Computer Networks and Internet Technology

2021W703033 VO Rechnernetze und Internettechnik
Winter Semester 2021/22

Jan Beutel

universität innsbruck

With materials and inspiration from Scott Shenker, Jennifer Rexford, Phil Levis, Adrian Perrig and Laurent Vanbever

Communication Networks and Internet Technology

Recap of last weeks lecture

Congestion Control

Congestion control aims at solving three problems

#1	bandwidth estimation	How to adjust the bandwidth of a single flow to the bottleneck bandwidth? could be 1 Mbps or 1 Gbps…
#2	bandwidth adaptation	How to adjust the bandwidth of a single flow to variation of the bottleneck bandwidth?
#3	fairness	How to share bandwidth “fairly" among flows, without overloading the network

Congestion control differs from flow control

both are provided by TCP though

Flow control	prevents one fast sender from overloading a slow receiver
Congestion control	prevents a set of senders from overloading the network

The sender adapts its sending rate based on these two windows

Receiving Window RWND	How many bytes can be sent without overflowing the receiver buffer? based on the receiver input
Congestion Window CWND	How many bytes can be sent without overflowing the routers? based on network conditions
Sender Window	minimum(CWND, RWND)

The 2 key mechanisms of Congestion Control

detecting
congestion

reacting to
congestion

Detecting losses can be done using ACKs or timeouts, the two signal differ in their degree of severity

duplicated ACKs	mild congestion signal packets are still making it
timeout	severe congestion signal multiple consequent losses

The 2 key mechanisms of Congestion Control

detecting congestion	reacting to congestion

TCP approach is to gently increase when not congested and to rapidly decrease when congested

question

What increase/decrease function should we use?

it depends on the problem we are solving…

#2 bandwidth adaptation

How to adjust the bandwidth of a single flow to variation of the bottleneck bandwidth?

Initially, you want to quickly get a first-order estimate of the available bandwidth

Intuition	Start slow but rapidly increase until a packet drop occurs	
Increase policy	cwnd = 1	initially
	cwnd += 1	upon receipt of an ACK

Alternatives

	increase behavior	decrease behavior
AIAD	gentle	gentle
AIMD	gentle	aggressive
MIAD	aggressive	gentle
MIMD	aggressive	aggressive

In practice,

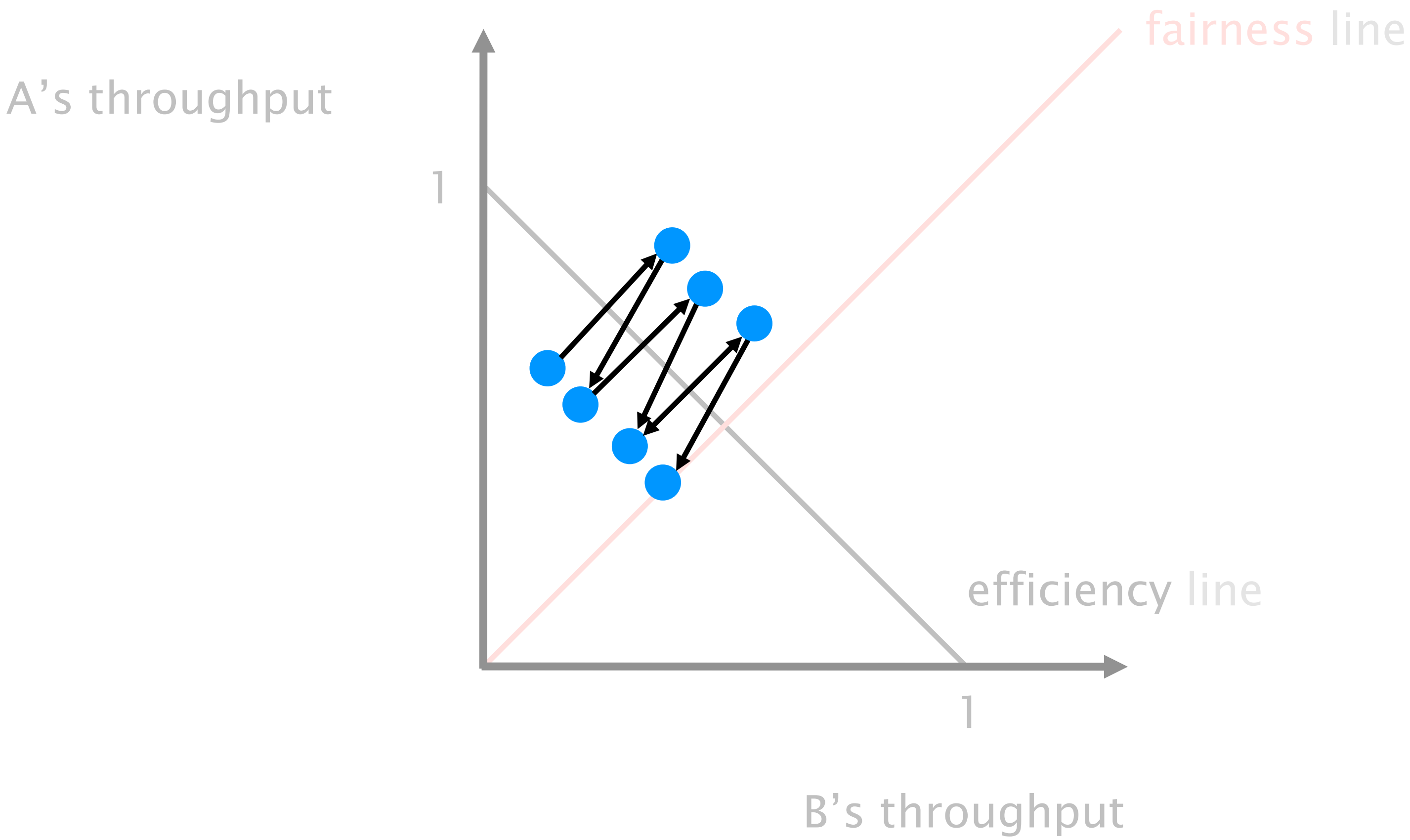
TCP implements AIMD

	increase behavior	decrease behavior
AIAD	gentle	gentle
AIMD	gentle	aggressive
MIAD	aggressive	gentle
MIMD	aggressive	aggressive

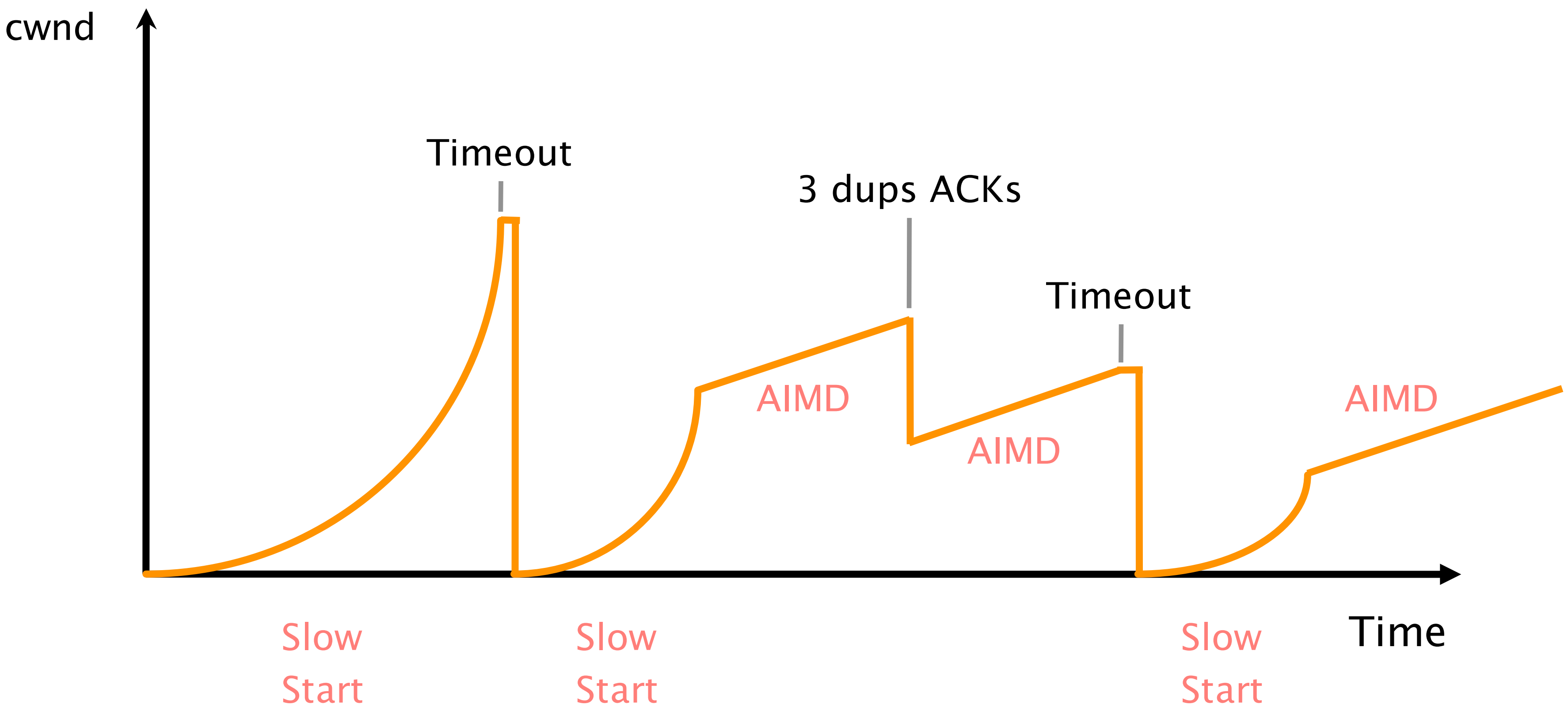
#3 fairness

How to share bandwidth “fairly" among flows, without overloading the network

AIMD converge to fairness and efficiency,
it then fluctuates around the optimum (in a stable way)



Congestion control makes TCP throughput look like a "sawtooth"



