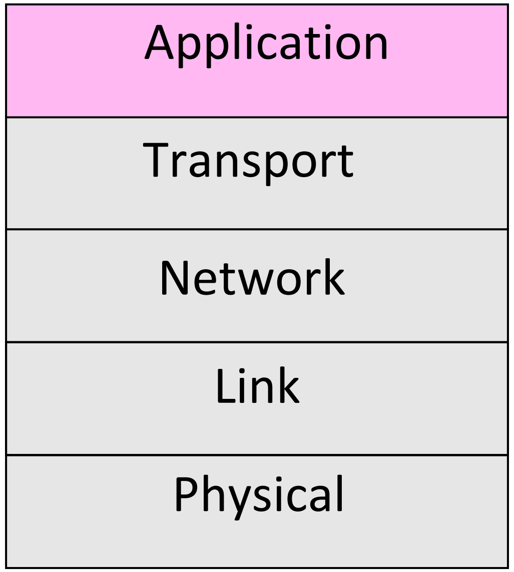
CSE 461: Computer networks

Spring 2021

Ratul Mahajan

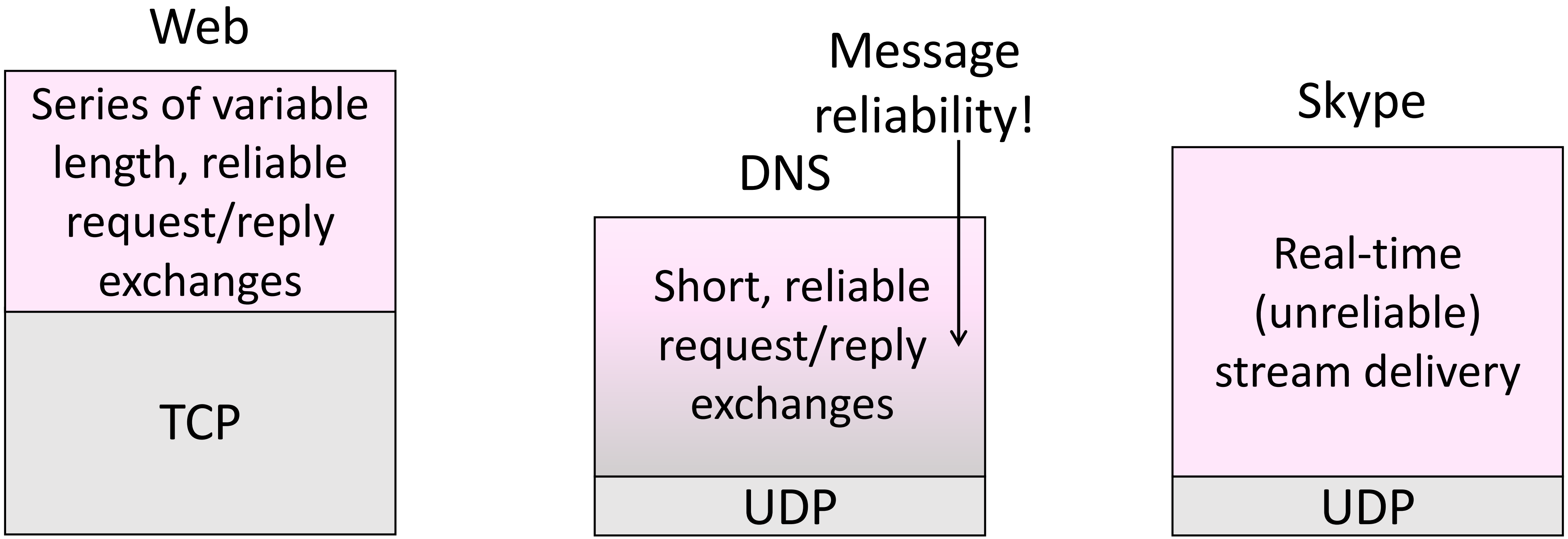
Applications

Remember this?



Application Communication Needs

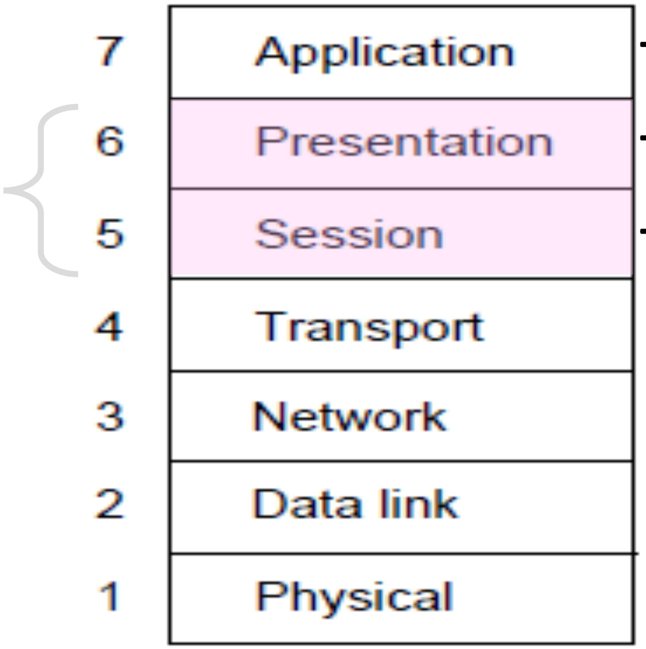
- Vary widely; build on Transport services; some use multiple transport protocols (e.g., Zoom)



Remember this?

- OSI layers that we ignore

Considered part of the application, not strictly layered!



