Lecture 4

The Application Layer

Part 2

Subjects of today:

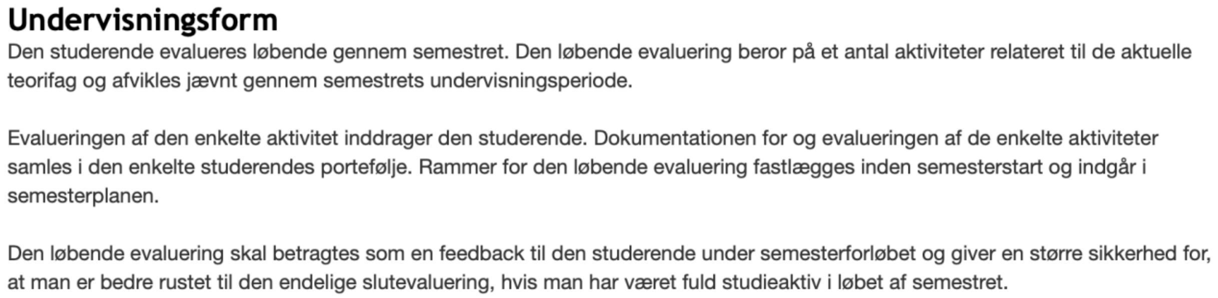
- A bit of admin
- The Domain Name System
- Wireshark Lab
- Python Lab

07 - Socket programming in Python LECTURE PLAN

20. okt. 08.00–12.00

A Bit of Admin

- The exercises
  - Course description “Idustrial Datakommunikation”:



Undervisningsform

Den studerende evalueres løbende gennem semestret. Den løbende evaluering beror på et antal aktiviteter relateret til de aktuelle teorifag og afvikles jævnt gennem semestrets undervisningsperiode.

Evalueringen af den enkelte aktivitet inddrager den studerende. Dokumentationen for og evalueringen af de enkelte aktiviteter samles i den enkelte studerendes portefølje. Rammer for den løbende evaluering fastlægges inden semesterstart og indgår i semesterplanen.

Den løbende evaluering skal betragtes som en feedback til den studerende under semesterforløbet og giver en større sikkerhed for, at man er bedre rustet til den endelige slutevaluering, hvis man har været fuld studieaktiv i løbet af semestret.

- Exam format
  - Oral examination
  - 4-8 subjects
    - Each for the for layer top + ? (TBD before last lecture)
    - Random draw
  - Prepared presentations in advance
    - Prefereable with visual aid, e.g. Power Point

4.1 The Domain Name System

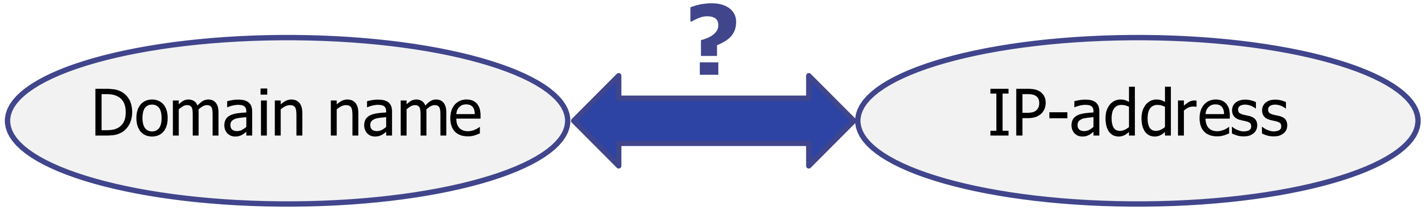
DNS: domain name system

People: many identifiers:

- SSN, name, passport#

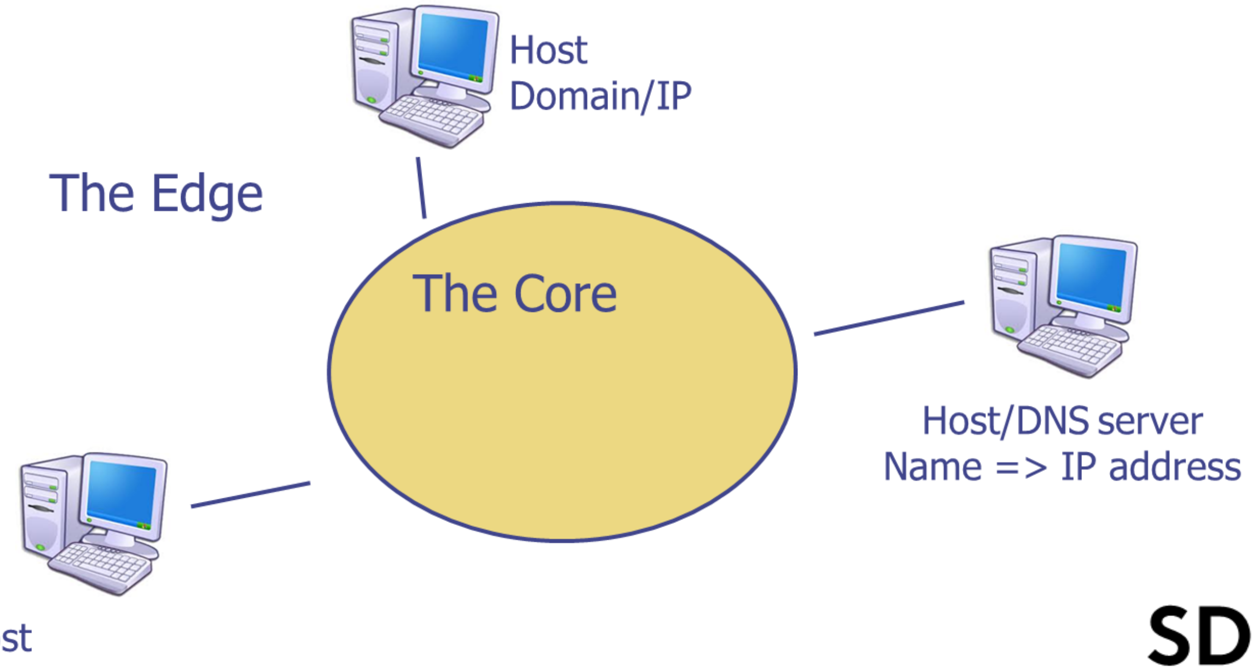
Internet hosts, routers:

- IP address (32 bit) - used for addressing datagrams
- “name”, e.g., cs.umass.edu - used by humans



DNS: domain name system

- Distributed database implemented in hierarchy of many name servers
- *Application-layer protocol:* Hosts, name servers communicate to resolve names
  - Note: Core Internet function, **implemented as application-layer protocol**
  - Complexity at network's "edge"



DNS: services, structure

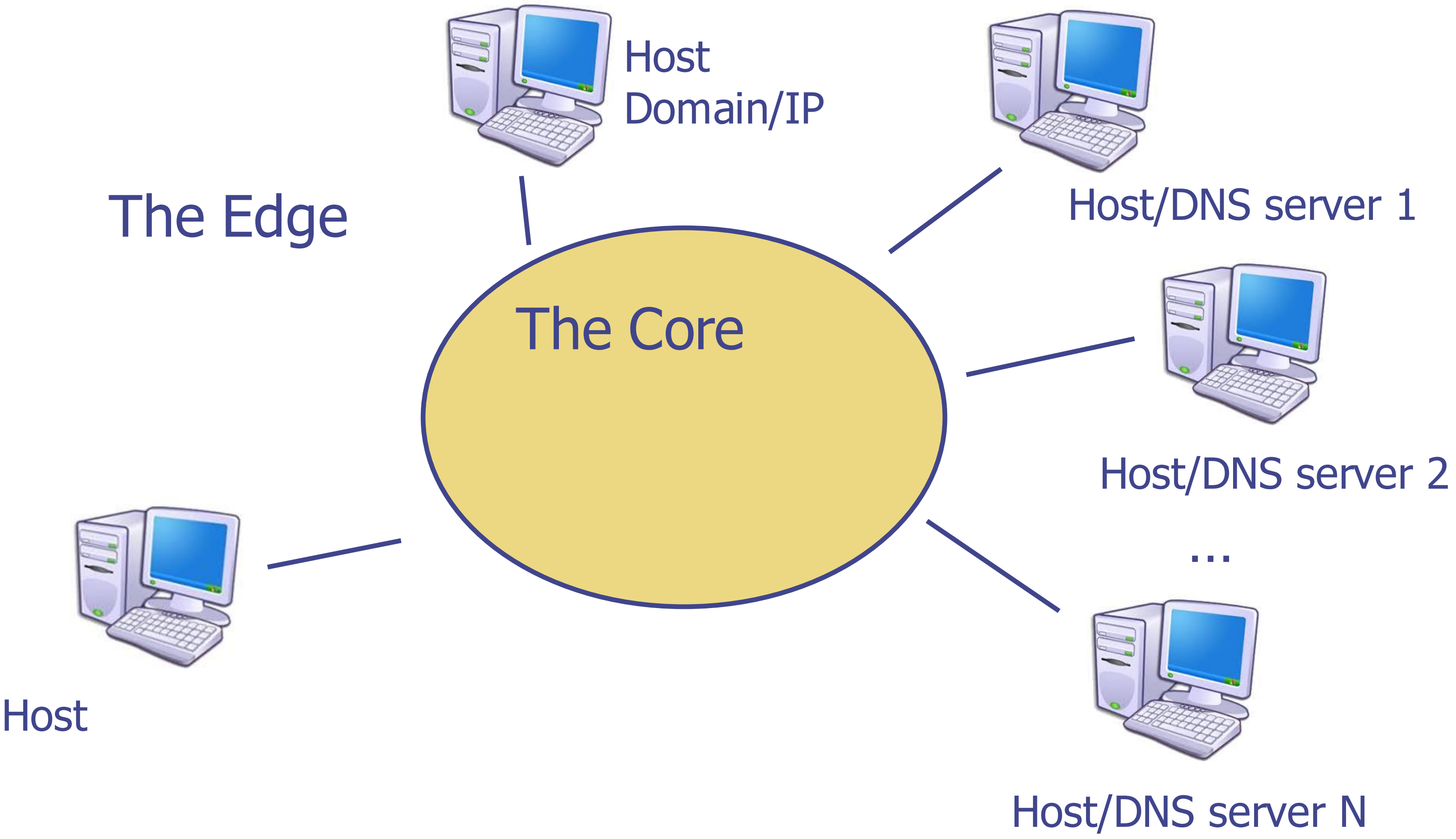
- Hostname to IP address translation
- Host aliasing — e.g. → gmail.com & googlemail.com
  - ○ Many hostnames to one IP
  - ○ Servers often have more "specific" canonical hostnames (due to infrastructure)
- Mail server aliasing
  - ○ Mail server can have same name as a webserver
- Load distribution
  - ○ Many IP addresses correspond to one hostname

Centralized DNS?

- Single point of failure
- Traffic volume
- Distant centralized database
- Maintenance

Should implementation be centralized or decentralized? Why?