Communication Networks and Internet Technology

This weeks lecture

http

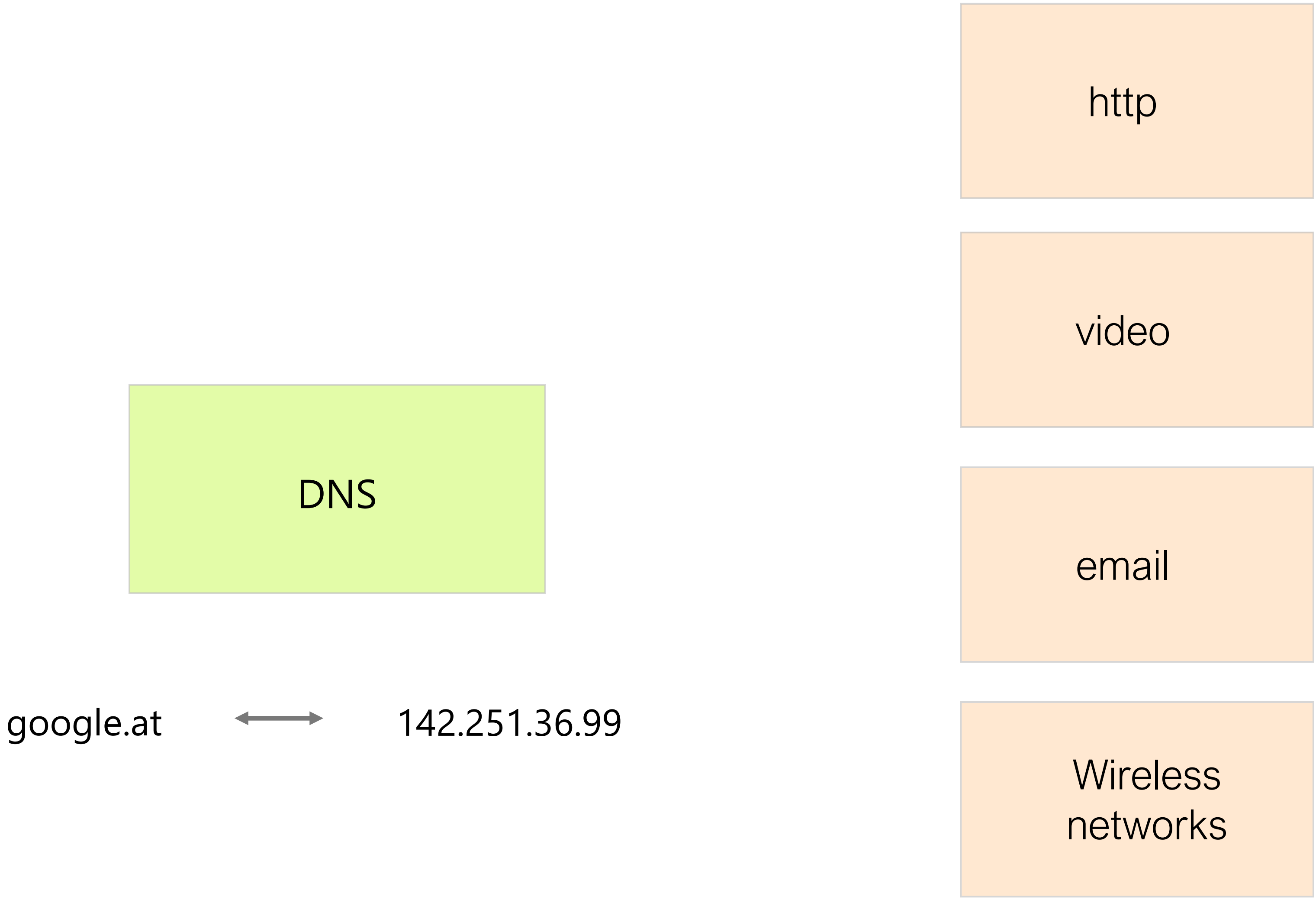
video

email

Wireless networks

DNS

google.at ⟷ 142.251.36.99


http
video
DNS
email
google.at
142.251.36.99
Wireless
networks

Internet has one global system for

- addressing hosts — IP

  by design

- naming hosts — DNS

  by "accident", an afterthought

Internet has one **global system** for

- **naming** hosts DNS

  by "accident", an afterthought

Using Internet services can be divided into four logical steps

step 1	A person has name of entity she wants to access	www.uibk.ac.at
step 2	She invokes an application to perform the task	Chrome
step 3	The application invokes DNS to resolve the name into an IP address	138.232.17.233
step 4	The application invokes transport protocol to establish an app-to-app connection	

The DNS system is a distributed database
which enables to resolve a name into an IP address

name	$\xleftrightarrow{\text{DNS}}$	IP address
www.uibk.ac.at		138.232.17.233

In practice,
names can be mapped to more than one IP

name	DNS ⟷	IP address	
www.uibk.ac.at		138.232.17.233	
www.netflix.com		52.18.41.127	(load-balancing)
		34.248.59.175	
		52.209.236.203	+5 more!

In practice,
IPs can be mapped by more than one name

name	DNS ⟷	IP address
www.uibk.ac.at		138.232.17.233
ifi-smokeping.uibk.ac.at		138.232.18.35
pbl.permasense.uibk.ac.at		138.232.18.35

How does one resolve a name into an IP?

initially

all host to address mappings
were in a file called hosts.txt

in /etc/hosts

problem

scalability in terms of query load & speed

management

consistency

availability

When you need… more flexibility,

you add… a layer of indirection

When you need… more scalability,

you add… a hierarchical structure

To scale,
DNS adopt three intertwined hierarchies

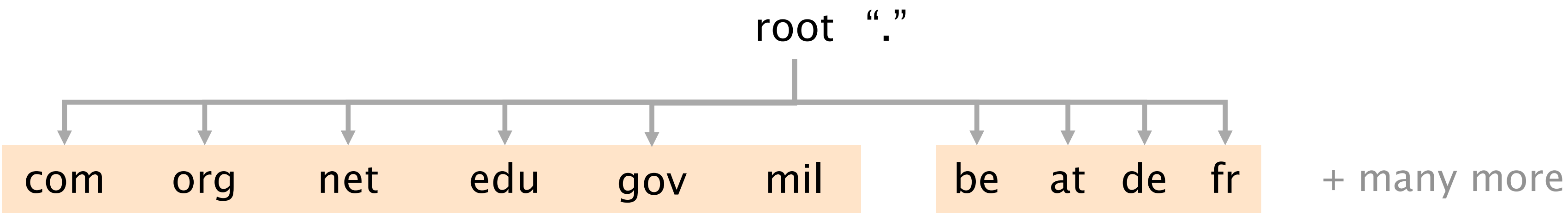
naming structure	hierarchy of addresses https://www.uibk.ac.at/informatik/
management	hierarchy of authority over names
infrastructure	hierarchy of DNS servers