– User’s tasks

– Converts different representations

– Manages task dialogs

Session Concept

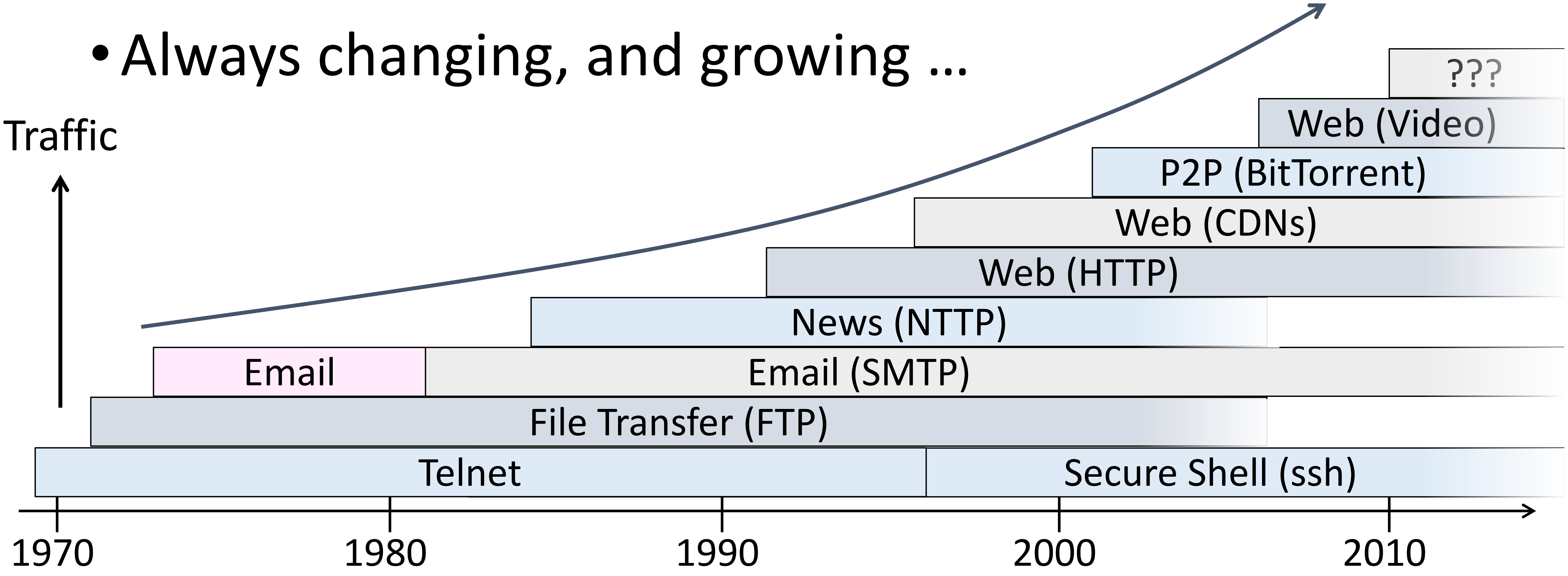
- A session is a series of related network interactions in support of an application task
  - Often informal, not explicit
- Examples:
  - Web page fetches multiple resources
  - Skype call involves audio, video, chat

Presentation Concept

- Apps need to identify the type of content, and encode it for transfer
  - These are Presentation functions
- Examples:
  - Media (MIME) types, e.g., image/jpeg, identify content type
  - Transfer encodings, e.g., gzip, identify the encoding of content
  - Application headers are often simple and readable versus packed for efficiency

Evolution of Internet Applications

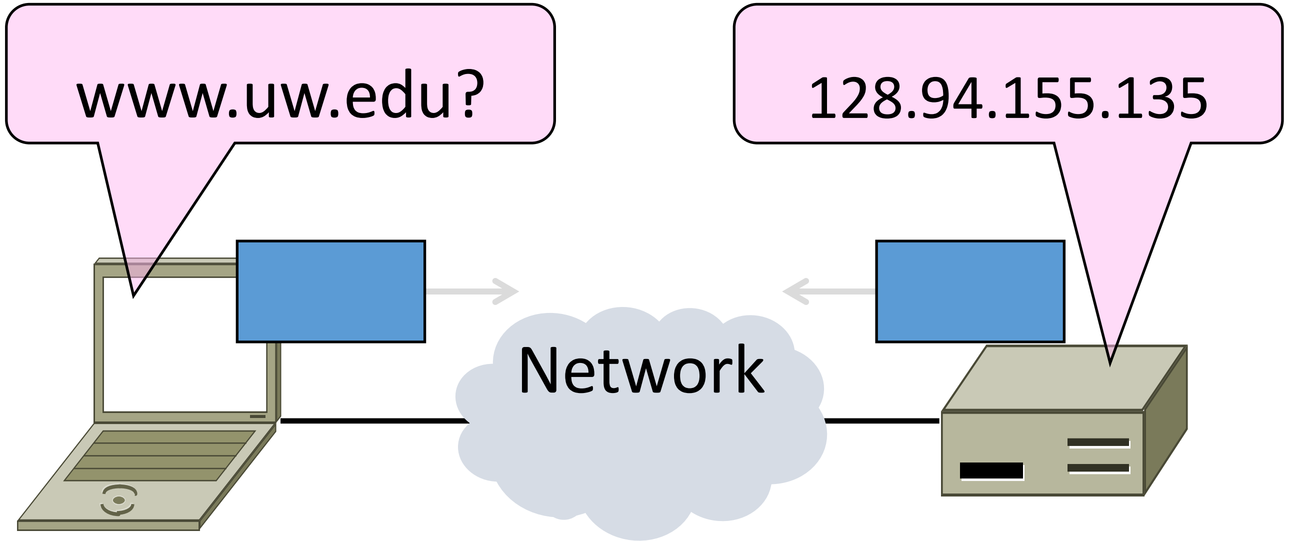
- Always changing, and growing …



Domain Name System

DNS

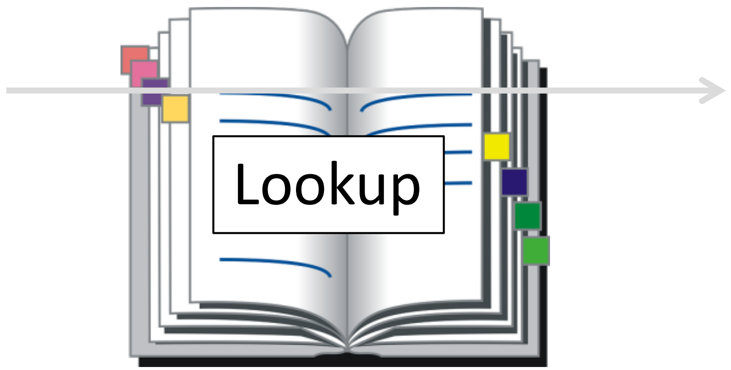
- Human-readable host names, and more



Names and Addresses

- <u>Names</u> are higher-level identifiers for resources
- <u>Addresses</u> are lower-level locators for resources
    - Multiple levels, e.g. full name → email → IP address → Ethernet addr
- <u>Resolution</u> (or lookup) is mapping a name to an address

Name, e.g.
"Joe Biden,"
or "whitehouse.gov"



Directory

Address, e.g.
"1600 Pennsylvania Ave, DC"
or IPv4 "184.24.56.92"