DNS - An application at the edge of the internet.



The Edge

The Core

Host
Domain/IP

Host/DNS server 1

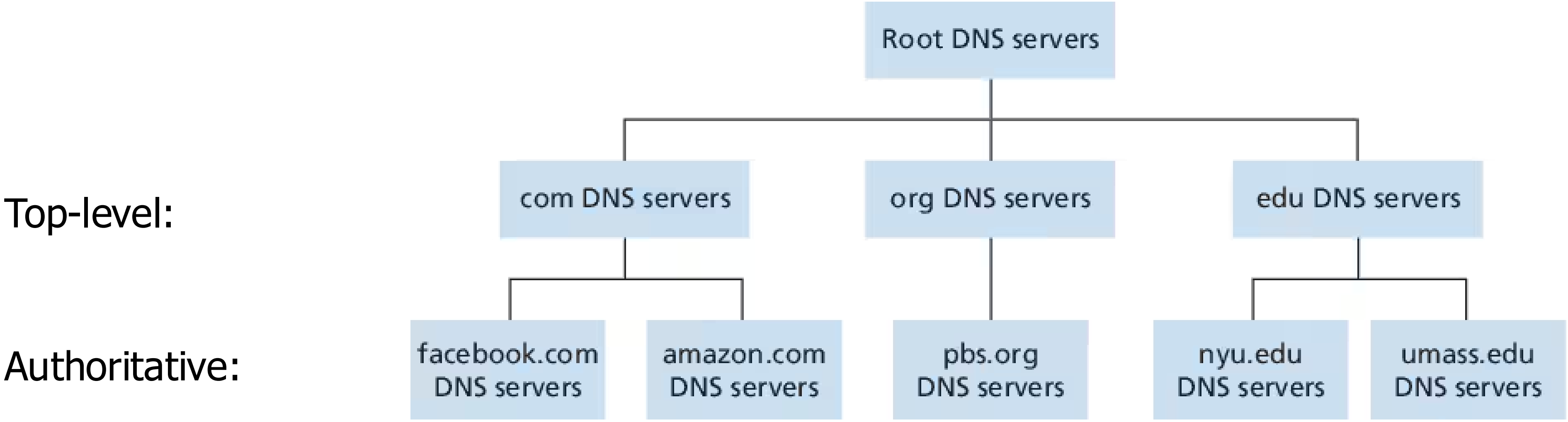
Host/DNS server 2

...

Host/DNS server N

Host

DNS: a distributed, hierarchical database

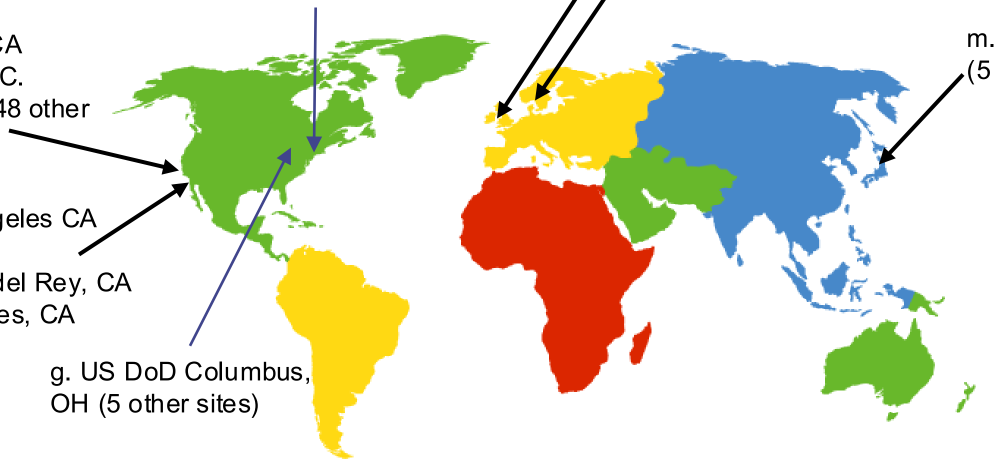


client wants IP for www.amazon.com; 1st approximation:

- client queries 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

DNS: root name servers

- Root name server:
  - Contacts authoritative name server if name mapping not known
  - Gets mapping
  - Returns mapping to local name server



c. Cogent, Herndon, VA (5 other sites)
d. U Maryland College Park, MD
h. ARL Aberdeen, MD
j. Verisign, Dulles VA (69 other sites)

k. RIPE London (17 other sites)

i. Netnod, Stockholm (37 other sites)

m. WIDE Tokyo
(5 other sites)

e. NASA Mt View, CA
f. Internet Software C.
Palo Alto, CA (and 48 other
sites)

a. Verisign, Los Angeles CA
(5 other sites)
b. USC-ISI Marina del Rey, CA
l. ICANN Los Angeles, CA
(41 other sites)

g. US DoD Columbus,
OH (5 other sites)

- *13 logical root name "servers" worldwide*
- *Each "server" replicated many times*

Top-Level Domain, Authoritative servers

TLD servers:

- Responsible for com, org, net, edu, aero, jobs, museums, and all top-level country domains, e.g.: uk, fr, ca, jp
- *Network Solutions* maintains servers for .com TLD
- *Educause* for .edu TLD
- *Punktum dk* for .dk

Authoritative DNS servers:

- Has the final authority on the name resolution!
- Organization's own DNS server(s), providing authoritative hostname/IP mappings for organization's named hosts
- Can be maintained by organization or service provider