naming structure

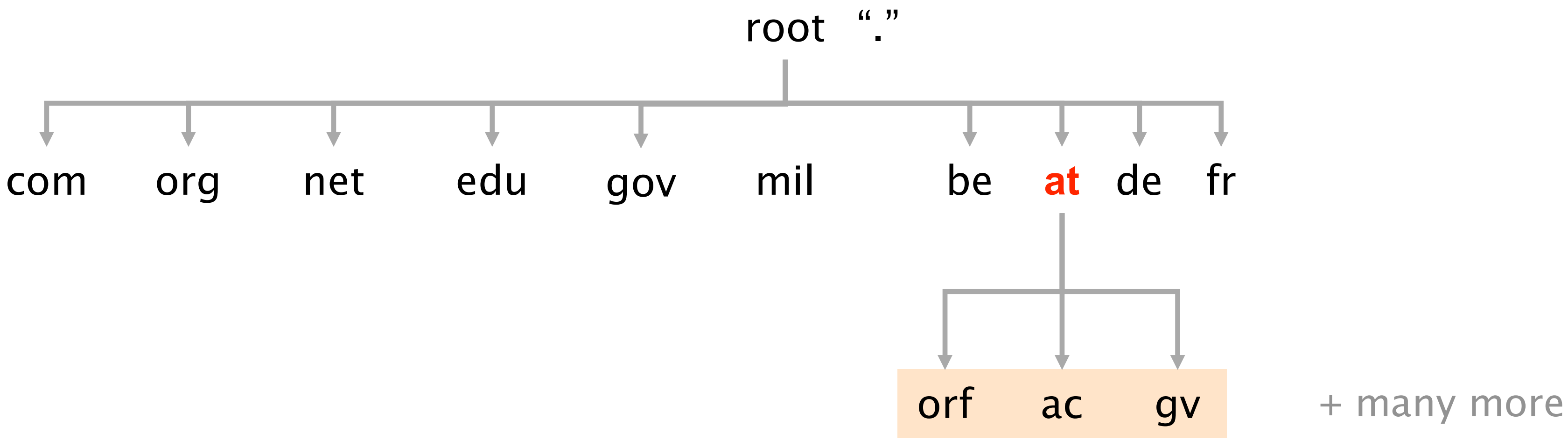
hierarchy of addresses

https://www.uibk.ac.at/informatik/

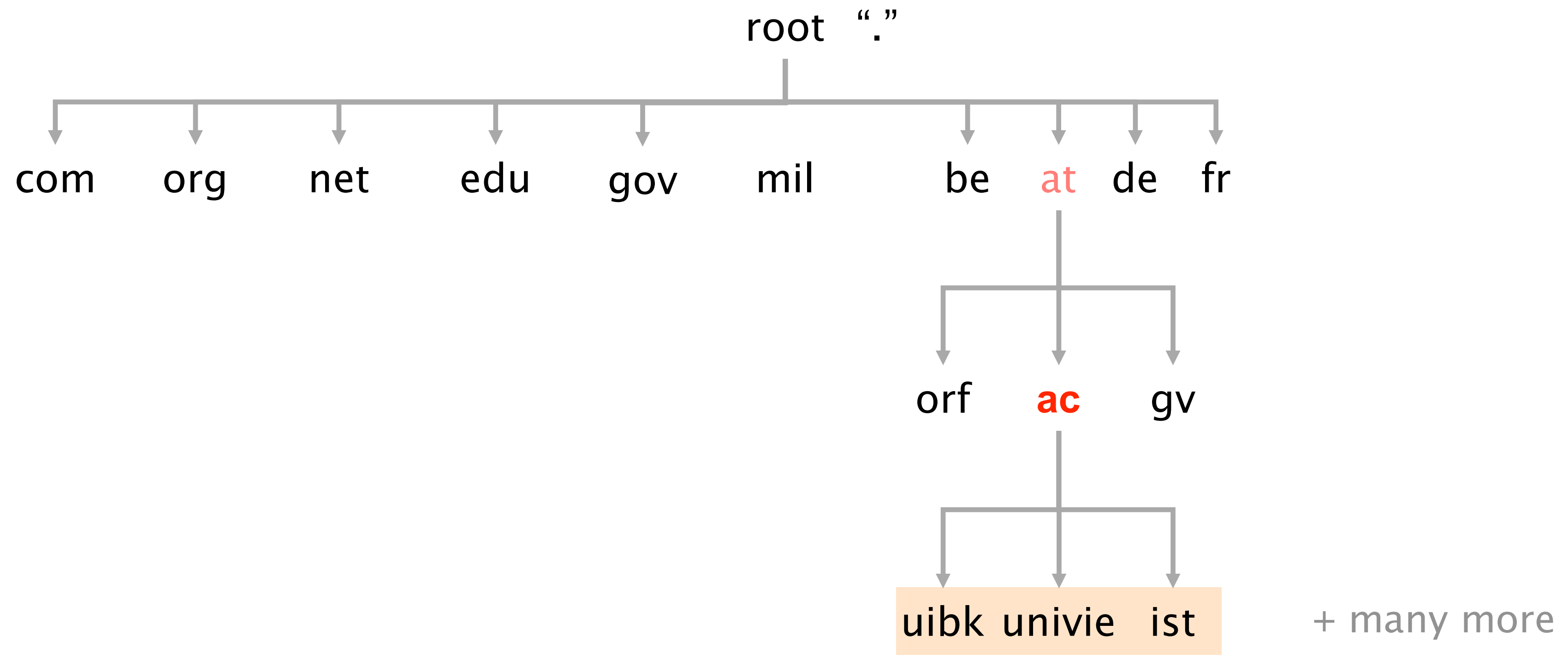
Top Level Domain (TLDs) sit at the top



Domains are subtrees



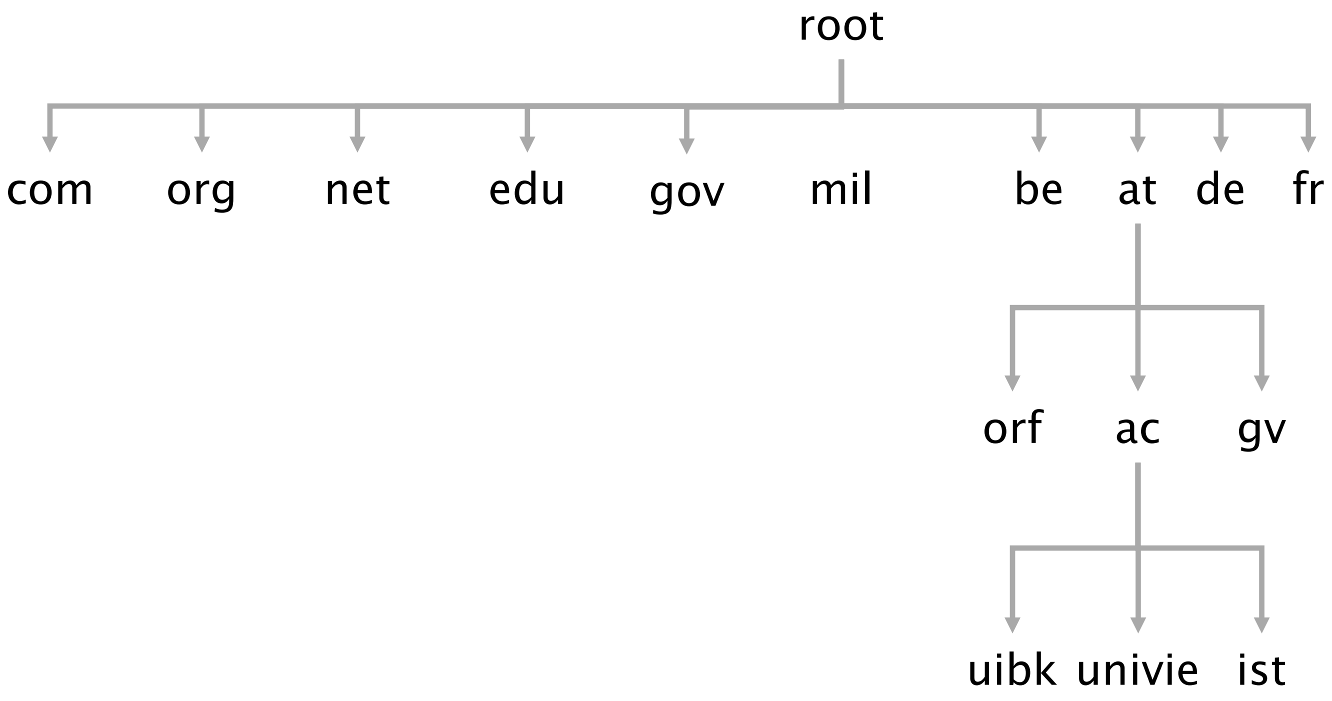
A name, *e.g.* uibk.ac.at, represents a leaf-to-root path in the hierarchy

