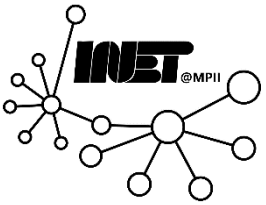
Application Layer
DNS

Prof. Anja Feldmann, Ph.D.

Dr. Oliver Gasser

(Based on slide deck of Computer Networking, 7th ed., Jim Kurose and Keith Ross.)

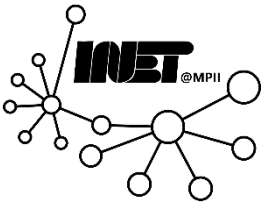


Domain Name System (DNS)

- People have many identifiers
  - SSN, name, Passport #
- Internet hosts, routers
  - IP address (32/128 bit) — used for addressing datagrams
  - "Name", e.g., mpi-inf.mpg.de — used by humans

How to map between IP addresses and name?

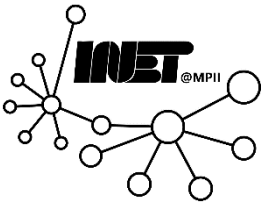
- Domain Name System (DNS) RFC1034/RFC1035, RFC3007,…



Domain Name System (DNS)

- **Distributed** database
  - Implemented in hierarchy of many **name servers**
- **Application-layer** protocol
  - Host, routers, name servers communicate to **resolve** names (address/name translation)

- Core Internet function implemented as application-layer protocol
- Complexity at network's "edge"



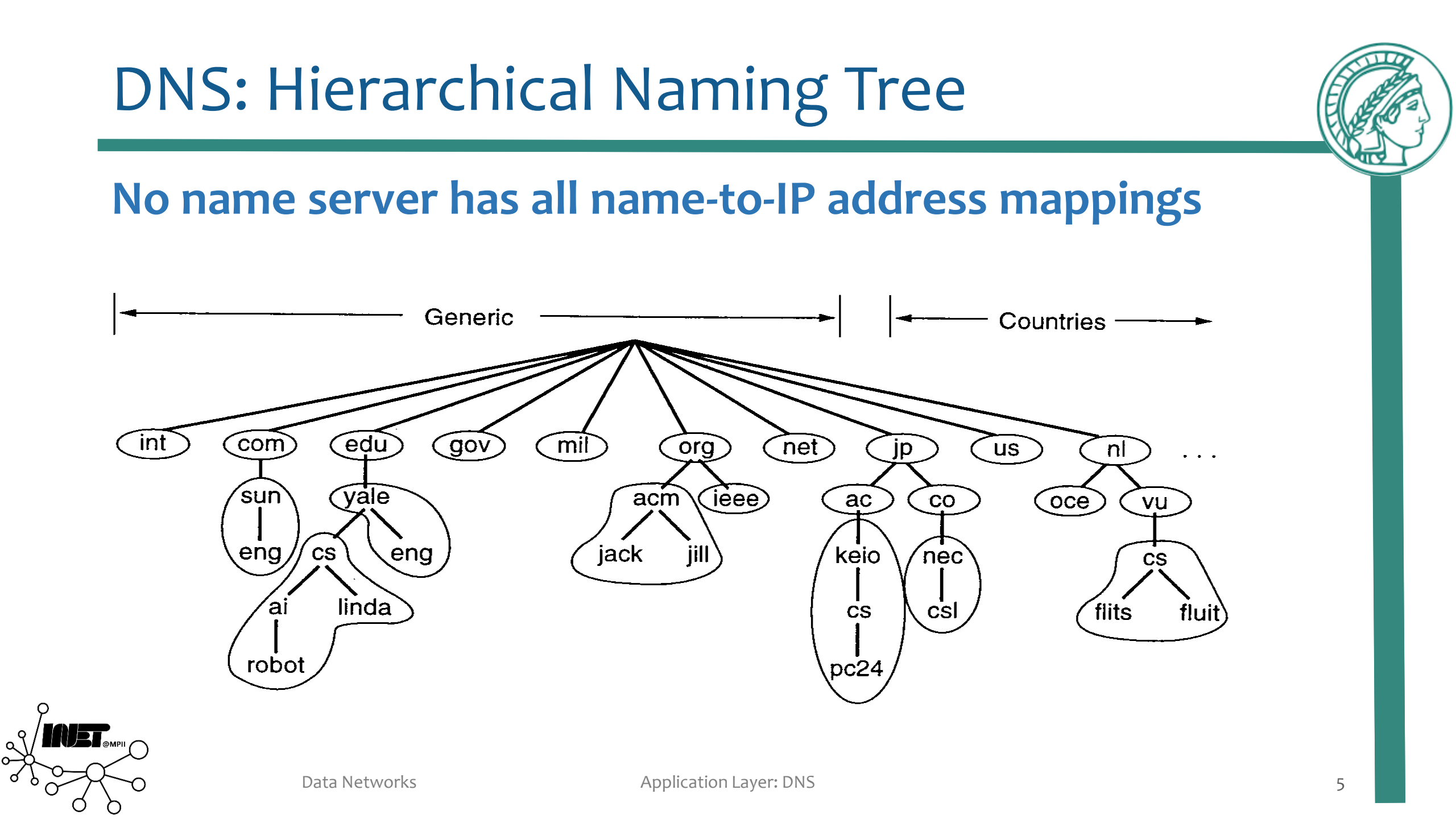
Domain Name System (DNS)