Before the DNS – HOSTS.TXT

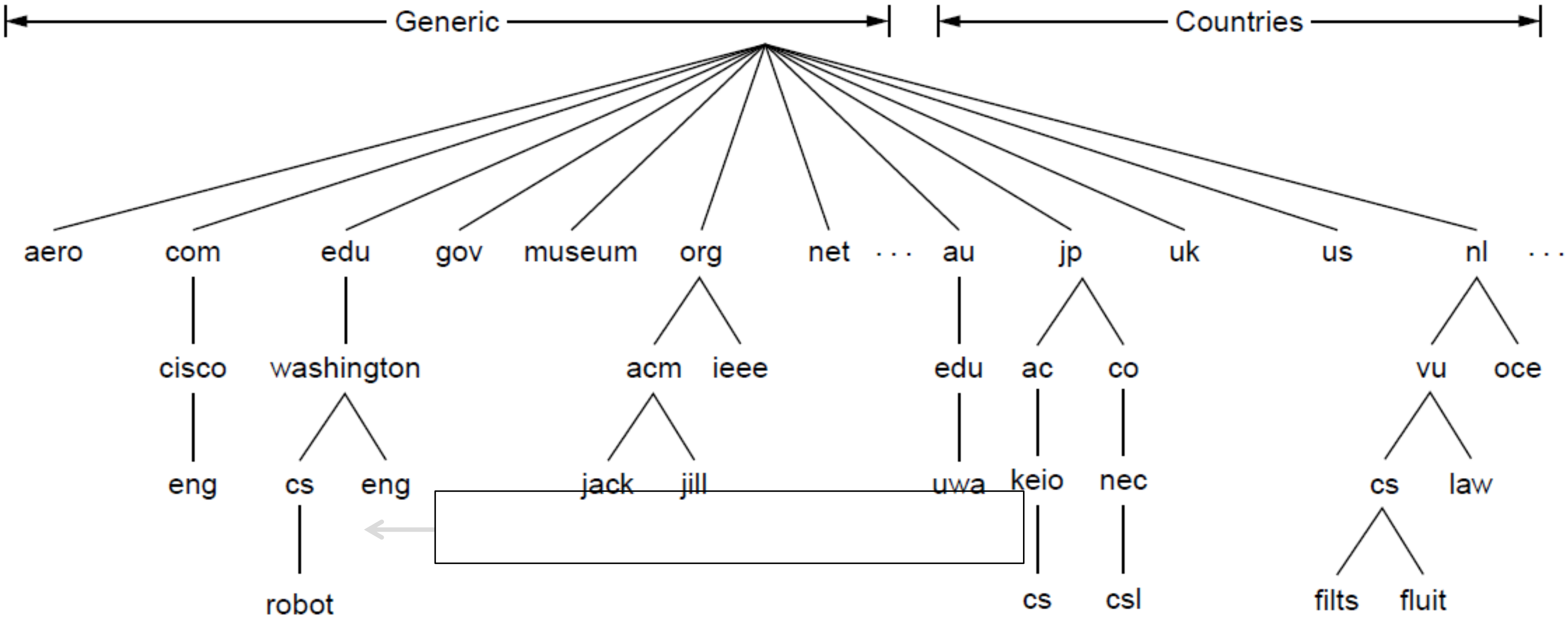
- Directory was a file HOSTS.TXT regularly retrieved for all hosts from a central machine at the NIC (Network Information Center)
- Names were initially flat, became hierarchical (e.g., lcs.mit.edu) ~85
- Not manageable or efficient as the ARPANET grew …

DNS

- A naming service to map between host names and their IP addresses (and more)
  - www.uwa.edu.au → 130.95.128.140
- Goals:
  - Easy to manage (esp. with multiple parties)
  - Efficient (good performance, few resources)
- Approach:
  - Distributed directory based on a hierarchical namespace
  - Automated protocol to tie pieces together

DNS Namespace

- Hierarchical, starting from "." (dot, typically omitted)

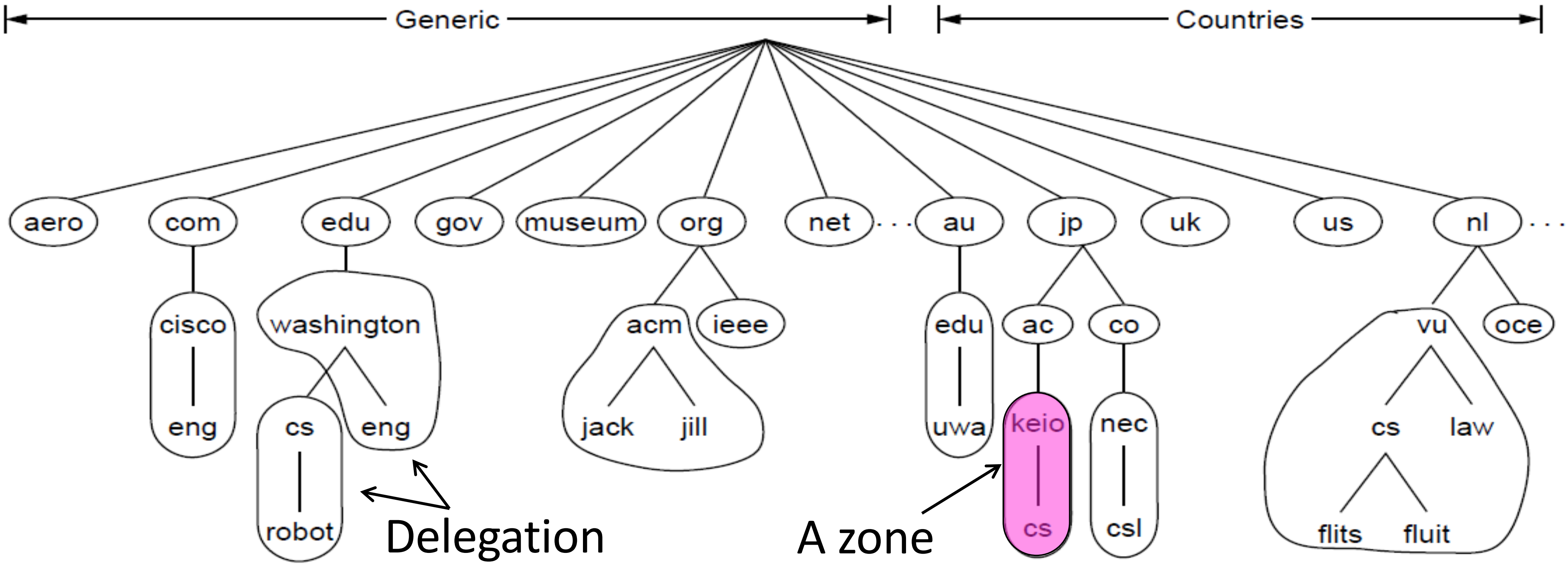


TLDs (Top-Level Domains)