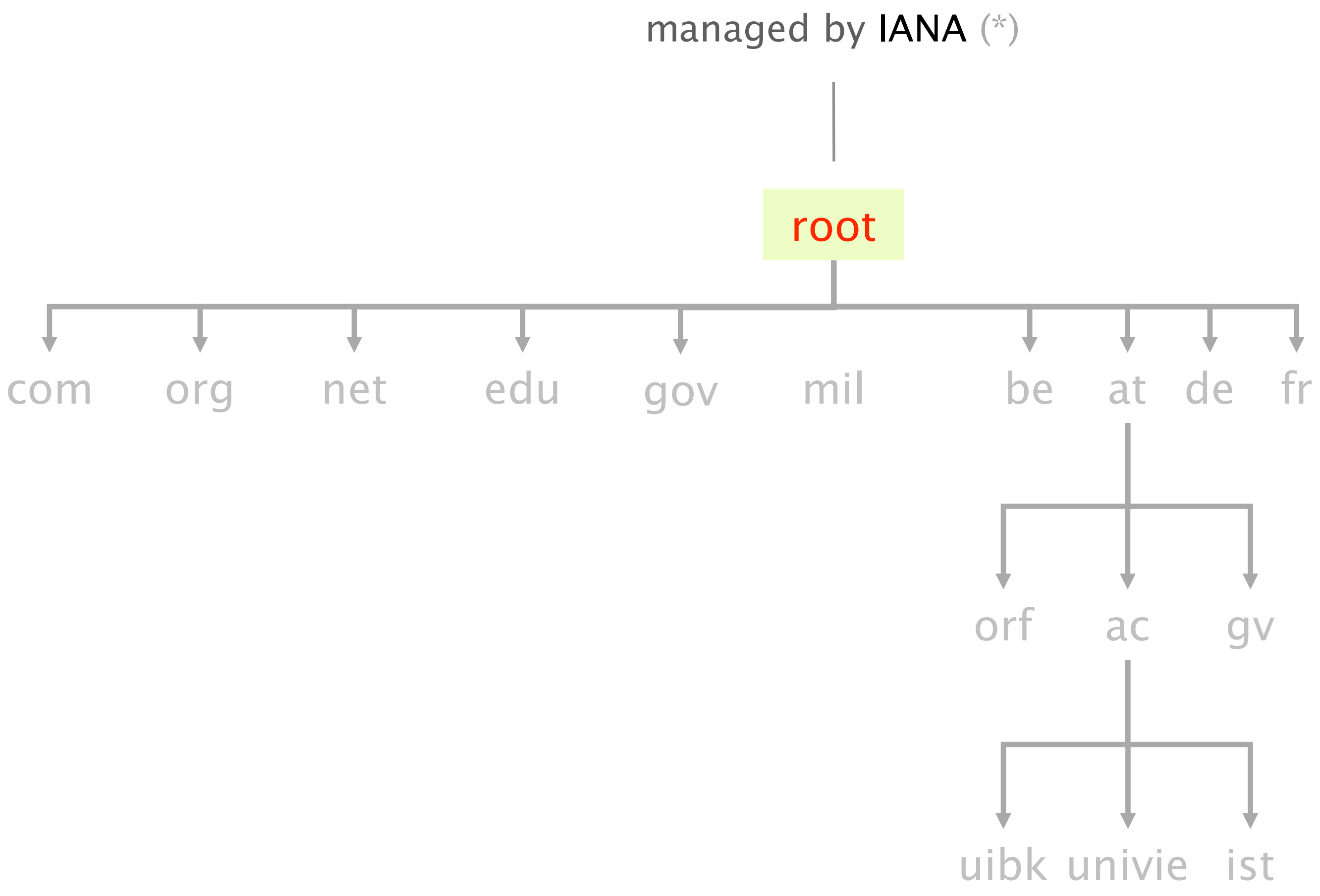


management

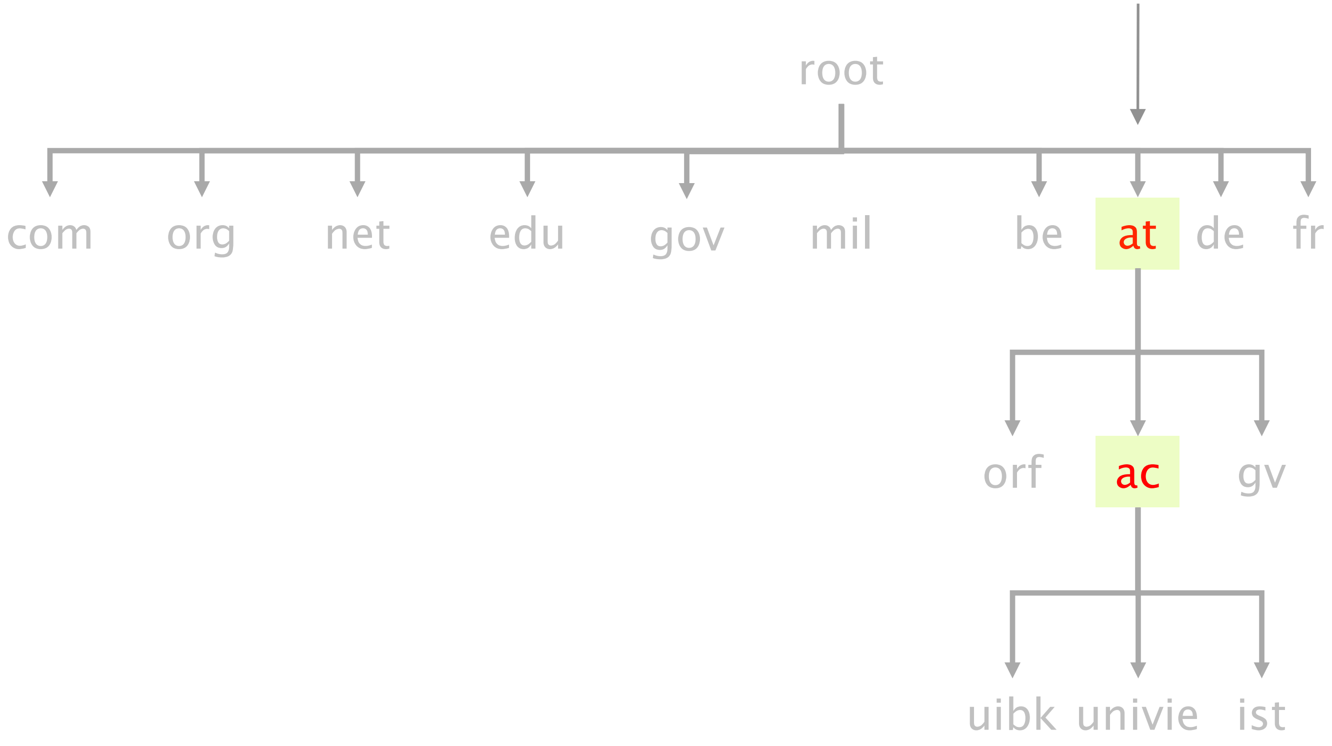
hierarchy of authority
over names

The DNS system is hierarchically administered





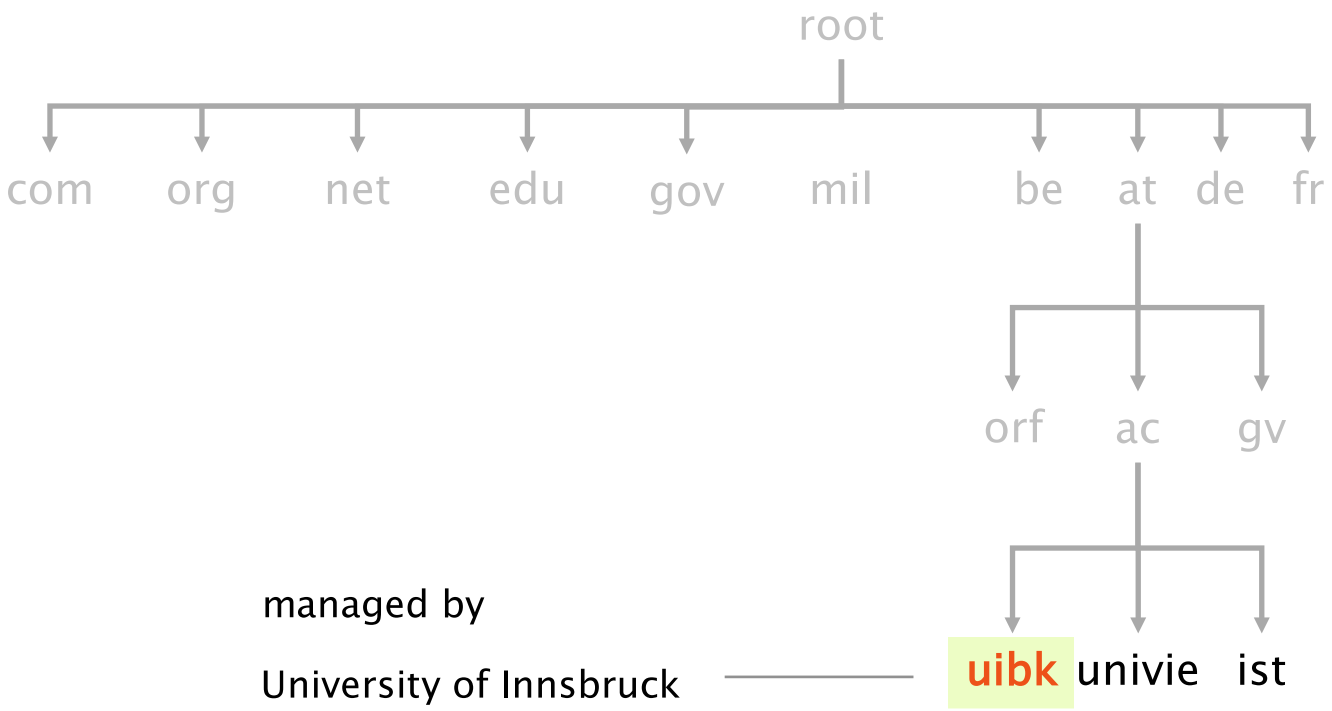
(*) see http://www.iana.org/domains/root/db managed by The Austrian Education & Research Network (*)



(*) see https://www.aco.net

managed by

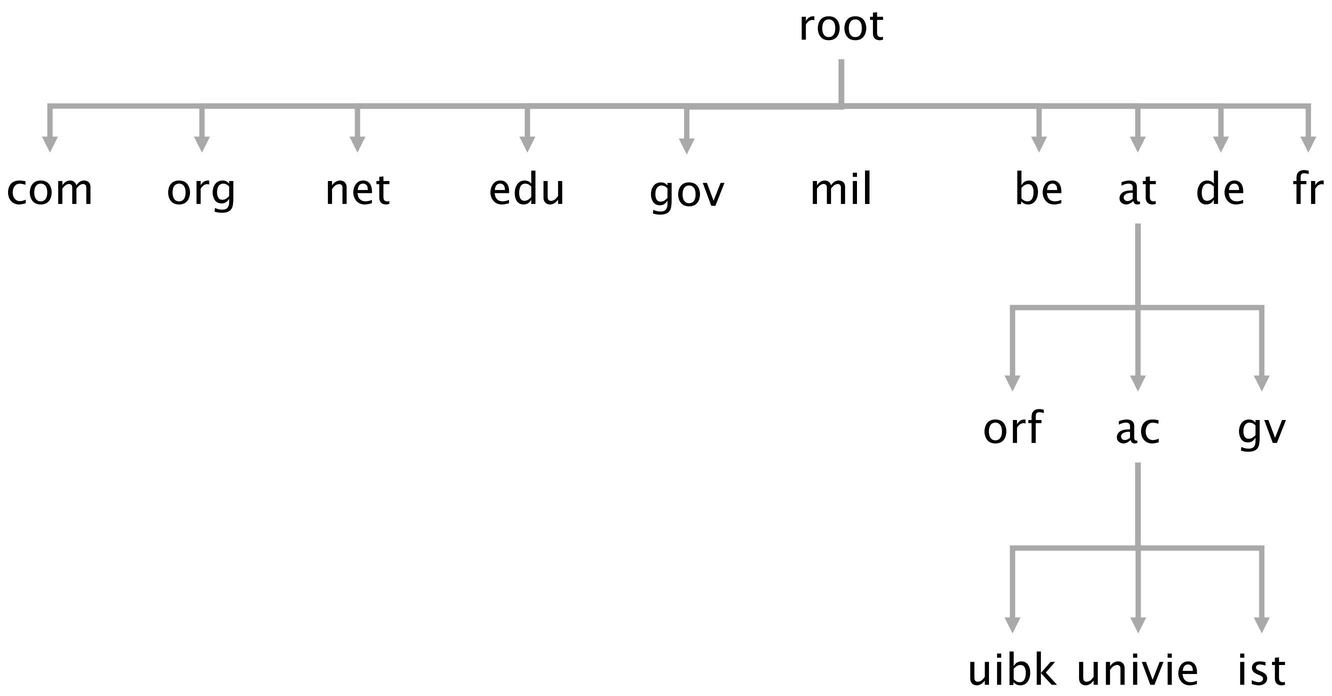
University of Innsbruck
Informatikdienste



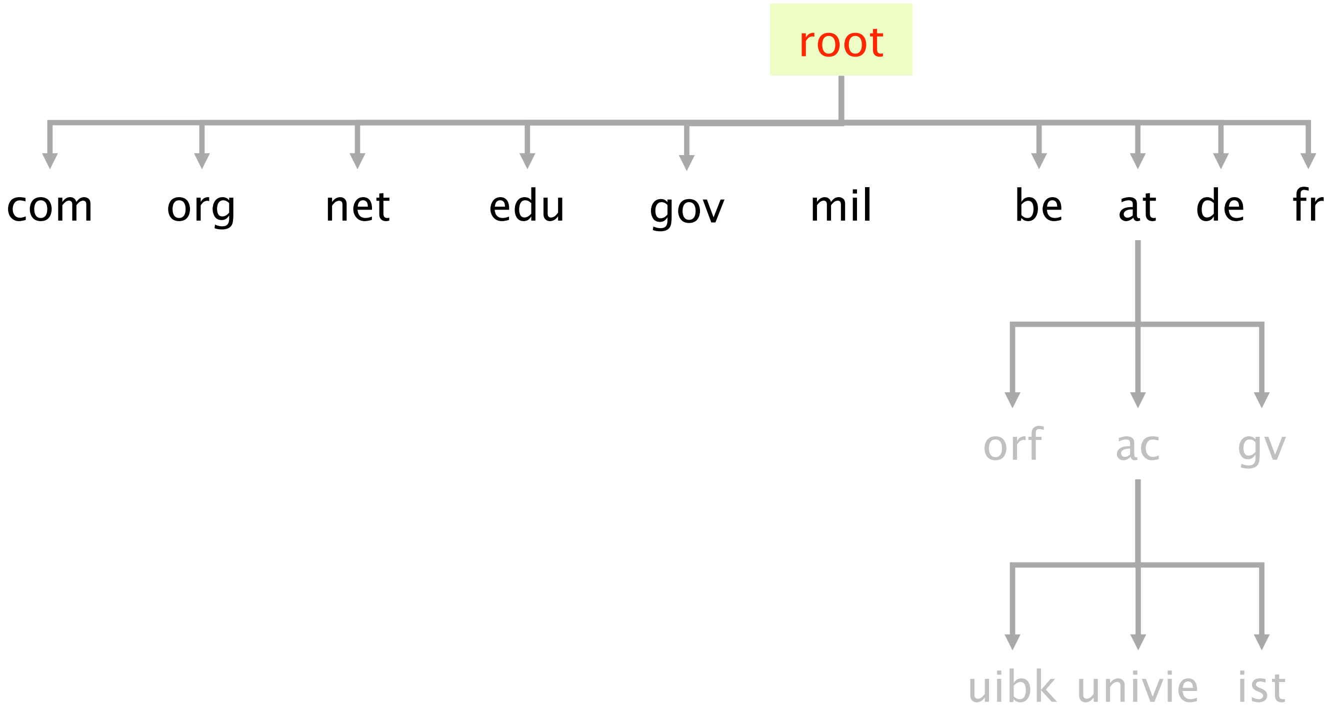
Hierarchical administration means
that name collision is trivially avoided