Why not centralize DNS?

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

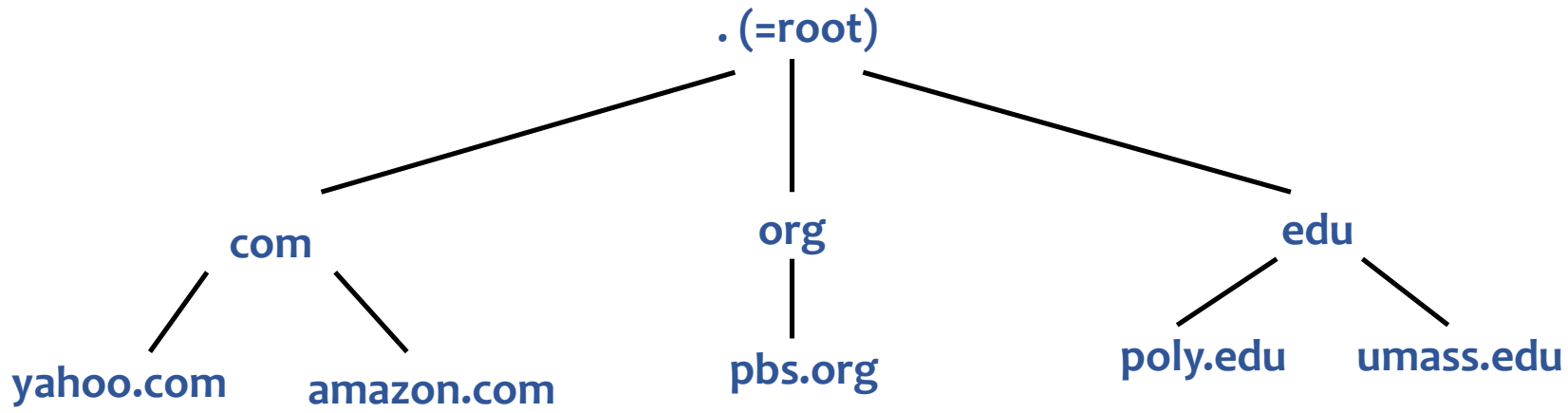
Does not scale!

DNS: Hierarchical Naming Tree



No name server has all name-to-IP address mappings

DNS: A Distributed, Hierarchical Database

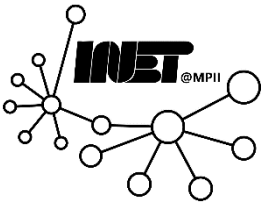


Client wants to get the IP address for www.amazon.com → first approximation:

1. Client's resolver queries **root** name server to find **com** name server
2. Client's resolver queries **com** name server to find **amazon.com** name server
3. Client's resolver queries **amazon.com** name server to get IP address for **www.amazon.com**