- Run by ICANN (Internet Corp. for Assigned Names and Numbers)
  - Starting in ‘98; naming is financial, political, and international ☺
- 700+ generic TLDs
  - Initially .com, .edu , .gov., .mil, .org, .net
  - Unrestricted (.com) vs Restricted (.edu)
  - Added regions (.asia, .kiwi), Brands (.apple), Sponsored (.aero) in 2012
- ~250 country code TLDs
  - Two letters, e.g., “.au”, plus international characters since 2010
  - Widely commercialized, e.g., .tv (Tuvalu)
  - Many domain hacks, e.g., instagr.am (Armenia)

DNS Zones

- A <u>zone</u> is a contiguous portion of the namespace



DNS Zones (2)

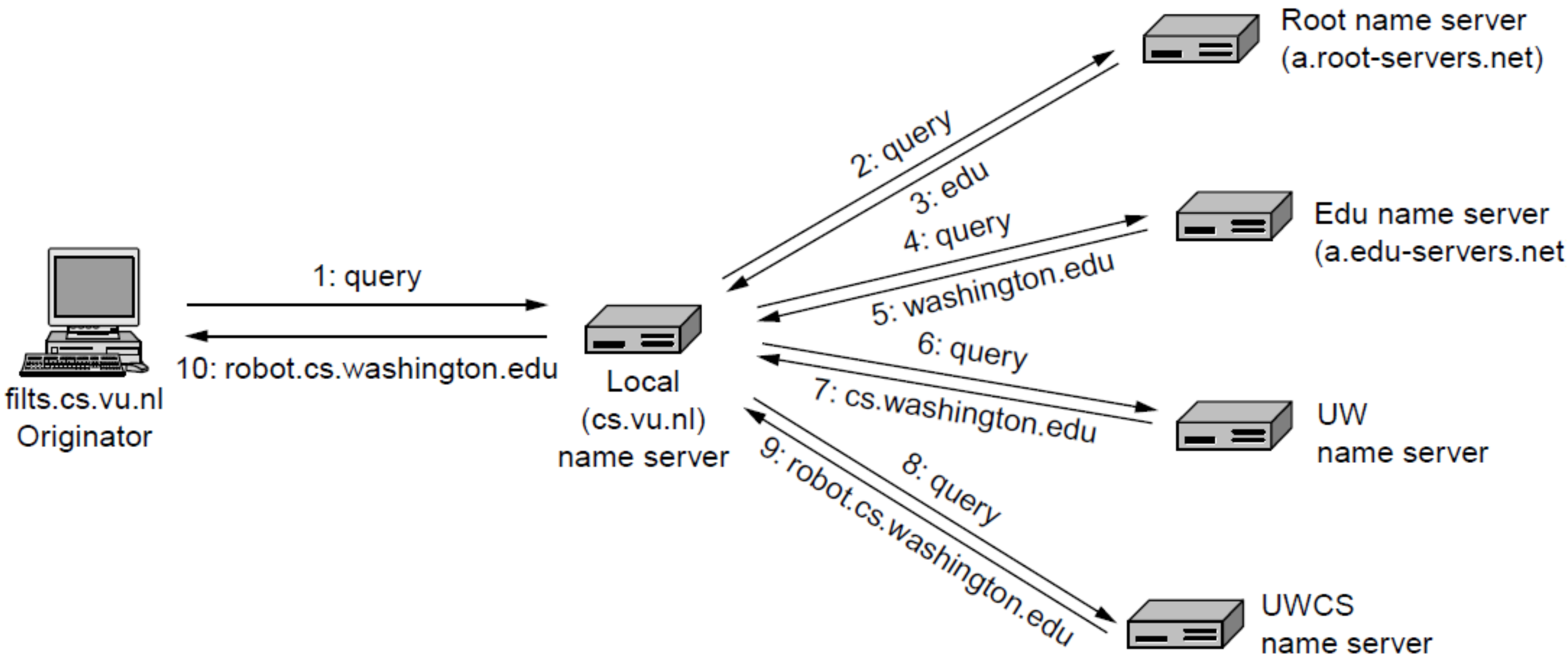
- Zones are the basis for distribution
  - EDU Registrar administers .edu
  - UW administers washington.edu
  - CSE administers cs.washington.edu
- Each zone has a <u>nameserver</u> to contact for information about it
  - Zone must include contacts for delegations, e.g., .edu knows nameserver for washington.edu

DNS Resolution

- DNS protocol lets a host resolve any host name (domain) to IP address
- If unknown, can start with the root nameserver and work down zones
- Let’s see an example first …

DNS Resolution (2)