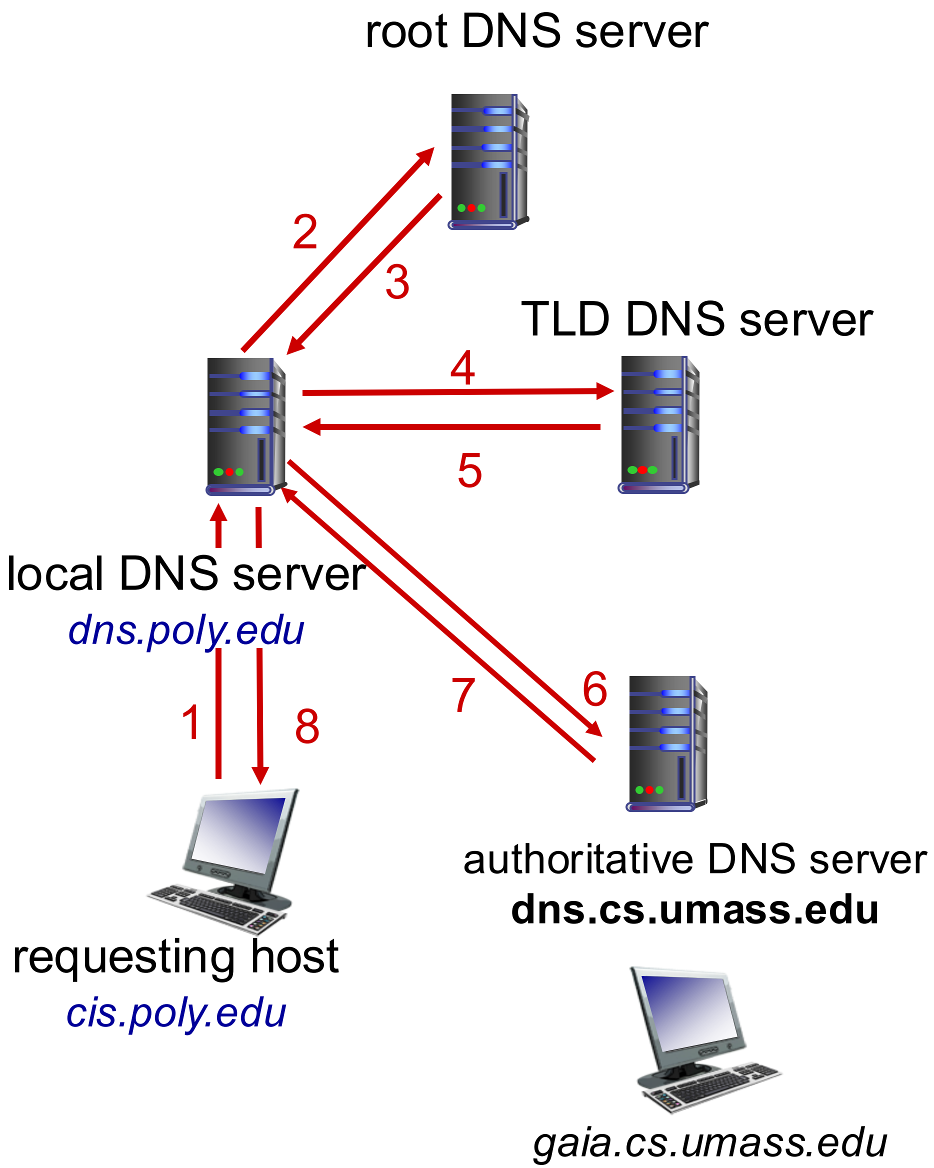
Local DNS name server

- 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 name resolution example

Iterative query:

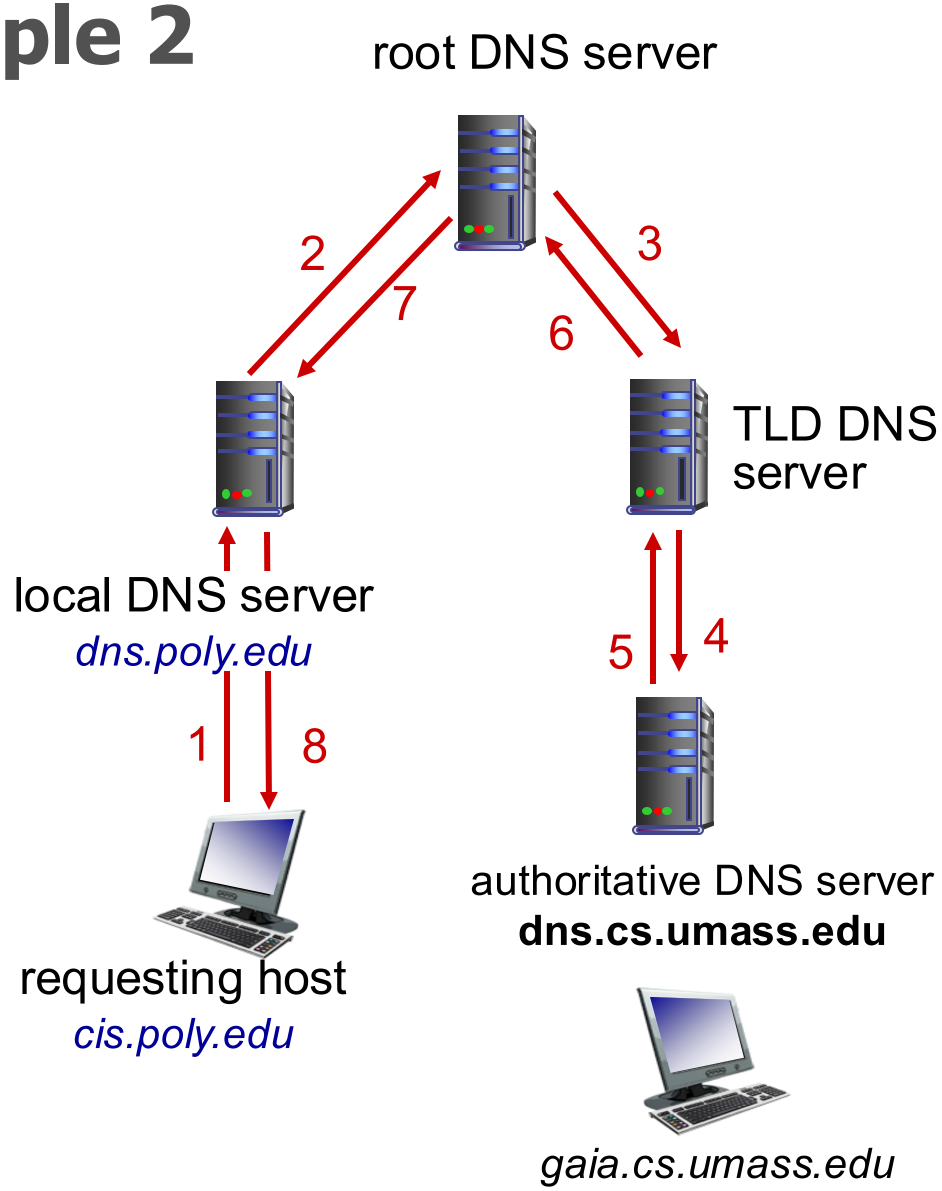
- Contacted server replies with name of server to contact
- “I don’t know this name, but ask this server”