infrastructure hierarchy of DNS servers

The DNS infrastructure is hierarchically organized

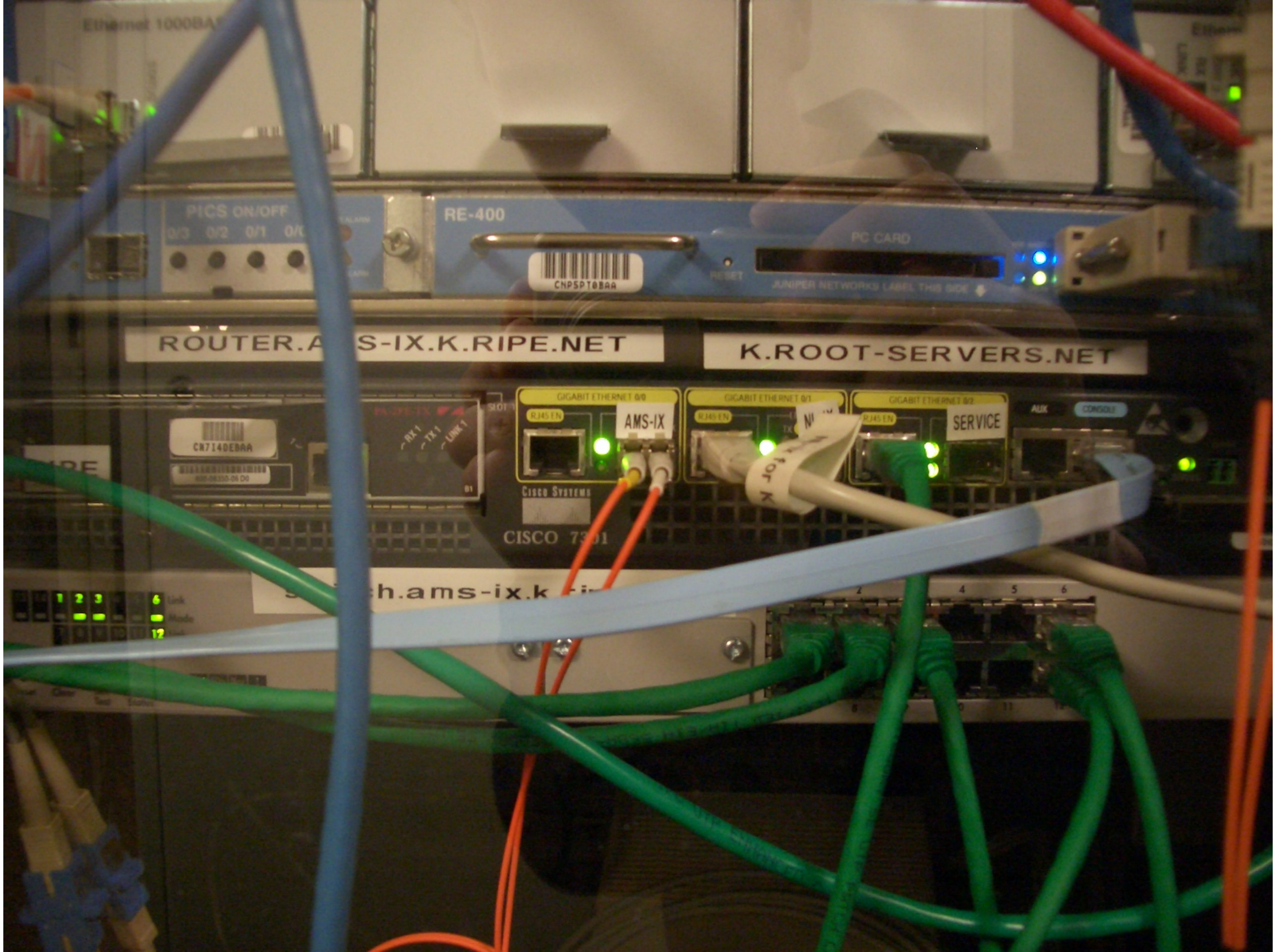


13 root servers (managed professionally) serve as root (*)



(*) see http://www.root-servers.org/

a. root-servers.net	VeriSign, Inc.
b. root-servers.net	University of Southern California
c. root-servers.net	Cogent Communications
d. root-servers.net	University of Maryland
e. root-servers.net	NASA
f. root-servers.net	Internet Systems Consortium
g. root-servers.net	US Department of Defense
h. root-servers.net	US Army
i. root-servers.net	Netnod
j. root-servers.net	VeriSign, Inc.
k. root-servers.net	RIPE NCC
l. root-servers.net	ICANN
m. root-servers.net	WIDE Project



PICS ON/OFF
RE-400
PC CARD
RESET
JUNIPER NETWORKS LABEL THIS SIDE
K.ROOT-SERVERS.NET
GIGABIT ETHERNET 0/0
AMS-IX
GIGABIT ETHERNET 0/2
SERVICE
AUX
CONSOLE

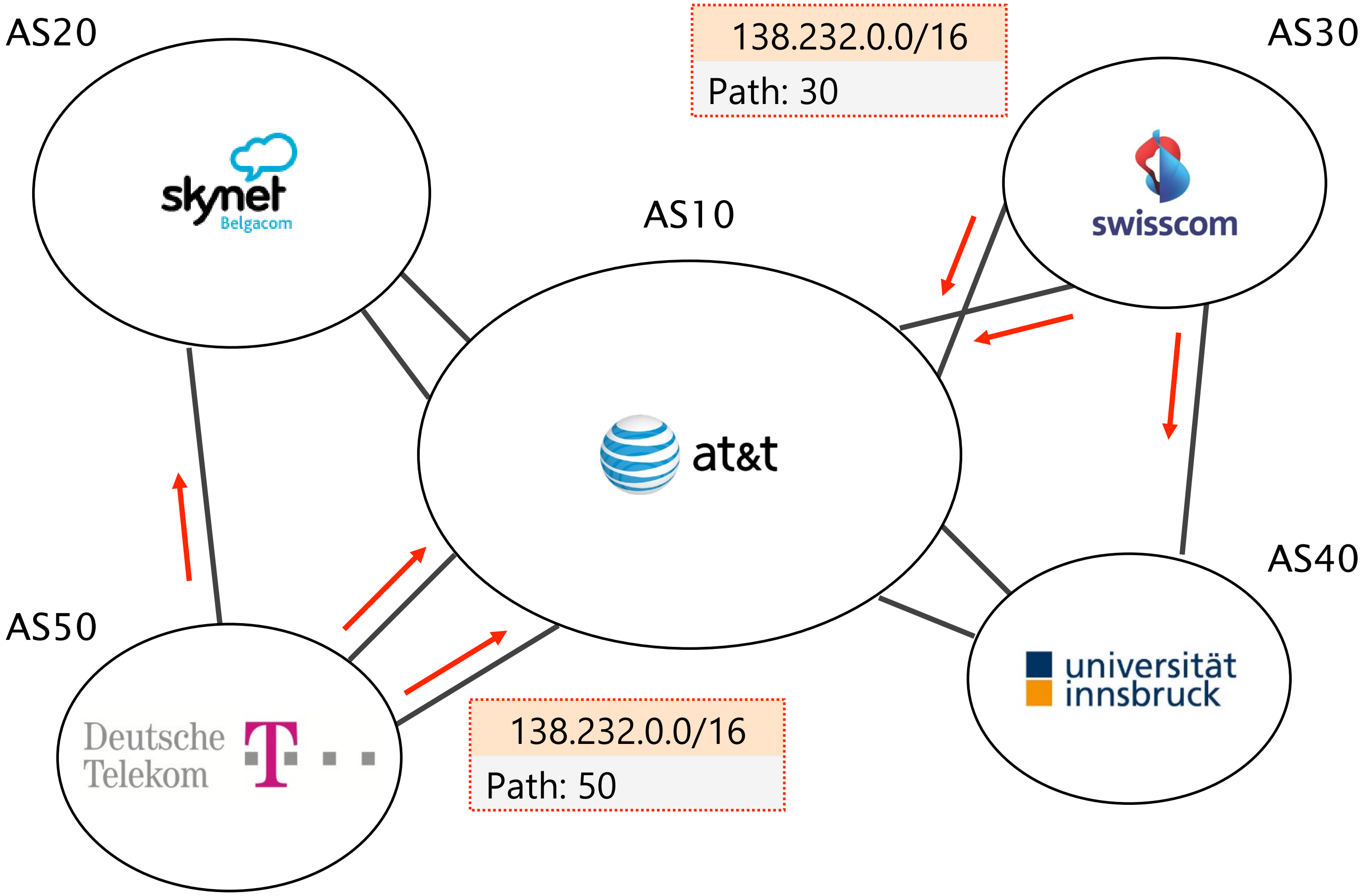
To scale root servers, operators rely on BGP anycast

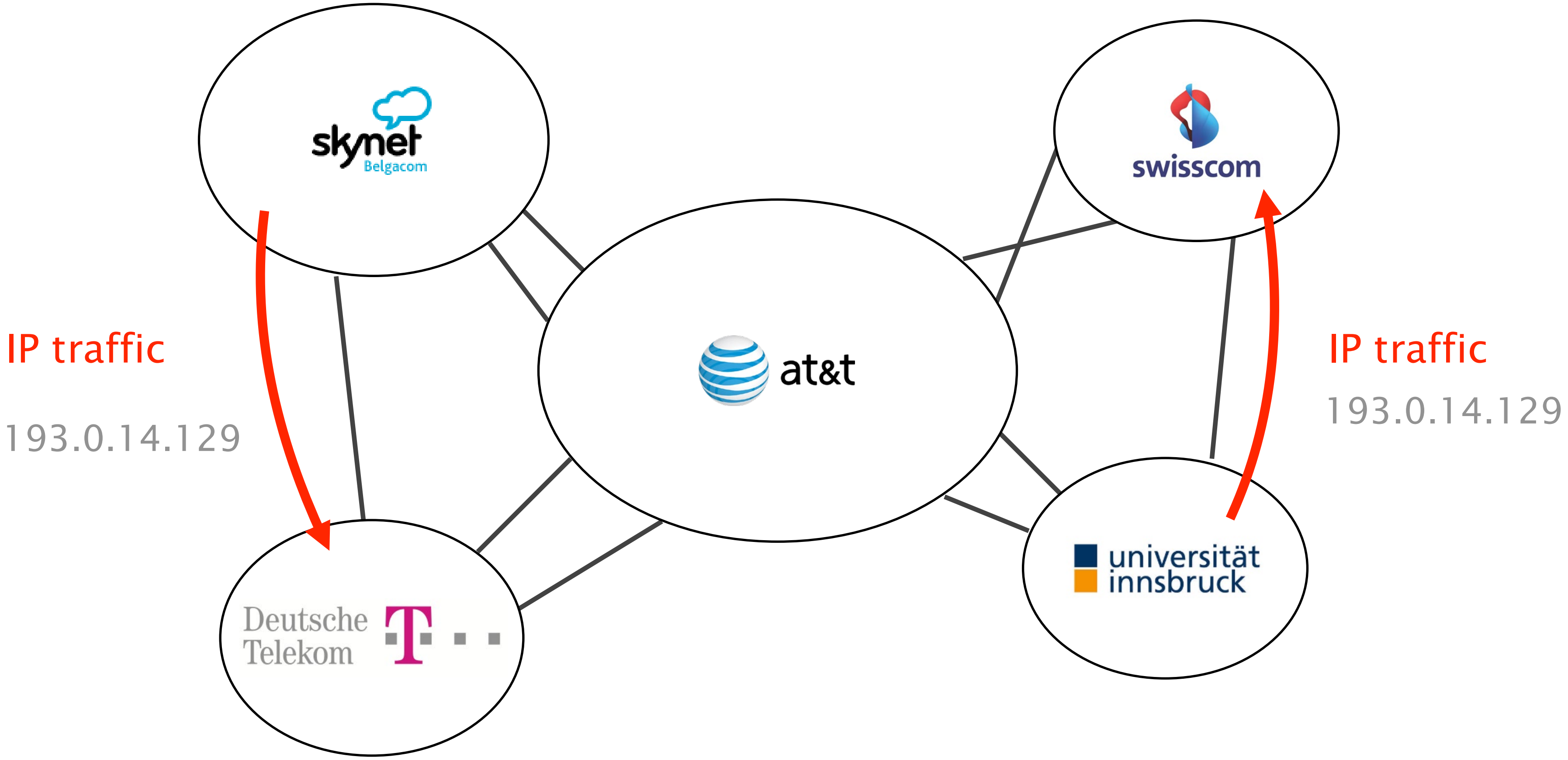
Intuition

Routing finds shortest-paths

If several locations announce the same prefix,
then routing will deliver the packets to
the “closest” location