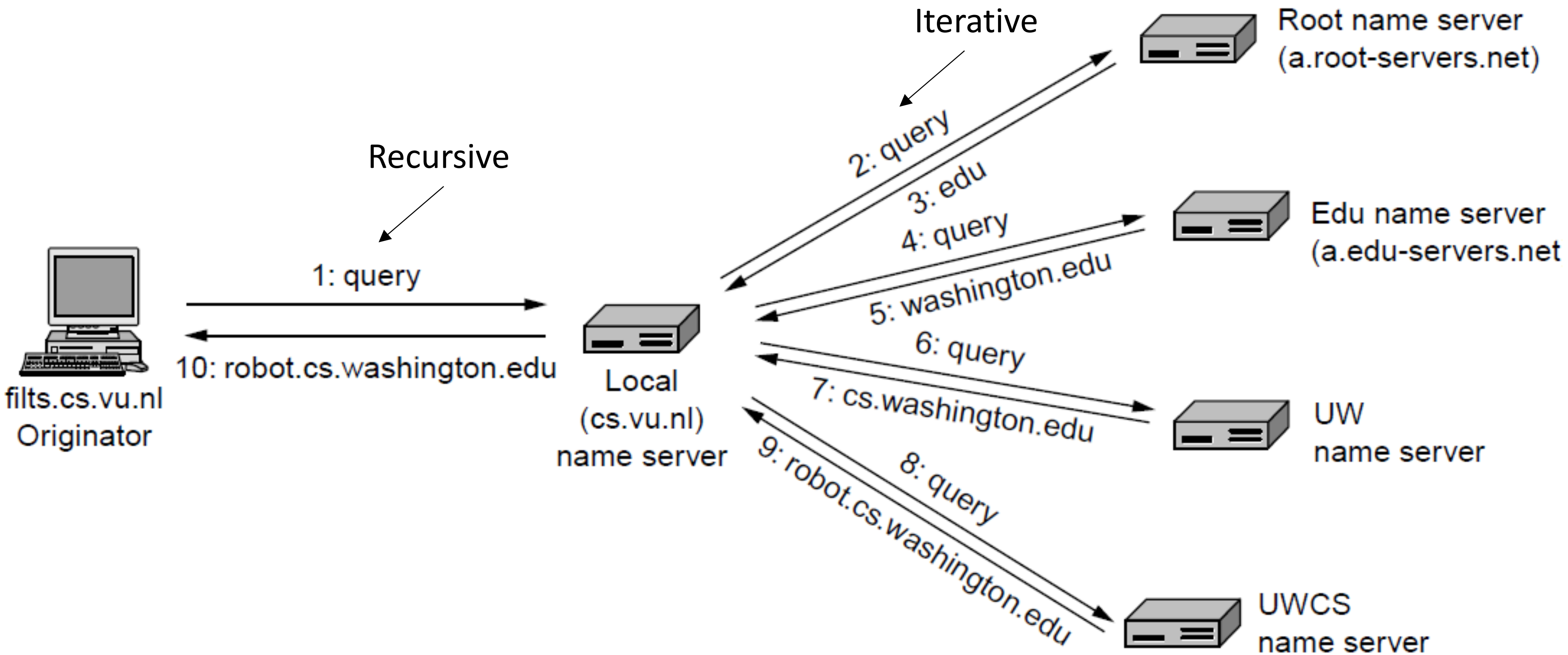
- flits.cs.vu.nl resolves robot.cs.washington.edu



Iterative vs. Recursive Queries

- Recursive query
  - Nameserver resolves and returns final answer
  - E.g., flits → local nameserver
- Iterative (Authoritative) query
  - Nameserver returns answer or who to contact for answer
  - E.g., local nameserver → all others

Iterative vs. Recursive Queries (2)



Iterative vs. Recursive Queries (3)

- Recursive query
  - Servers can offload client burden
  - Servers can cache results for a pool of clients
- Iterative query
  - Server can “file and forget”
  - Easy to build high load servers

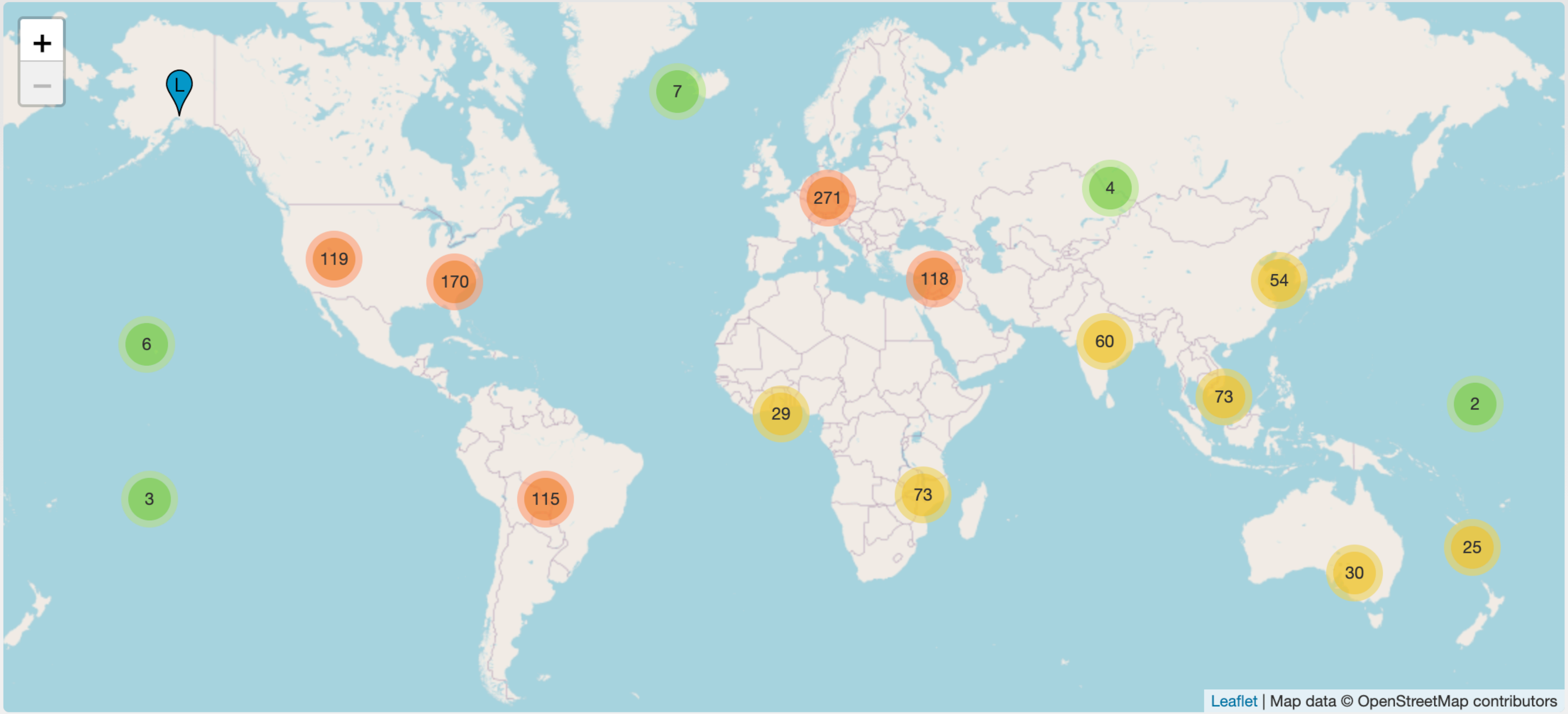
Local Nameservers

- Local nameservers often run by IT (enterprise, ISP)
  - But may be your host or AP
  - Or alternatives e.g., Google public DNS (8.8.8.8) Cloudflare’s public DNS (1.1.1.1)
- Clients need to be able to contact local nameservers
  - Typically configured via DHCP

Root Nameservers

- Root (dot) is served by 13 server names
  - a.root-servers.net to m.root-servers.net
  - All nameservers need root IP addresses
  - Handled via configuration file (named.ca)
- There are >250 distributed server instances
  - Highly reachable, reliable service
  - Most servers are reached by IP anycast (Multiple locations advertise same IP! Routes take client to the closest one.)
  - Servers are IPv4 and IPv6 reachable