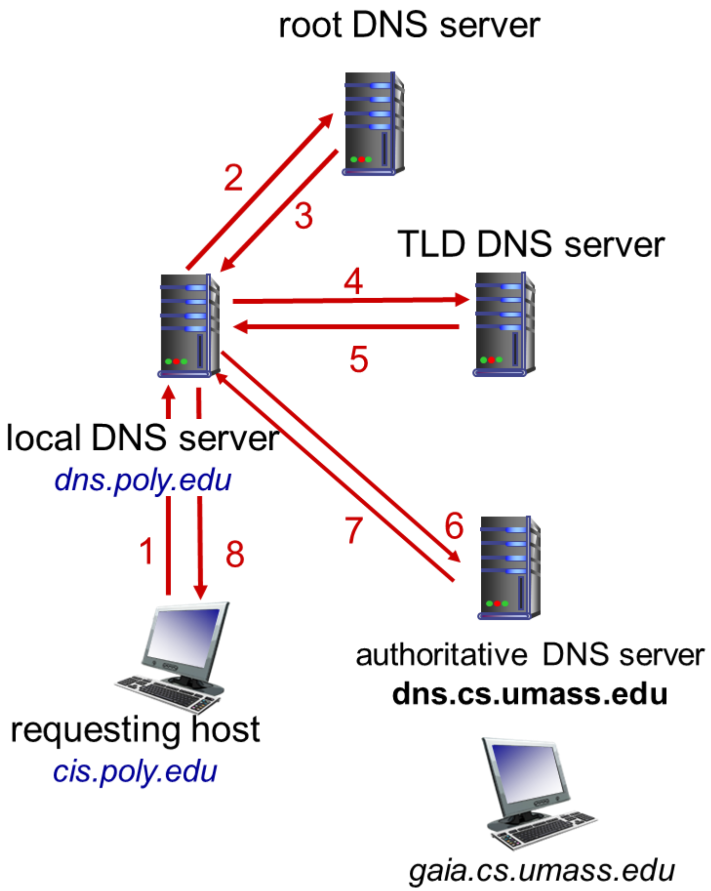
DNS name resolution example 2

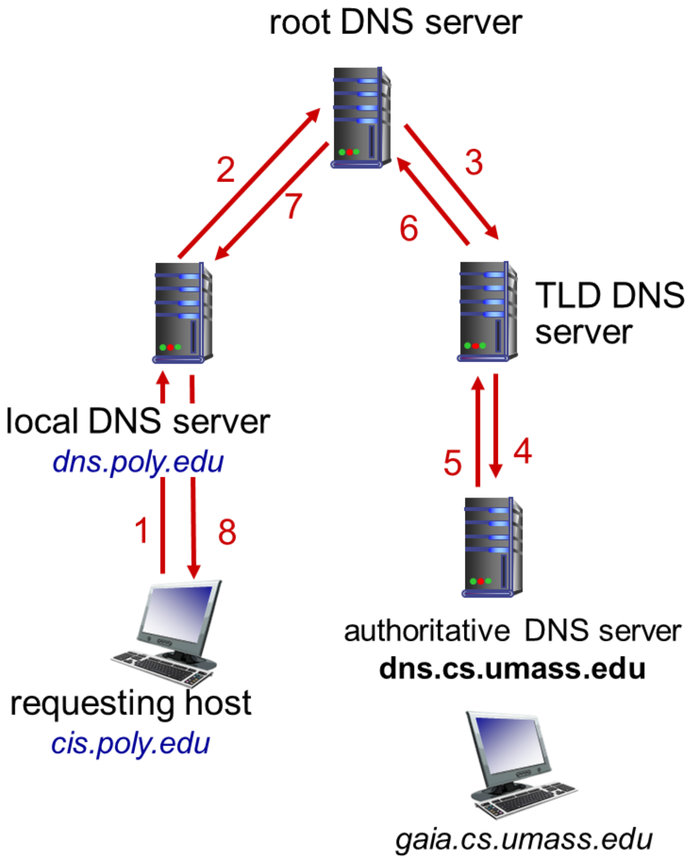
Recursive query:

- Requesting host makes a query for a full name resolution



DNS name resolving resource considerations





Discuss where are the load heaviest in the two scenarios?