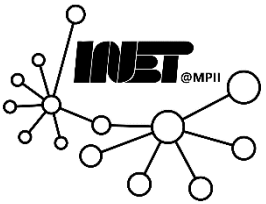
DNS: Name Servers vs. Resolvers

- **Name server**
  - **Authoritative** for a certain number of names
  - **Responds to DNS queries** issued by a resolver
  - Types: Root name server, TLD name server, SLD name server,…
- **Resolver**
  - **Issues DNS queries** to a name server to resolve a name to an IP address
  - Types: stub resolver, forwarding resolver, recursive resolver



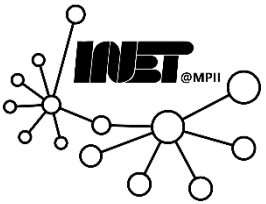
Authoritative Name Servers

- Root name servers
- Top-level domain servers
- Second-level domain servers
- …

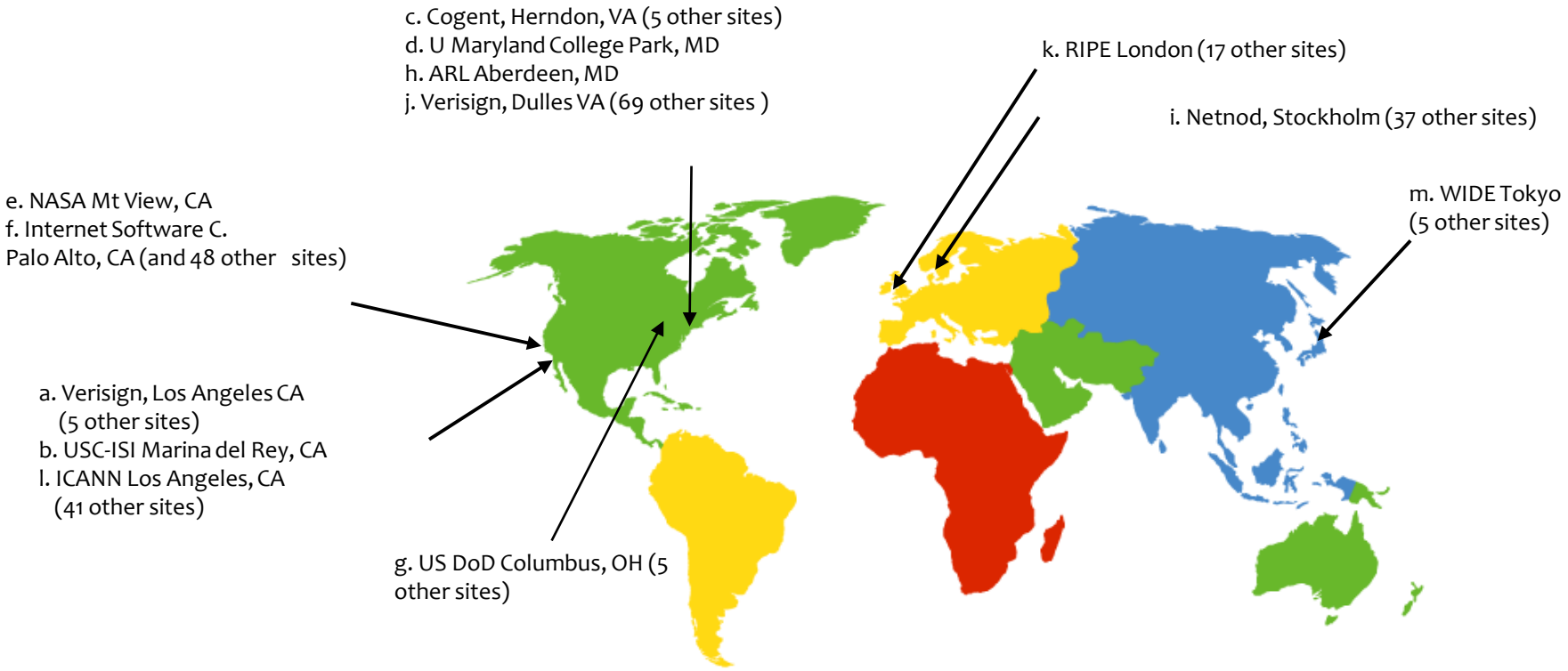


Root Name Servers

- Authoritative for **root zone** (.)
- Contain name server mappings for all top-level domains
- **Bootstrapping** problem: How does a resolver find the root servers?
  - Root hints file
  - Directly shipped with resolver software