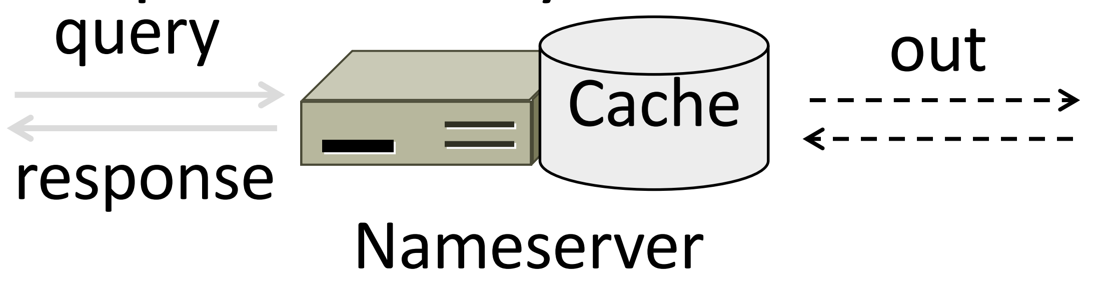
Root Server Deployment



As of 04/30/2021 5:56 a.m., the root server system consists of 1378 instances operated by the 12 independent root server operators.

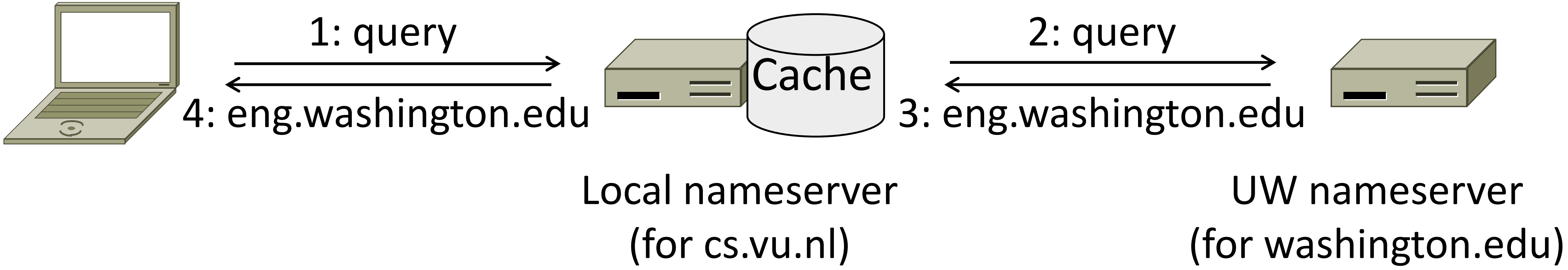
Caching

- Goal: Low resolution latency
- Observation: Names don't have much churn
- Cache query/responses to answer future queries immediately
  - Including partial (iterative) answers
  - Responses carry a TTL for caching



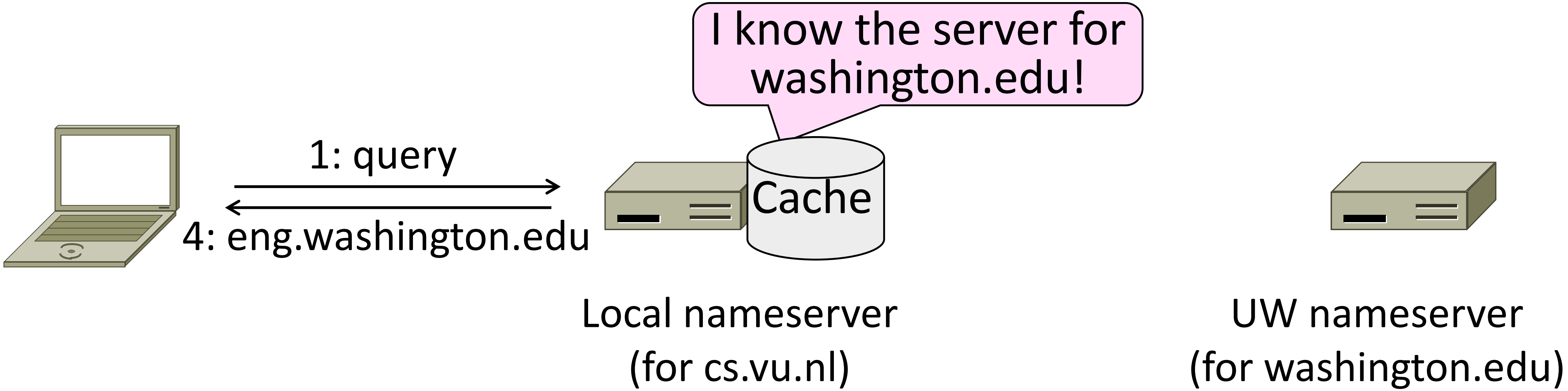
Caching (2)

- flits.cs.vu.nl looks up and stores eng.washington.edu



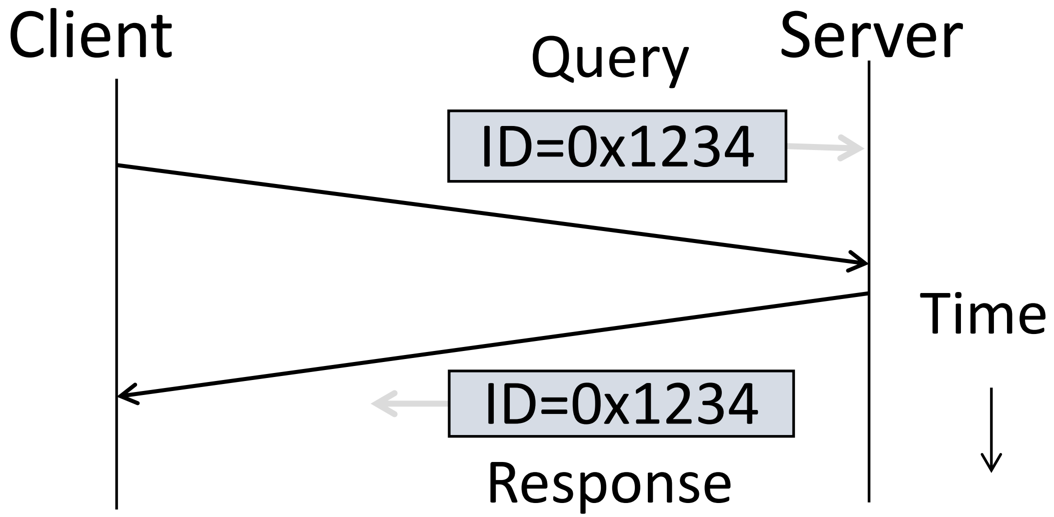
Caching (3)