DNS: caching, updating records

- Once (any) name server learns a mapping, it *caches* that mapping
  - Cache entries timeout (disappear) after some time (TTL)
  - TLD servers typically cached in local name servers
    - Thus, root name servers not often visited
- Cached entries may be *out-of-date*
  - If name host changes IP address, may not be known Internet-wide until all TTLs expire
  - **Local DNS servers typically have a cache (for speed!)**
- Update/notify mechanisms proposed IETF standard
  - RFC 2136

What could be arguments for not using the RFC 2136?

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 name
- **value** is canonical name
- Recall: **www.ibm.com** is really **servereast.backup2.ibm.com**

type=MX

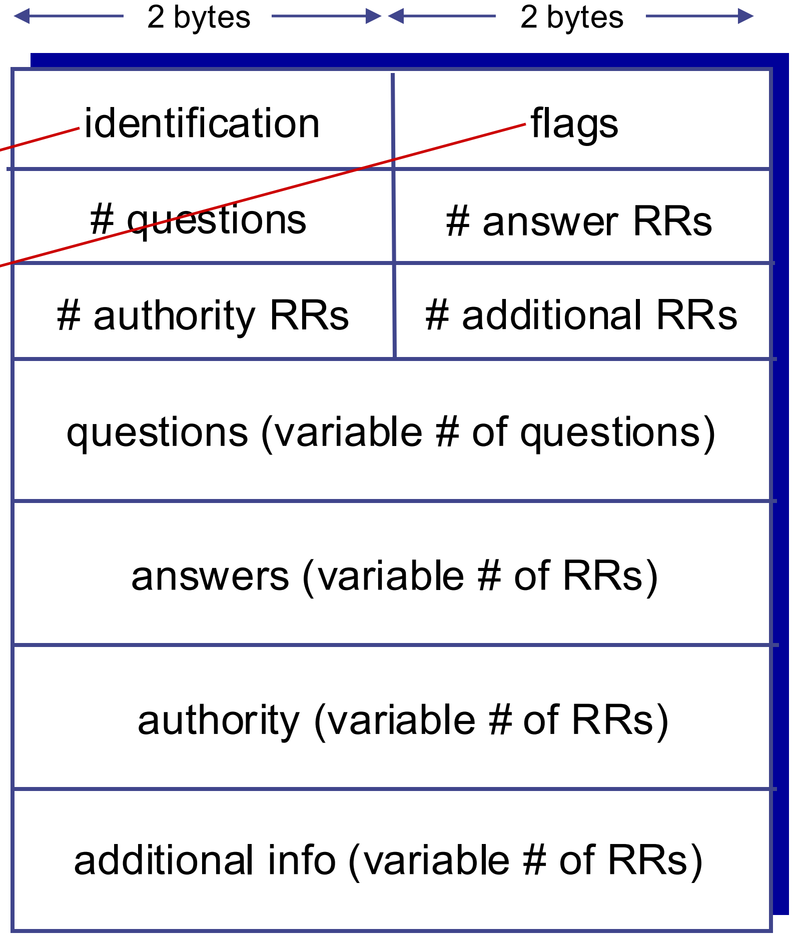
- **value** is name of mailserver associated with **name**
- Recall: Mailserver can have same as a webserver

DNS protocol, messages

- **Query** and **reply** messages, both with same message format

Message header

- identification: 16-bit number reply to query uses same
- flags:
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative



2 bytes | 2 bytes

identification | flags

# questions | # answer RRs

# authority RRs | # additional RRs

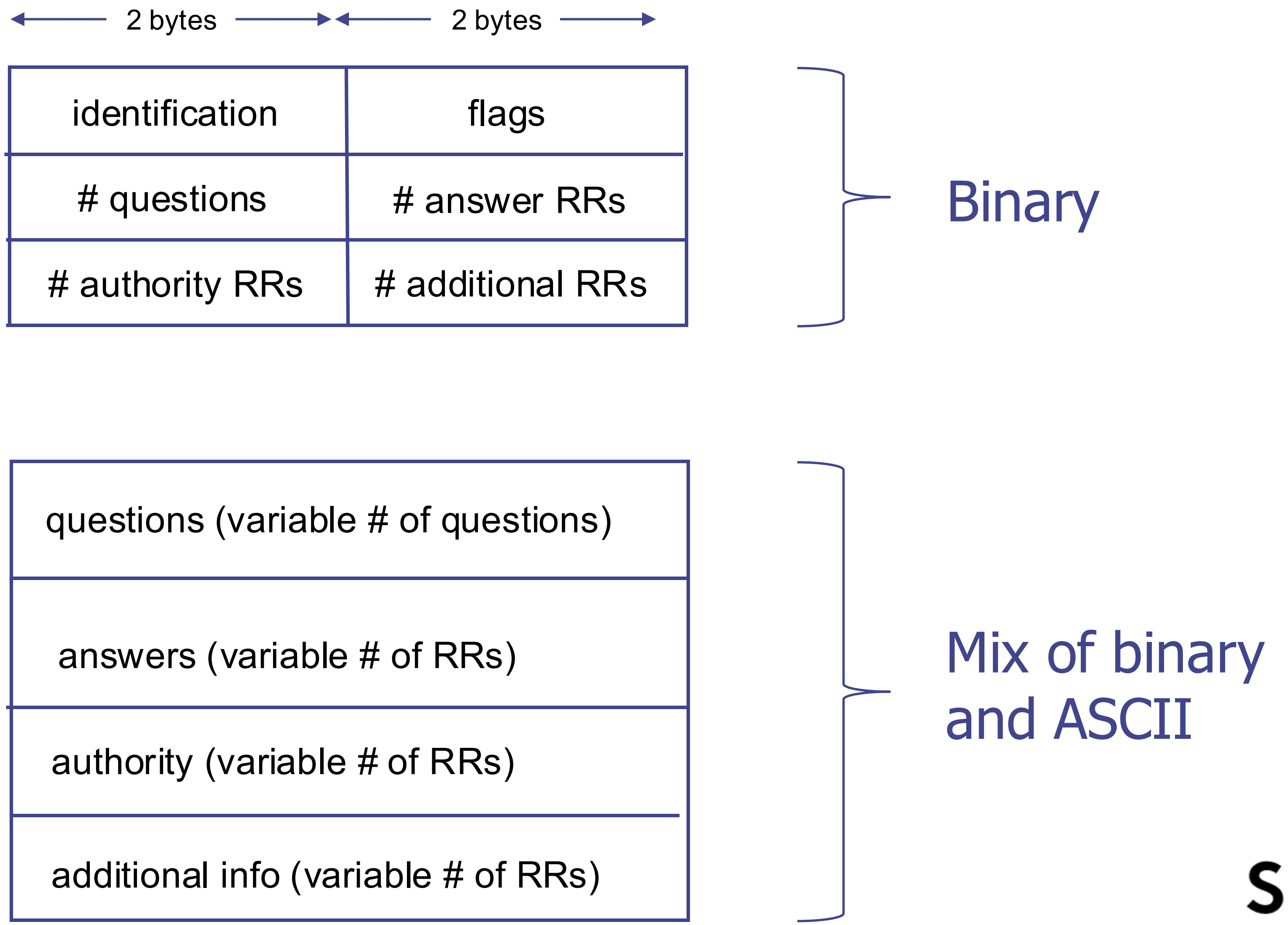
questions (variable # of questions)

answers (variable # of RRs)

authority (variable # of RRs)

additional info (variable # of RRs)

DNS protocol, messages



2 bytes	2 bytes
identification	flags
# questions	# answer RRs
# authority RRs	# additional RRs

Binary

questions (variable # of questions)

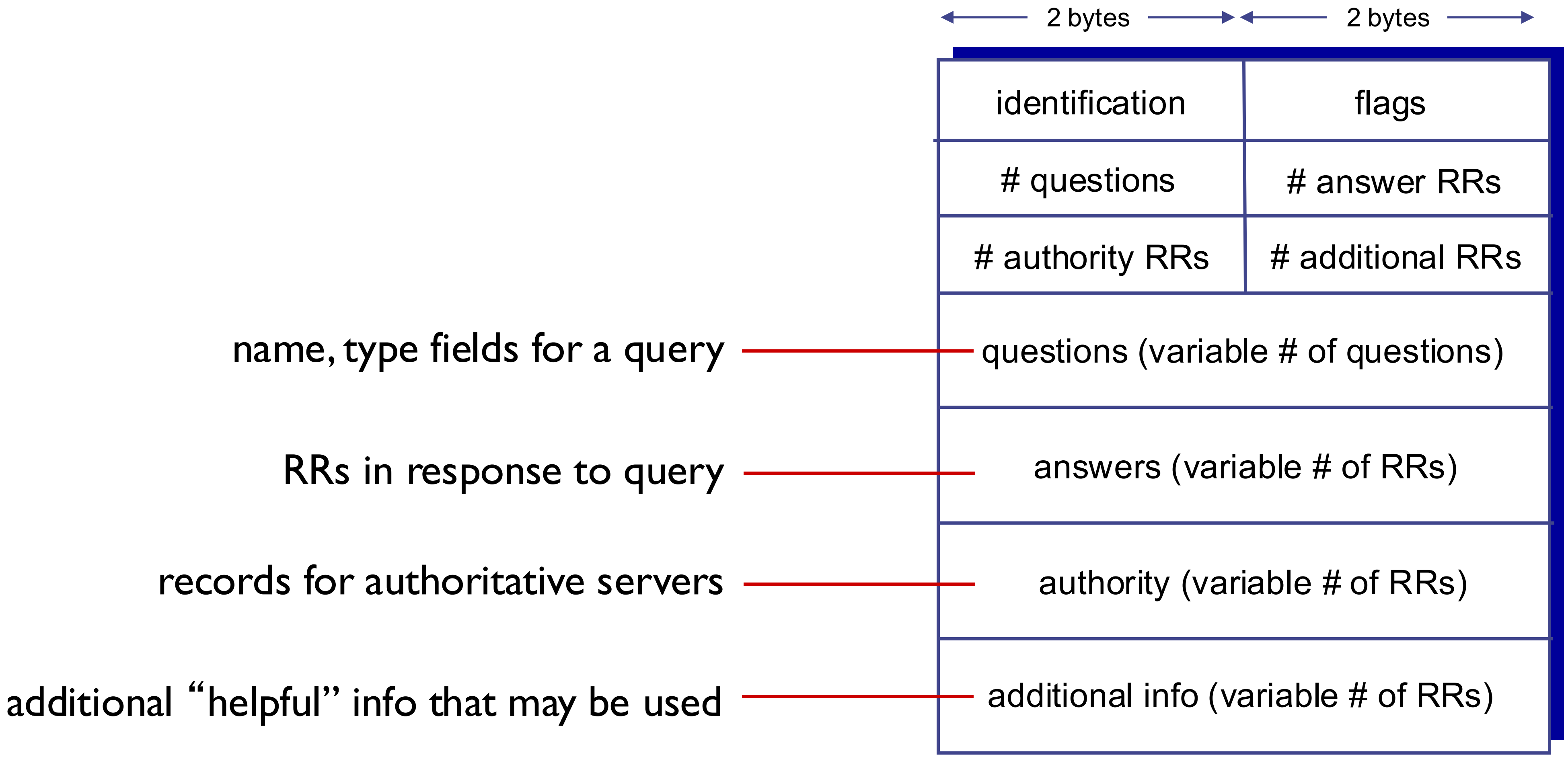
answers (variable # of RRs)