Different Root Name Servers



- 13 logical root name "servers" worldwide
  - [a-m].root-servers.net
- Each logical "server" **replicated many times** (currently 1379 instances)

Top-Level Domain (TLD) Servers

- Responsible for **top-level domains**, i.e. domains just below the root zone

- Country code TLDs (ccTLDs): .de, .us, .in, .pk, .fr, .uk,…
- Generic TLDs (gTLDs): .com, .org, .net, .int, .edu, .gov, .mil
- "New gTLDs"
  - Launched in 2013
  - E.g.: .dev, .top, .online, .google, .nyc, .berlin, .saarland, …
  - More than 1300 new gTLDs
- Infrastructure TLD: .arpa

- Different TLD name servers are maintained by **different institutions**

Second-Level Domain (SLD),… Name Servers

- Responsible for **second-level domains**, i.e. domains just below the top-level domain zone
- E.g. example.com, amazon.com, uni-saarland.de,…

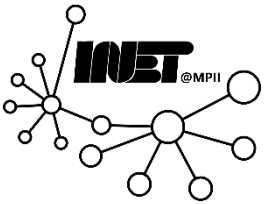
- What about co.uk?
  - Technically speaking a second-level domain
  - But: Treated by software as a top-level domain
  - co.uk is an **effective TLD** (eTLD)

- Different SLD name servers are maintained by **different institutions**

Resolvers

Do not strictly belong to the DNS hierarchy!

- Stub resolver
- Forwarding resolver
- Recursive resolver
- All resolvers can make use of **caching** of DNS answers

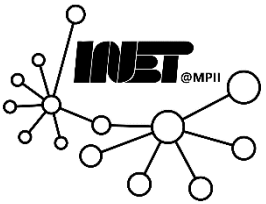


Stub Resolver

- "Dumb" resolver software **running on the local machine**
- Sends DNS queries to configured resolver
  - /etc/resolv.conf or "default name server" setting
- Returns answers to querying software

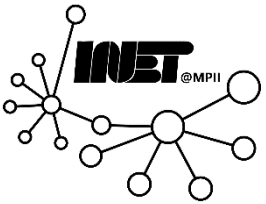
Forwarding Resolver

- Running e.g. on your home router or in infrastructure of large DNS services
- Acts as a **proxy**: receives queries (e.g. from a stub resolver), forwards queries to a recursive resolver
- Configured in resolver software
- Returns answers to querying resolver



Recursive Resolver

- Each ISP (residential ISP, company, university) has one
- Does the actual **heavy lifting**:
  - Receives queries from forwarding or stub resolver
  - Recursively queries root, TLD, SLD,… name servers
- Returns answer to querying resolver



DNS Records

DNS is a distributed database for 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 IP address of authoritative name server for this domain

Type=CNAME

for alias

Type=MX

for mail

Type=AAAA

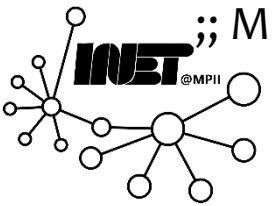
for IPv6

A Record: Example

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;mpi-inf.mpg.de.                IN      A