- flits.cs.vu.nl now directly resolves eng.washington.edu



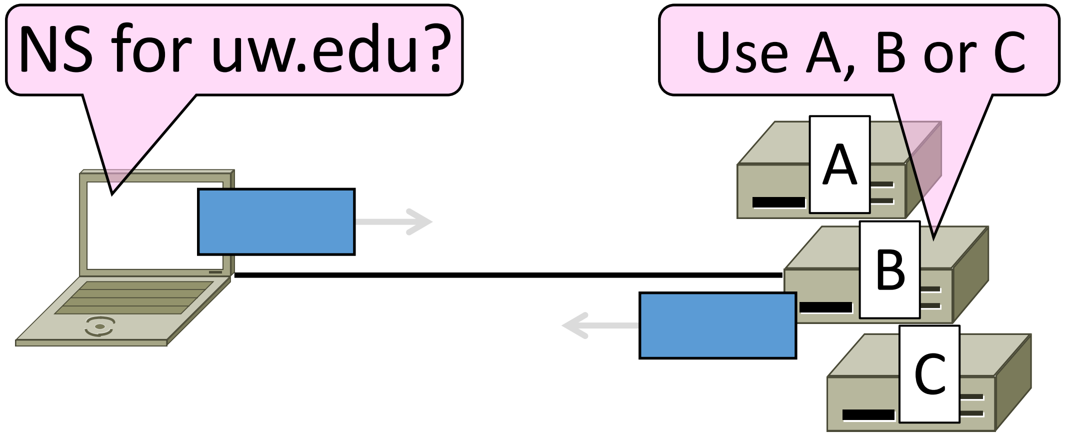
DNS Protocol

- Query and response messages
  - Built on UDP messages, port 53
  - ARQ for reliability; server is stateless!
  - Messages linked by a 16-bit ID field



DNS Protocol (2)

- Service reliability via replicas
  - Run multiple nameservers for domain
  - Return the list; clients use one answer
  - Helps distribute load too



DNS Resource Records

- A zone is comprised of DNS resource records that give information for its domain names

Type	Meaning
SOA	Start of authority, has key zone parameters
A	IPv4 address of a host
AAAA (“quad A”)	IPv6 address of a host
CNAME	Canonical name for an alias
MX	Mail exchanger for the domain
NS	Nameserver of domain or delegated subdomain

DNS Resource Records (2)

; Authoritative data for cs.vu.nl				
cs.vu.nl.	86400	IN	SOA	star boss (9527,7200,7200,241920,86400)
cs.vu.nl.	86400	IN	MX	1 zephyr
cs.vu.nl.	86400	IN	MX	2 top
cs.vu.nl.	86400	IN	NS	star
star	86400	IN	A	130.37.56.205
zephyr	86400	IN	A	130.37.20.10
top	86400	IN	A	130.37.20.11
www	86400	IN	CNAME	star.cs.vu.nl
ftp	86400	IN	CNAME	zephyr.cs.vu.nl
flits	86400	IN	A	130.37.16.112
flits	86400	IN	A	192.31.231.165
flits	86400	IN	MX	1 flits
flits	86400	IN	MX	2 zephyr
flits	86400	IN	MX	3 top
rowboat		IN	A	130.37.56.201
		IN	MX	1 rowboat
		IN	MX	2 zephyr
little-sister		IN	A	130.37.62.23
laserjet		IN	A	192.31.231.216

← Start of Authority (SOA record)

← Name server (NS record)

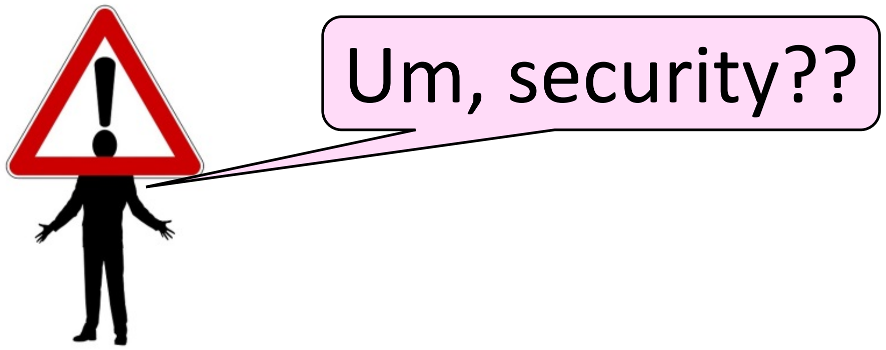
← IP addresses of computers (A records)

← Mail gateways (rowboat MX records)

DIG DEMO

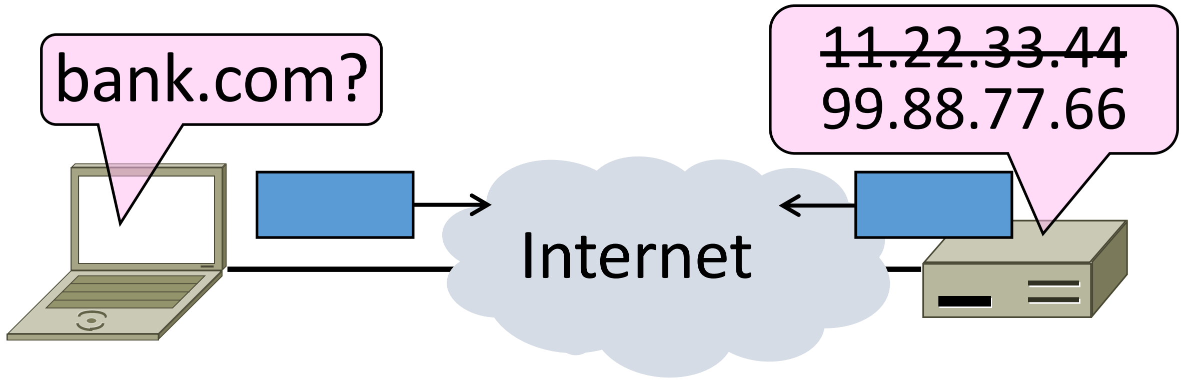
DNS Security

- Security is a major issue
  - Compromise redirects to wrong site!
  - Not part of initial protocols ..
- DNSSEC (DNS Security Extensions)
  - Mostly deployed