authority (variable # of RRs)

additional info (variable # of RRs)

Mix of binary and ASCII

DNS protocol, messages



2 bytes | 2 bytes

identification | flags

# questions | # answer RRs

# authority RRs | # additional RRs

name, type fields for a query — questions (variable # of questions)

RRs in response to query — answers (variable # of RRs)

records for authoritative servers — authority (variable # of RRs)

additional "helpful" info that may be used — additional info (variable # of RRs)

Inserting records into DNS

- Example: new startup “Network Utopia”

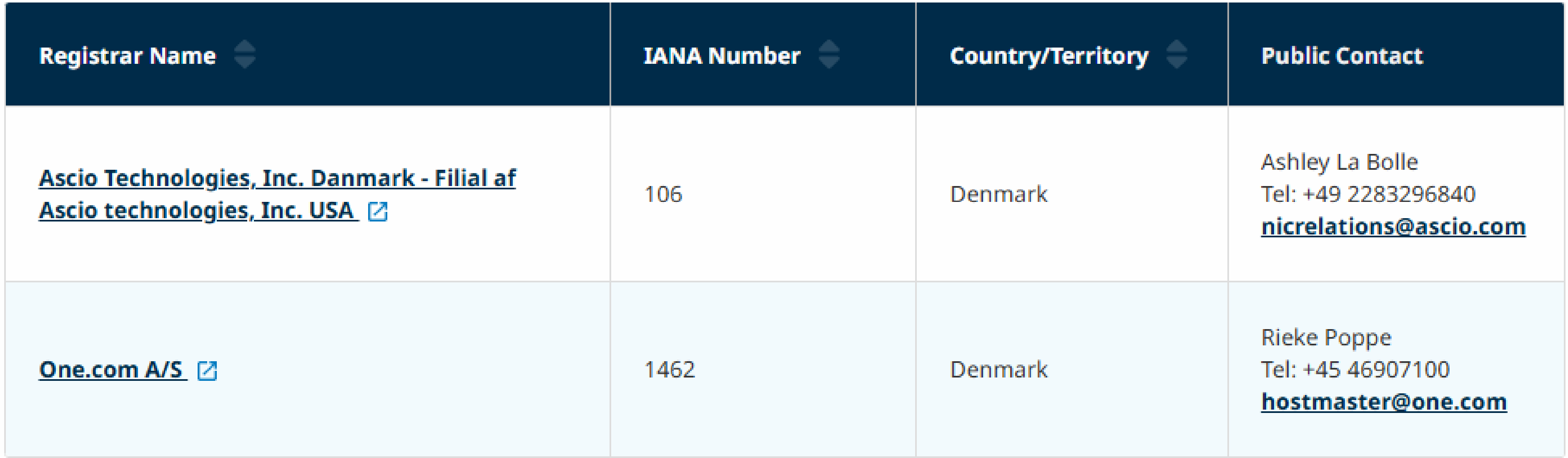
- Register name networkuptopia.com at *DNS registrar* (e.g., Network Solutions)
  - Provide names, IP addresses of authoritative name server (primary and secondary)
  - Registrar inserts two RRs into .com TLD server:

    ```
    (networkutopia.com, dns1.networkutopia.com, NS)
     (dns1.networkutopia.com, 212.212.212.1, A)
    ```

- Create authoritative server type A record for www.networkuptopia.com; type MX record for networkutopia.com

- In reality you can allow third party DNS providers to be your authoritative DNS server

Danish Registrars.



Registrar Name	IANA Number	Country/Territory	Public Contact
Ascio Technologies, Inc. Danmark - Filial af Ascio technologies, Inc. USA	106	Denmark	Ashley La Bolle Tel: +49 2283296840 nicrelations@ascio.com
One.com A/S	1462	Denmark	Rieke Poppe Tel: +45 46907100 hostmaster@one.com

Danish Registrars. Kilde: www.internic.net 09/07-2025

IP-addresses - Reminder

IPv4:
32 bit
ddd.ddd.ddd.ddd
e.g.: 192.168.2.10

IPv6:
128 bit
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
e.g.: 2a02:cb40:200::1c4 = 2a02:cb40:200:0:0:0:0:1c4
=
001010100000001011001011010000000000001000000000
000000000000000000000000000000000000000000000000
$00000000000000000000000111000100_2$

Wireshark Labs

- Lab 3: DNS
  - nslookup
  - ipconfig
  - See: https://public-dns.info/nameserver/dk.html for public DNS servers
    - ns.sni.dk works

Python Labs

- Python Lab 1: Getting started
  - Helpful tool for debugging later python labs

- Supported programming language for exercises
  - Mandatory for related hand-ins

Next time

We move down the stack to the transport layer.
Read in chapter 3 (3.1-3.5) in the book (page215-289).