;; ANSWER SECTION:
mpi-inf.mpg.de.         7201    IN      A       139.19.86.161

;; Query time: 2722 msec
;; SERVER: 172.27.216.42#53(172.27.216.42)
;; WHEN: Thu Oct 25 15:47:03 CEST 2018
;; MSG SIZE  rcvd: 59
```

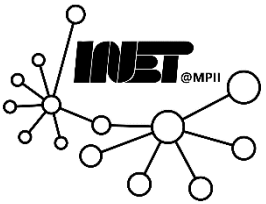


Internationalized Domain Names (IDNs)

- Allows use of **non-ASCII characters** in domain names
- E.g. umlauts, Chinese, Arabic, diacritics,…

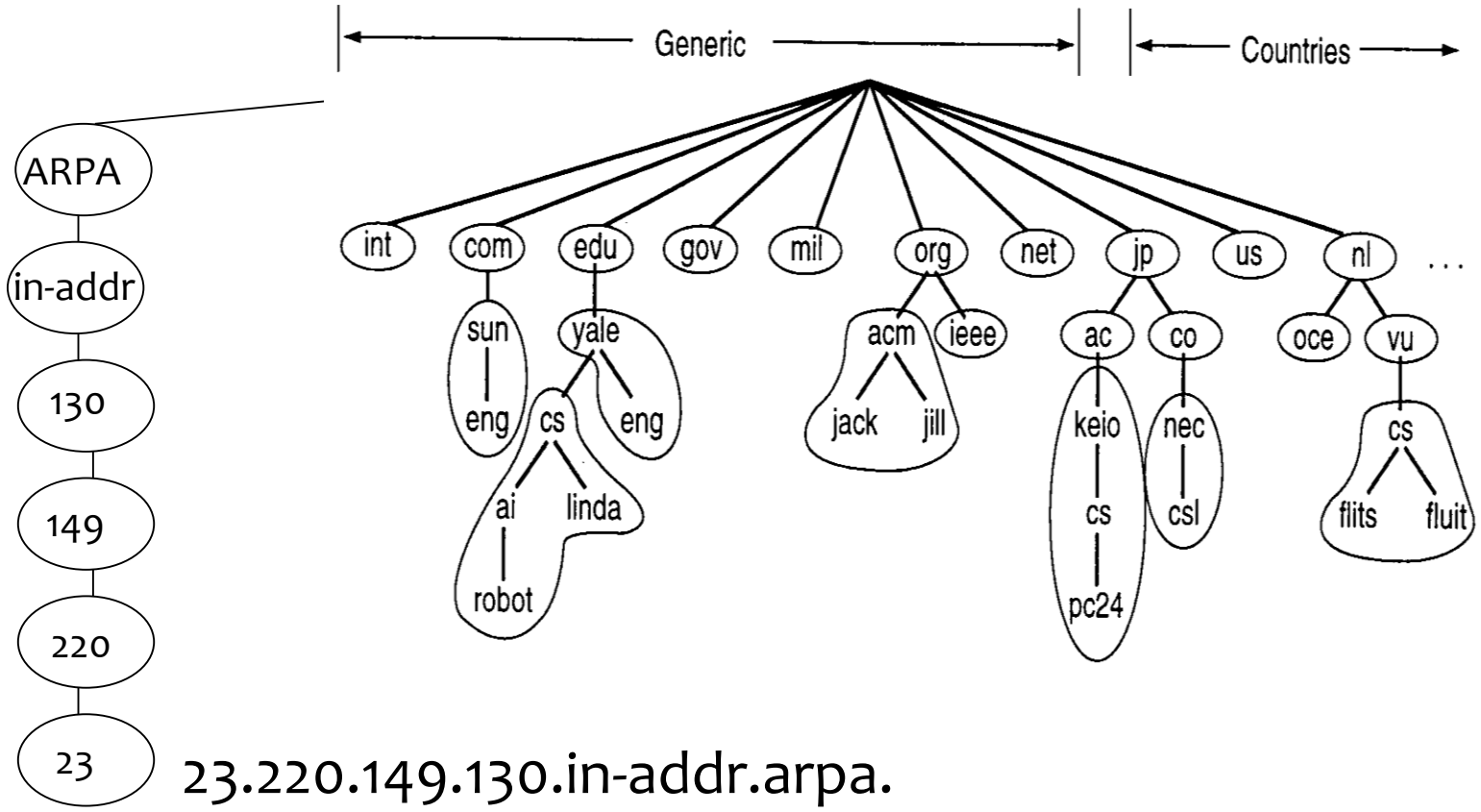
- Encoded as ASCII using **Punycode** (xn-encoding)
  - Makes use of generalized variable-length integers
  - bücher.example → xn--bcher-kva.example

- **Homograph** attacks
  - amazon.de vs. amazon.de
  - Hint: Cyrillic “a” on right domain