This enables seamless replications of resources


AS20
AS10
AS30
AS40
AS50
138.232.0.0/16
Path: 30
138.232.0.0/16
Path: 50

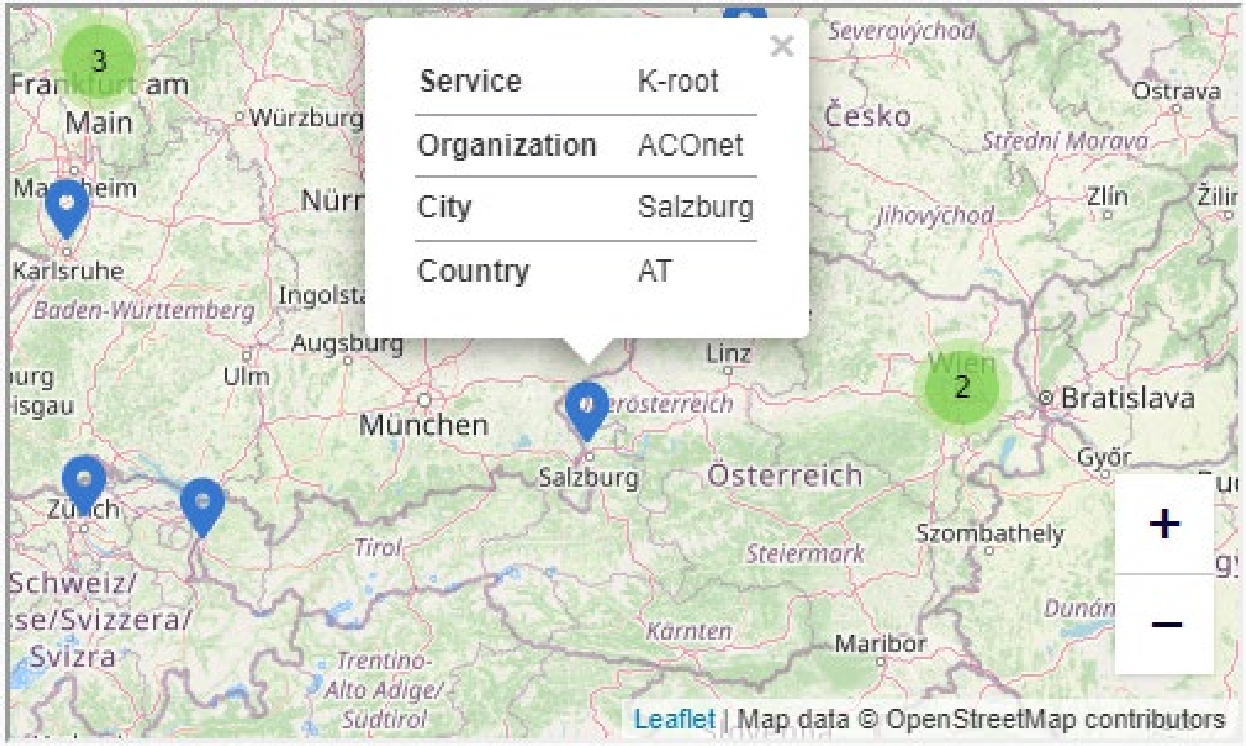

skynet
Belgacom
swisscom
at&t
IP traffic
193.0.14.129
IP traffic
193.0.14.129
Deutsche Telekom
universität innsbruck

Do you see any problems in
performing load-balancing this way?

Instances of the k-root server (*) are hosted in more than 40 locations worldwide

(*) see k.root-servers.org

Four of these locations are in Austria:
in Feldkirch, Salzburg and Vienna



All locations announce 193.0.14.0/23 in BGP,

with 193.0.14.129 being the IP of the server

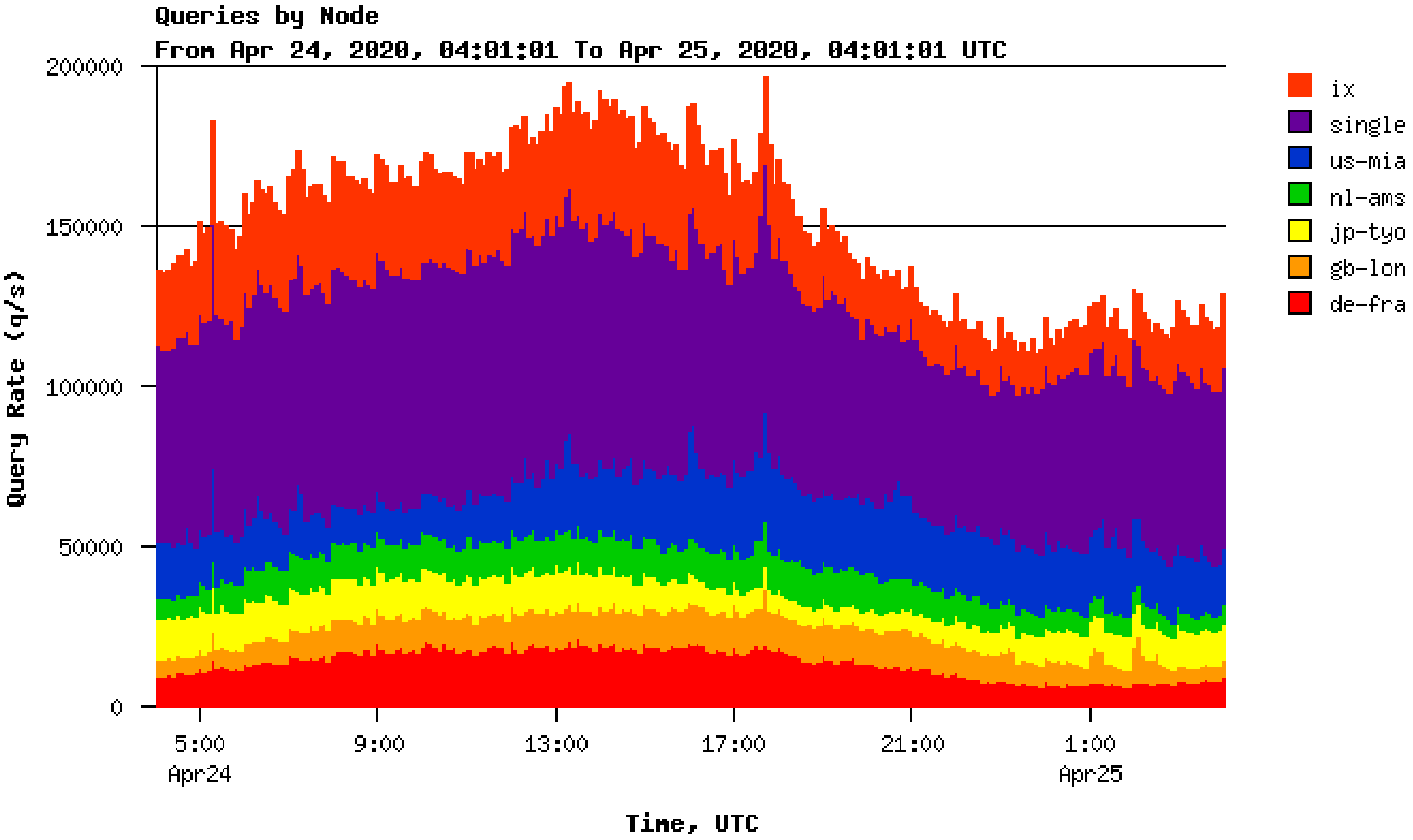
Four of these locations are in Austria:

in Feldkirch, Salzburg and Vienna

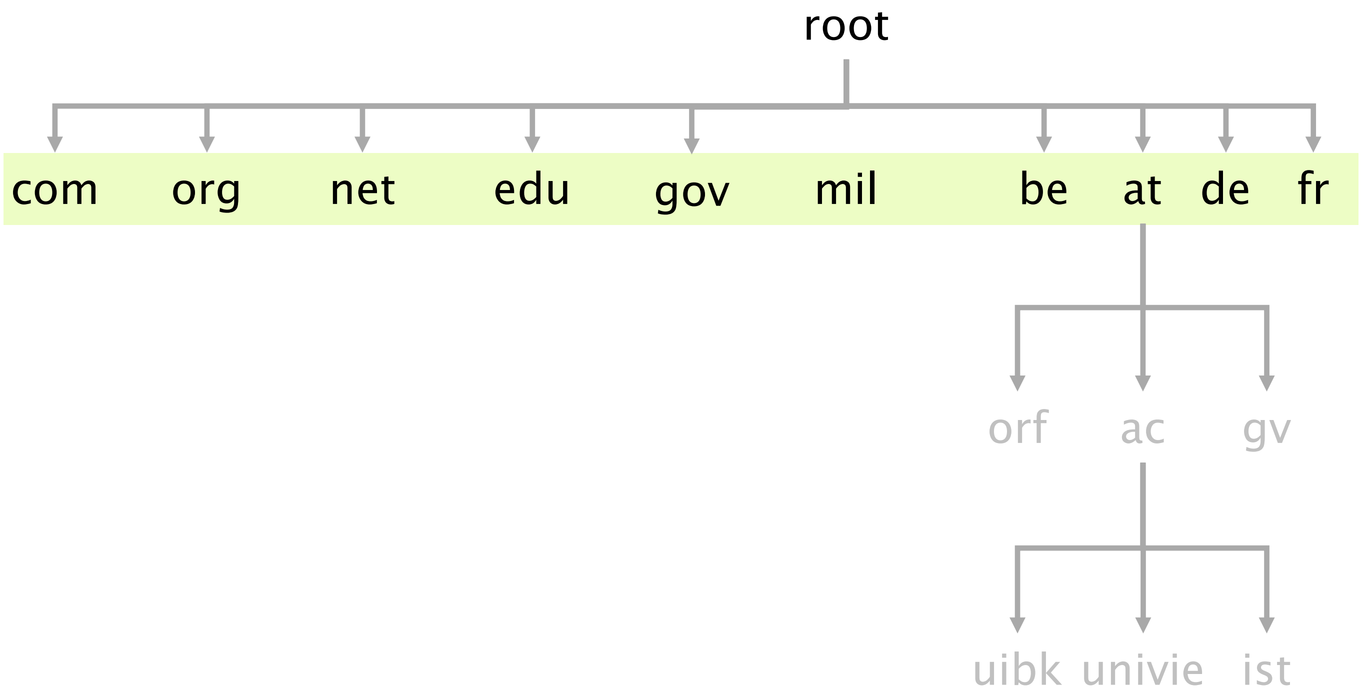
Do you mind guessing which one we use, here… in Innsbruck?