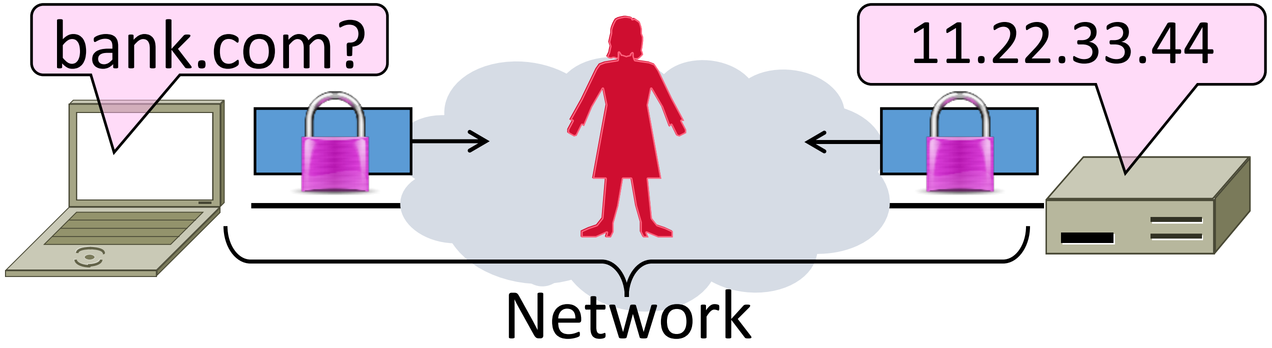
Goal and Threat Model

- Naming is a crucial Internet service
  - Binds host name to IP address
  - Wrong binding can be disastrous…



Goal and Threat Model (2)

- Goal is to secure the DNS so that the returned binding is correct
  - Integrity vs confidentiality
- Attacker can tamper with messages on the network



DNS Spoofing

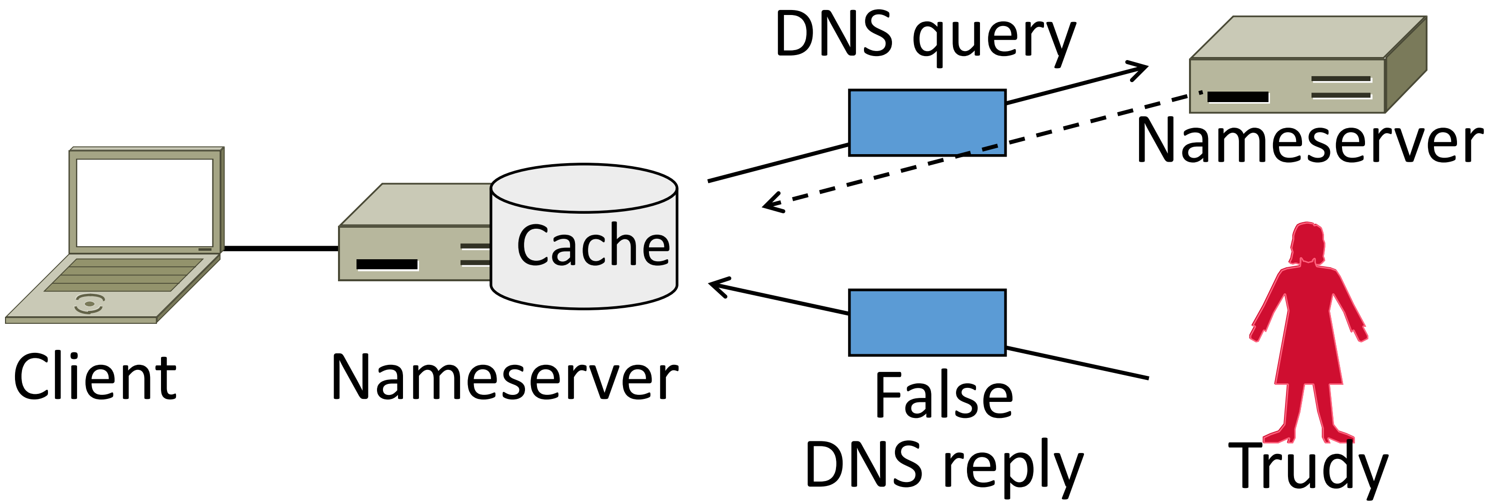
- Hang on – how can attacker corrupt the DNS?

DNS Spoofing

- Hang on – how can attacker corrupt the DNS?
- Can trick nameserver into caching the wrong binding
  - By using the DNS protocol itself
  - This is called <u>DNS spoofing</u>

DNS Spoofing (2)

- To spoof, Trudy returns a fake DNS response that appears to be true
  - Fake response contains bad binding



DNS Spoofing (3)

- Lots of questions!
  1. How does Trudy know when the DNS query is sent and what it is for?
  2. How can Trudy supply a fake DNS reply that appears to be real?
  3. What happens when the real DNS reply shows up?
- There are solutions to each issue …

DNS Spoofing (4)

1. How does Trudy know when the query is sent and what it is for?

DNS Spoofing (5)

1. How does Trudy know when the query is sent and what it is for?

- Trudy can make the query herself!
  - Nameserver works for many clients
  - Trudy is just another client

DNS Spoofing (6)

2. How can Trudy supply a fake DNS reply that appears to be real?

DNS Spoofing (7)

2. How can Trudy supply a fake DNS reply that appears to be real?

- A bit more difficult. DNS checks:
  - Reply is from authoritative nameserver (e.g., .com)
  - Reply ID that matches the request
  - Reply is for outstanding query
- (Nothing about content though …)

DNS Spoofing (8)

2. How can Trudy supply a fake DNS reply that appears to be real?

- Example Technique:
    1. Put IP of authoritative nameserver as the source IP ID is 16 bits (64K)
    2. Send reply right after query
    3. Send many guesses! (Or if a counter, sample to predict.)
- Good chance of succeeding!

DNS Spoofing (8)

3. What happens when real DNS reply shows up?

DNS Spoofing (9)

3. What happens when real DNS reply shows up?

- Likely not be a problem
  - There is no outstanding query after fake reply is accepted
  - So real reply will be discarded

DNSSEC (DNS Security Extensions)

- Extends DNS with new record types
  - RRSIG for digital signatures of records
  - DNSKEY for public keys for validation
  - DS for public keys for delegation
  - First version in ‘97, revised by ’05
- Deployment requires software upgrade at both client and server
  - Root servers upgraded in 2010
  - Followed by uptick in deployment