Mapping IP address to Names

Special domains: in-addr.arpa. / ip6.arpa.



23.220.149.130.in-addr.arpa.

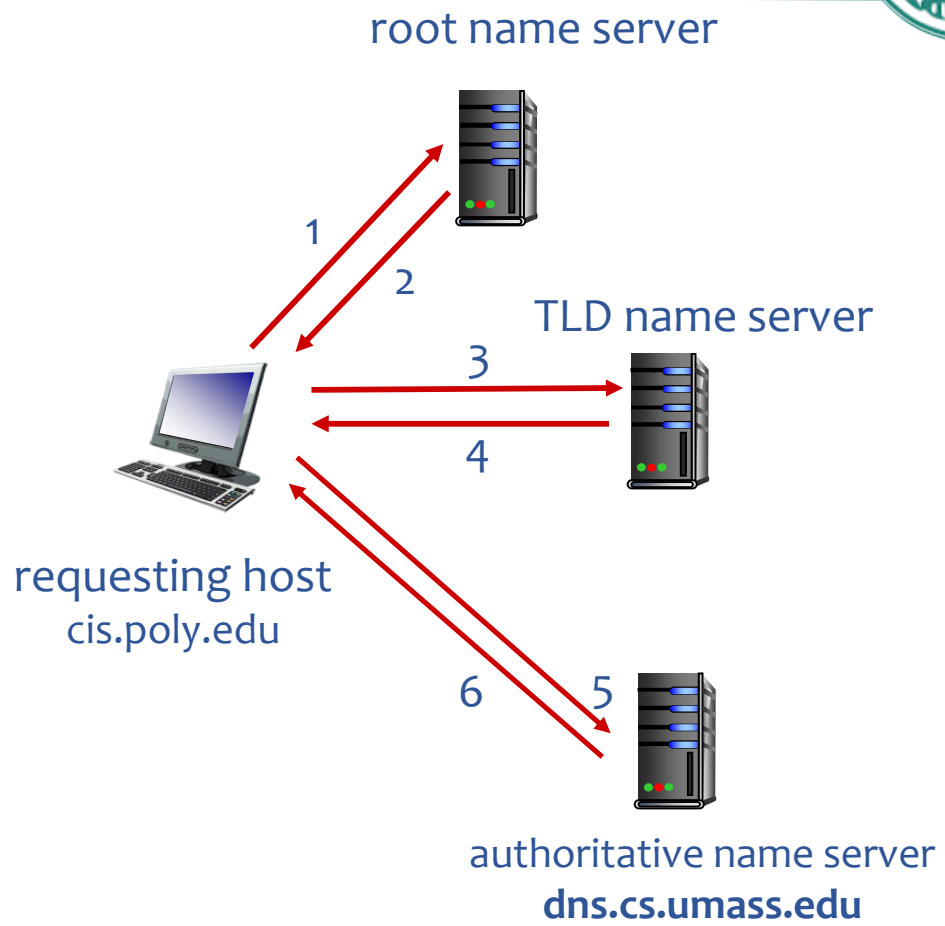
1.a.c.6.d.c.5.8.0.5.6.8.0.6.d.c.1.0.0.0.9.b.6.9.0.7.4.0.1.0.0.2.ip6.arpa.

DNS Name Resolution Iterative Example

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

Iterative query:

- Contacted server replies with name of server to contact
- "I don't know this name, but ask this server"

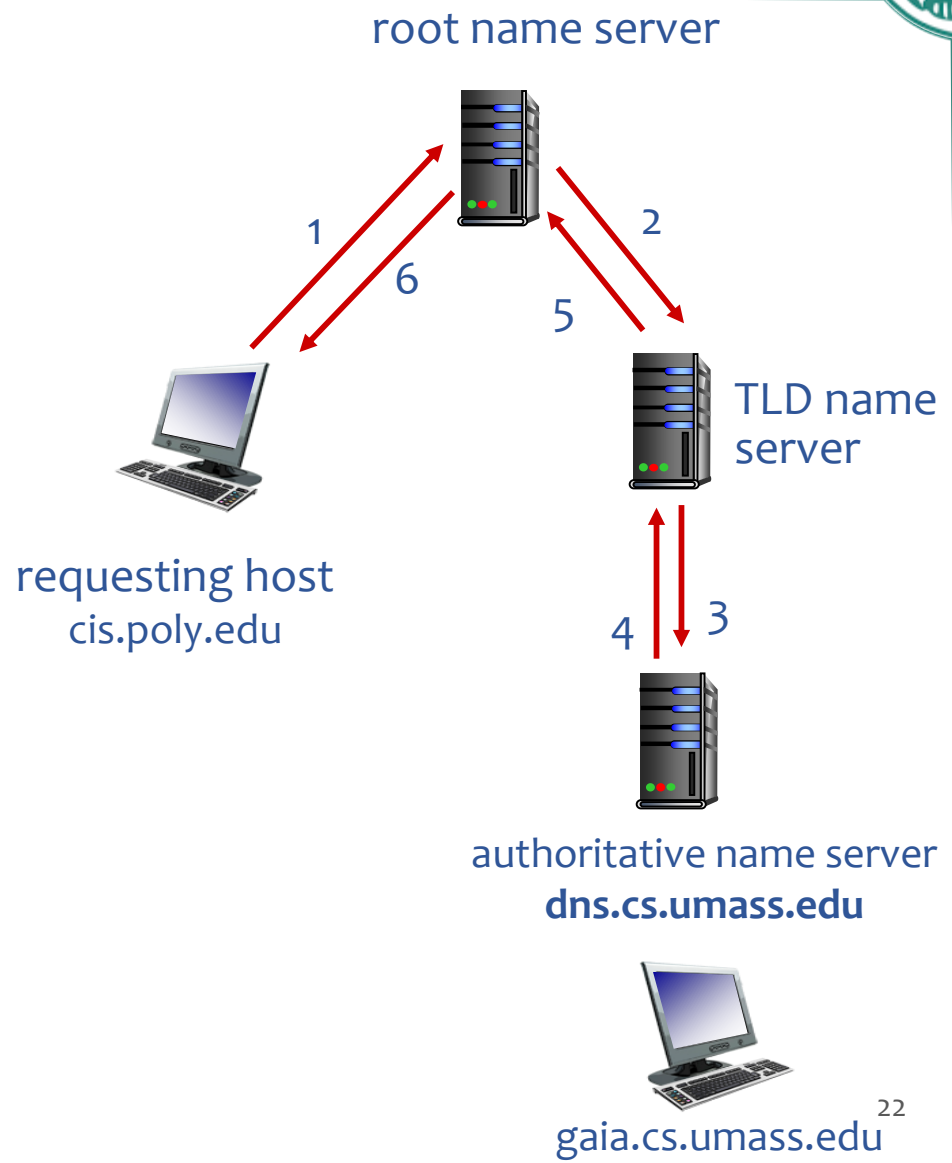


DNS Name Resolution Recursive Example

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

Recursive query:

- Puts burden of name resolution on contacted name server
- Heavy load at upper levels of hierarchy?