Each instance receives up to 70k queries per second

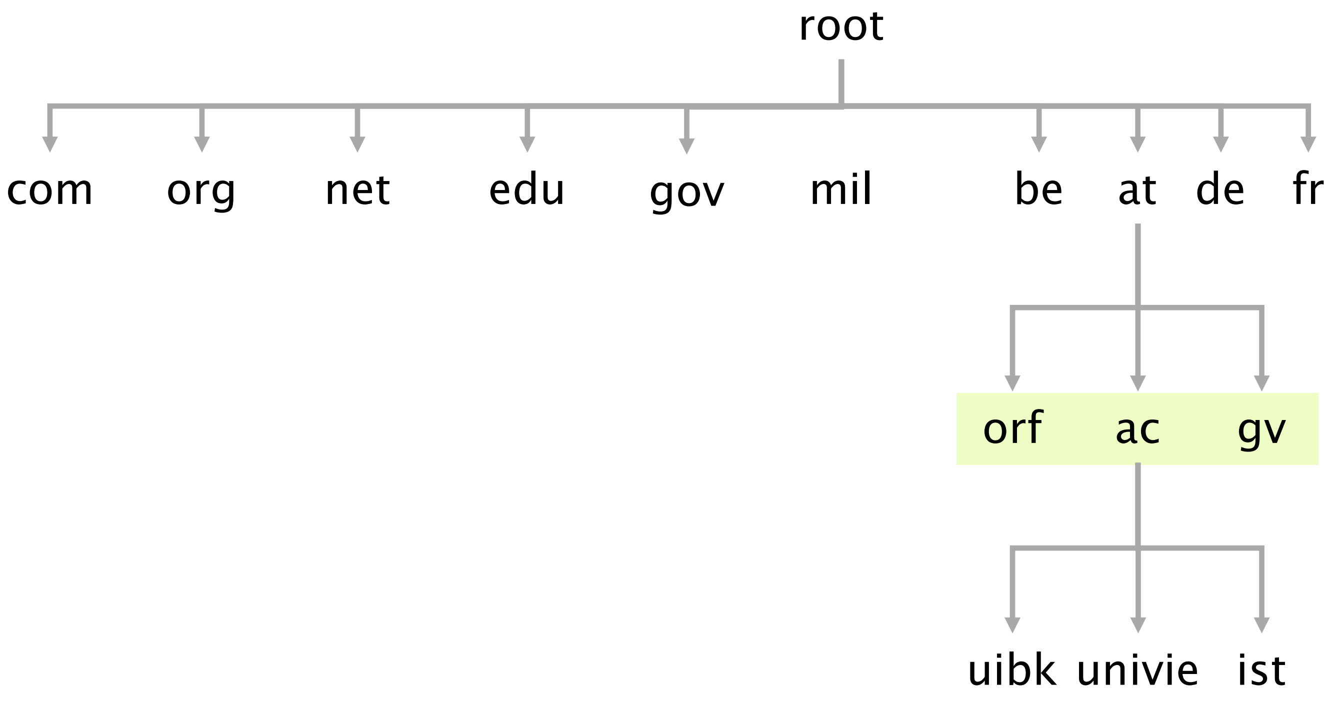
summing up to more than 4 billions queries per day



TLDs server are also managed professionally by private or non-profit organization



The bottom (and bulk) of the hierarchy is managed by Internet Service Provider or locally

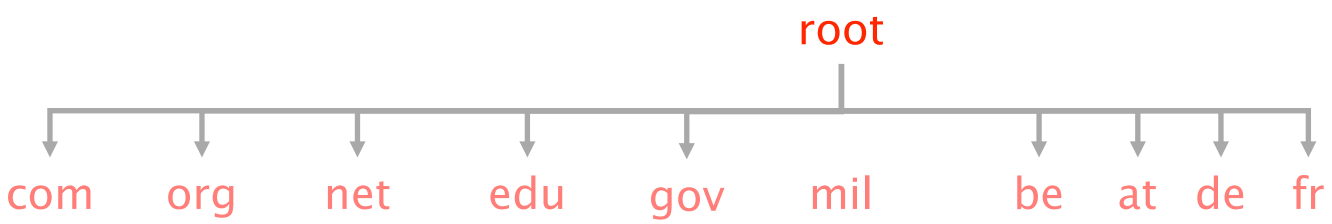


Every server knows the address of the root servers (*)

required for bootstrapping the systems

(*) see https://www.internic.net/domain/named.root

Each root server knows the address of all TLD servers

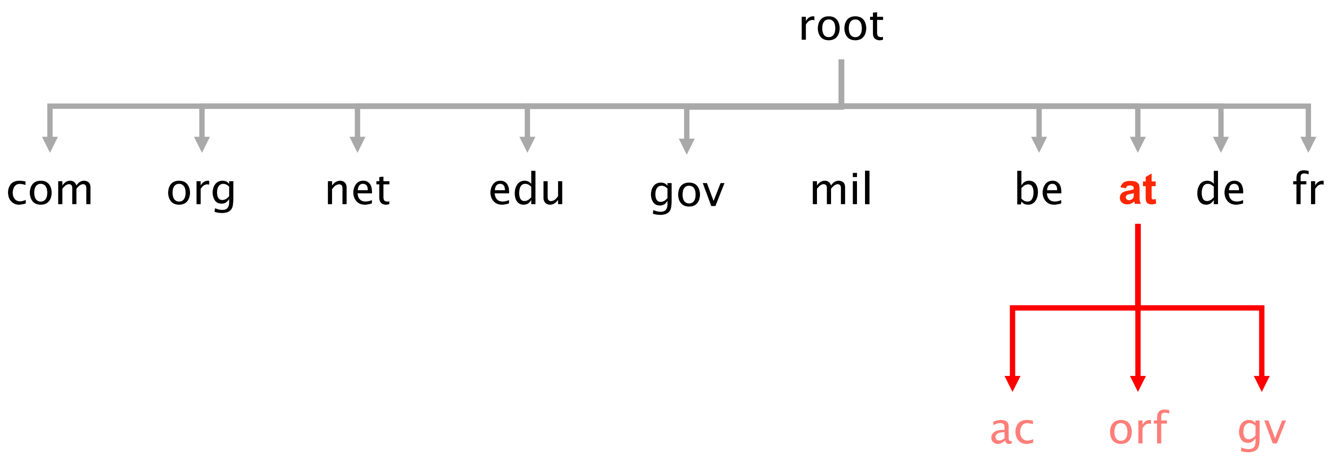



```
$ dig @a.root-servers.net at.

at.            172800  IN    NS    a.nic.at.
at.            172800  IN    NS    b.nic.at.
at.            172800  IN    NS    c.nic.at.
at.            172800  IN    NS    d.nic.at.
at.            172800  IN    NS    e.nic.at.
at.            172800  IN    NS    f.nic.at.
at.            172800  IN    NS    h.nic.at.
```

also see https://www.iana.org/domains/root/db/at.html

Any .at DNS server knowns the addresses of the DNS servers of all sub-domains




```
$ dig @d.ns.at orf.at


orf.at.        10800   IN    NS    ns4.apa.net.
orf.at.        10800   IN    NS    ns3.apa.at.
```

Once arrived at the leaf of the hierarchy (*.orf.at), each DNS server knows the IP address of all children

```
$ dig www.orf.at @ns3.apa.at

www.orf.at.     86400  IN  A   194.232.104.149
www.orf.at.     86400  IN  A   194.232.104.141
www.orf.at.     86400  IN  A   194.232.104.4
www.orf.at.     86400  IN  A   194.232.104.140
www.orf.at.     86400  IN  A   194.232.104.150
www.orf.at.     86400  IN  A   194.232.104.142
www.orf.at.     86400  IN  A   194.232.104.3
www.orf.at.     86400  IN  A   194.232.104.139
```

To ensure availability, each domain must have at least a primary and secondary DNS server

Ensure name service availability

as long as one of the servers is up

DNS queries can be load-balanced

across the replicas

On timeout, client use alternate servers

exponential backoff when trying the same server

Overall, the DNS system is highly scalable, available, and extensible

scalable	#names, #updates, #lookups, #users, but also in terms of administration
available	domains replicate independently of each other
extensible	any level (including the TLDs) can be modified independently

You've founded next-startup.ch and want to host it yourself, how do you insert it into the DNS?

You register next-startup.ch at a registrar *X*

e.g. ACONET or GoDaddy

Provide *X* with the name and IP of your DNS servers

e.g., [ns1.next-startup.ch,129.132.19.253]

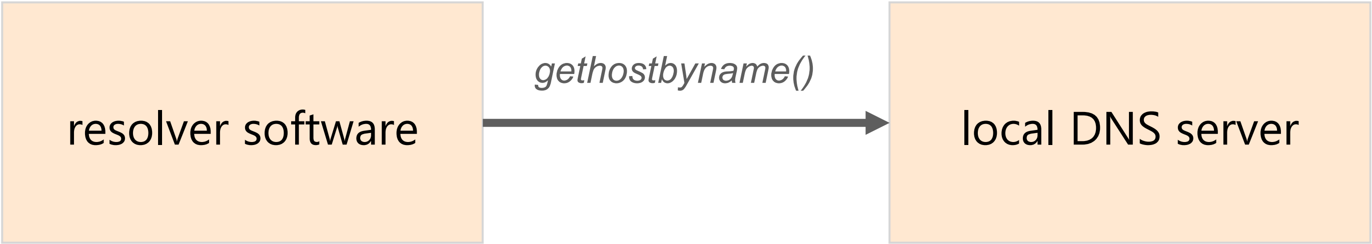
You set-up a DNS server @129.132.19.253

define A records for www, MX records for next-startup.ch…

A DNS server stores Resource Records composed of a (name, value, type, TTL)

Records	Name	Value
A	hostname	IP address
NS	domain	DNS server name
MX	domain	Mail server name
CNAME	alias	canonical name
PTR	IP address	corresponding hostname

Using DNS relies on two components



trigger resolution process

send request to local DNS server

usually, near the endhosts

configured statically (resolv.conf)

or dynamically (DHCP)

DNS query and reply uses UDP (port 53),
reliability is implemented by repeating requests (*)