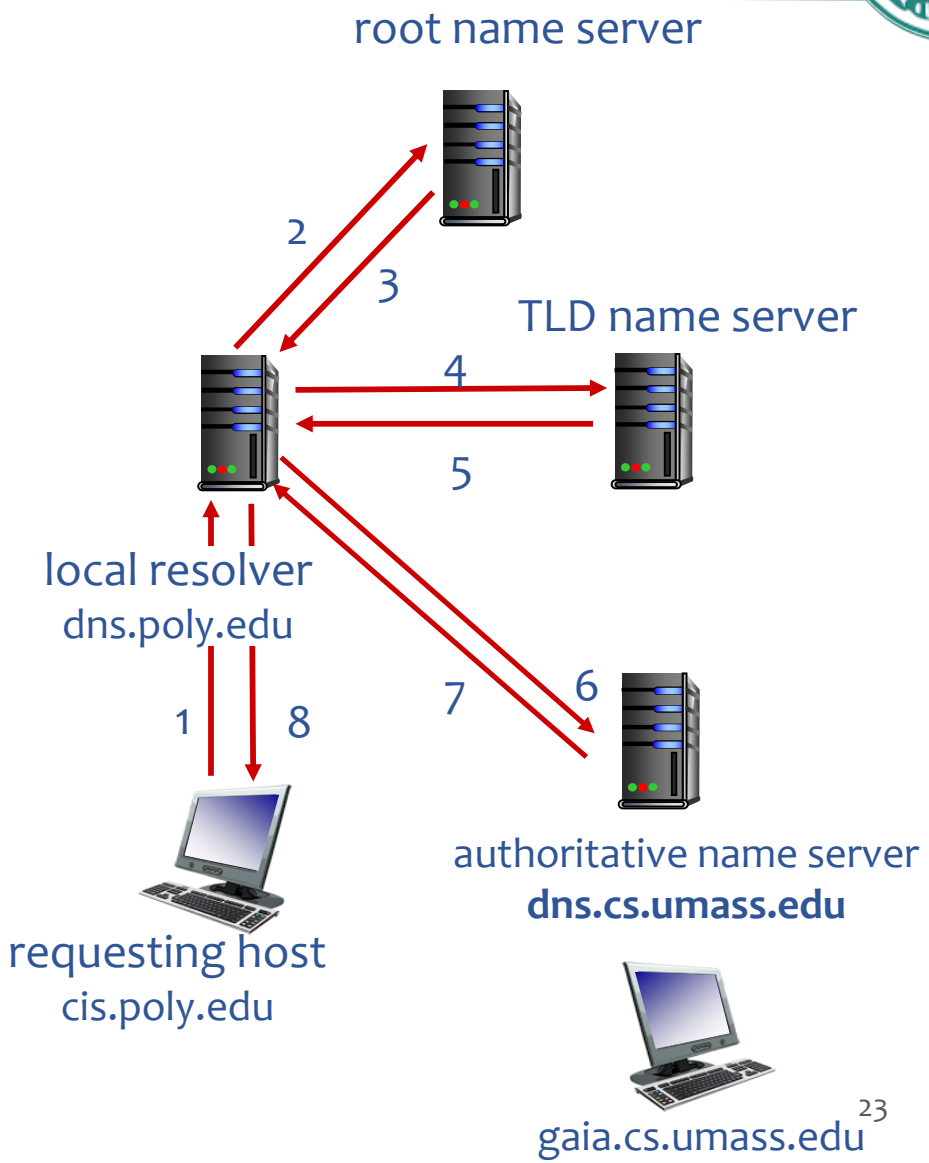


DNS Name Resolution Real-World Example

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

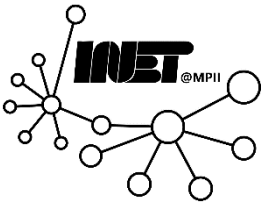
Mix of iterative and recursive queries:

- Requesting host sends a recursive query
- Local resolver uses iterative queries



DNS: Caching and Updating Records

- Once (any) resolver **learns** mapping, it **caches** it
  - Cached 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 (best effort name-to-address translation!)
  - If name host changes IP address, may not be known Internet-wide until all TTLs expire
- Update/notify mechanisms proposed IETF standard
  - RFC 2136



Attacking the DNS

Bombard root servers with traffic

- Not successful to date
- Traffic filtering
- Resolvers cache IPs of TLD servers, allowing root server bypass

Bombard TLD servers

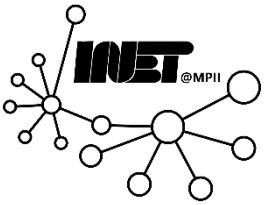
- Potentially more dangerous

Redirect attacks

- Man-in-the-middle
- Intercept queries
- DNS poisoning
- Send bogus replies to DNS server, which caches them

Exploit DNS for DDoS

- Send queries with spoofed source address: target IP
- Use DNS amplification



Recap

- DNS
  - Distributed, hierarchical database for names
- Authoritative name servers
  - Store mapping between name and address
  - Answer queries from resolvers
- Resolvers
  - Query authoritative name servers to learn mapping