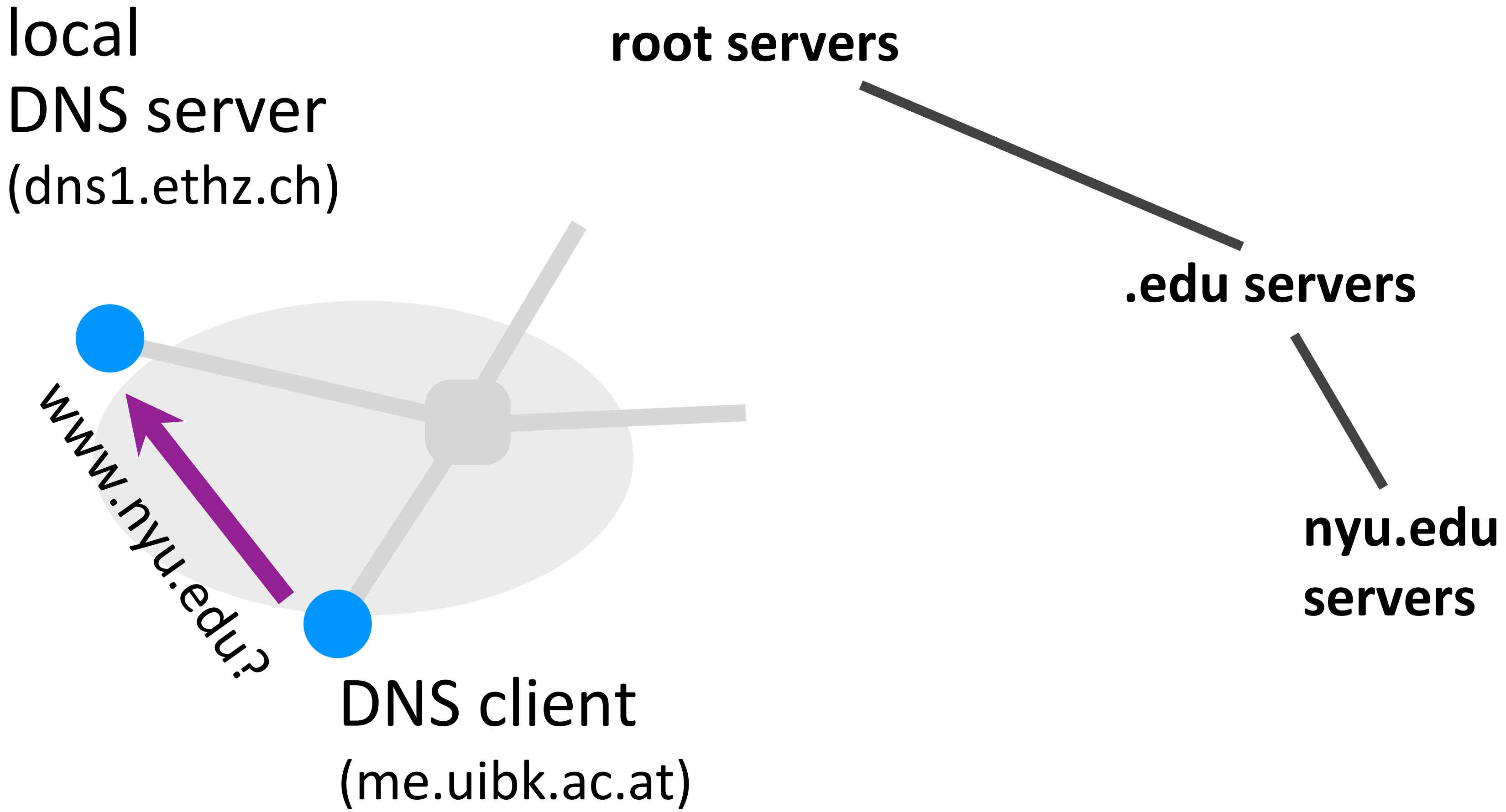
(*) see Book (Section 5)

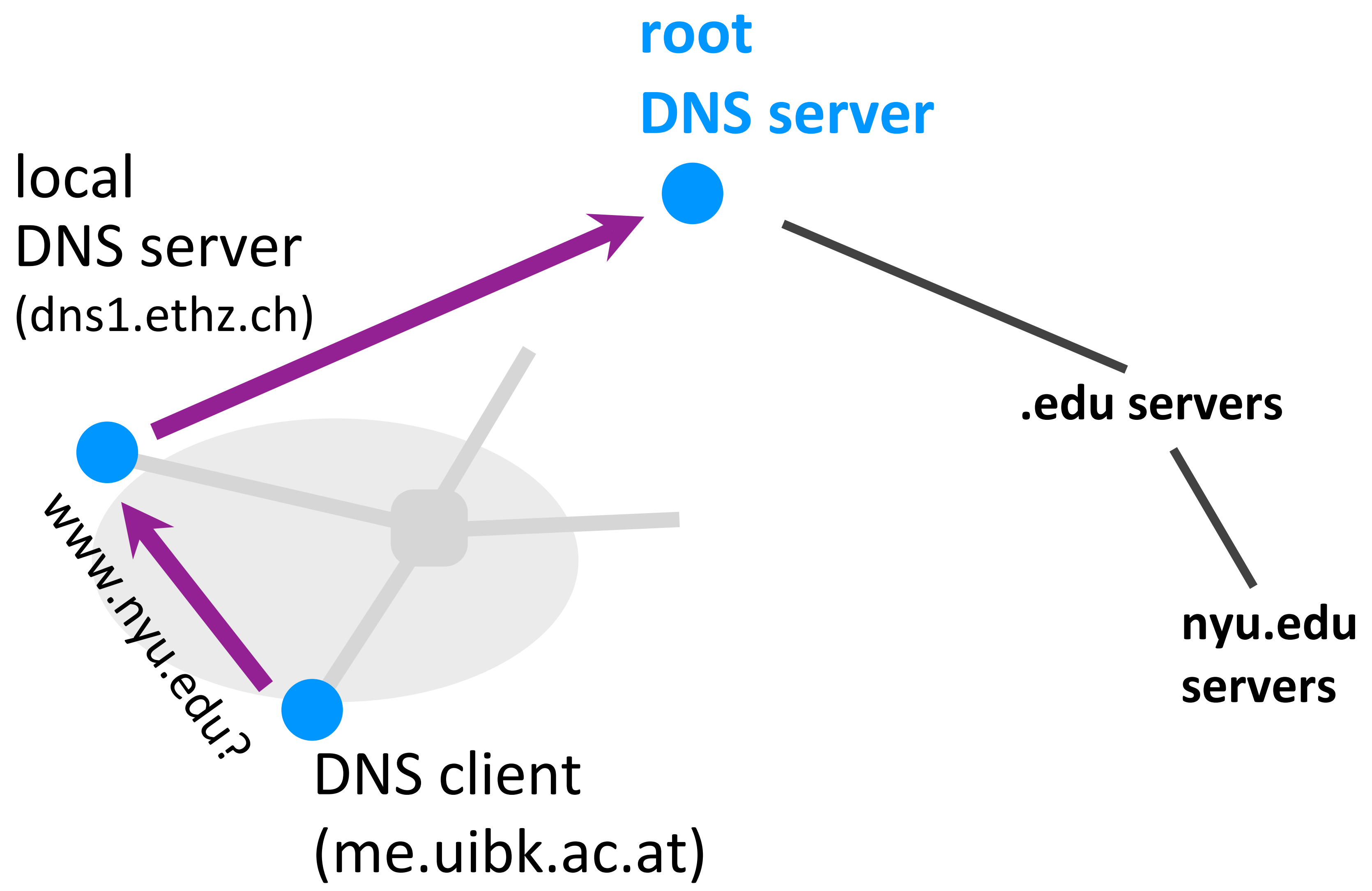
DNS resolution can either be

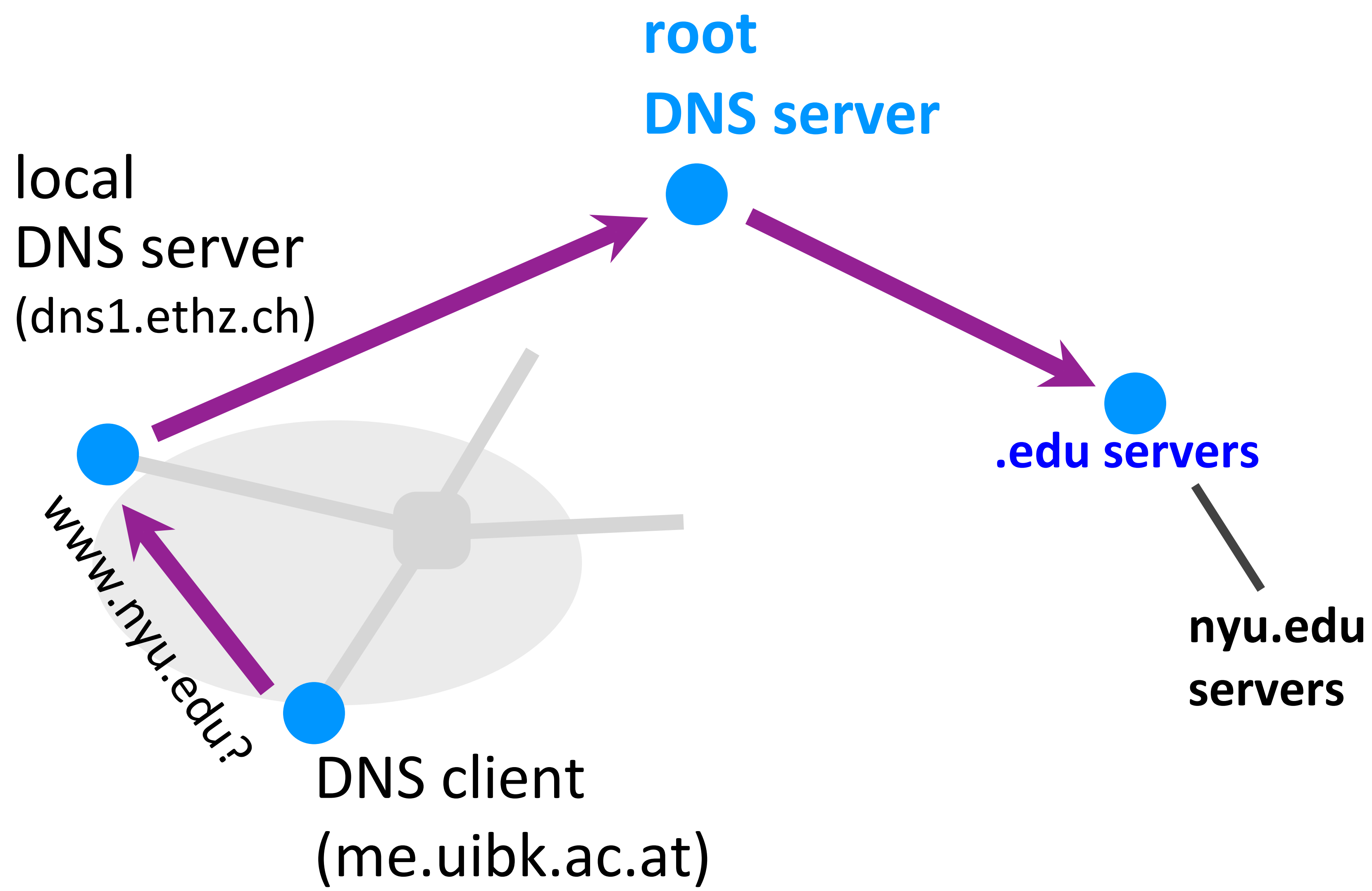
recursive or iterative

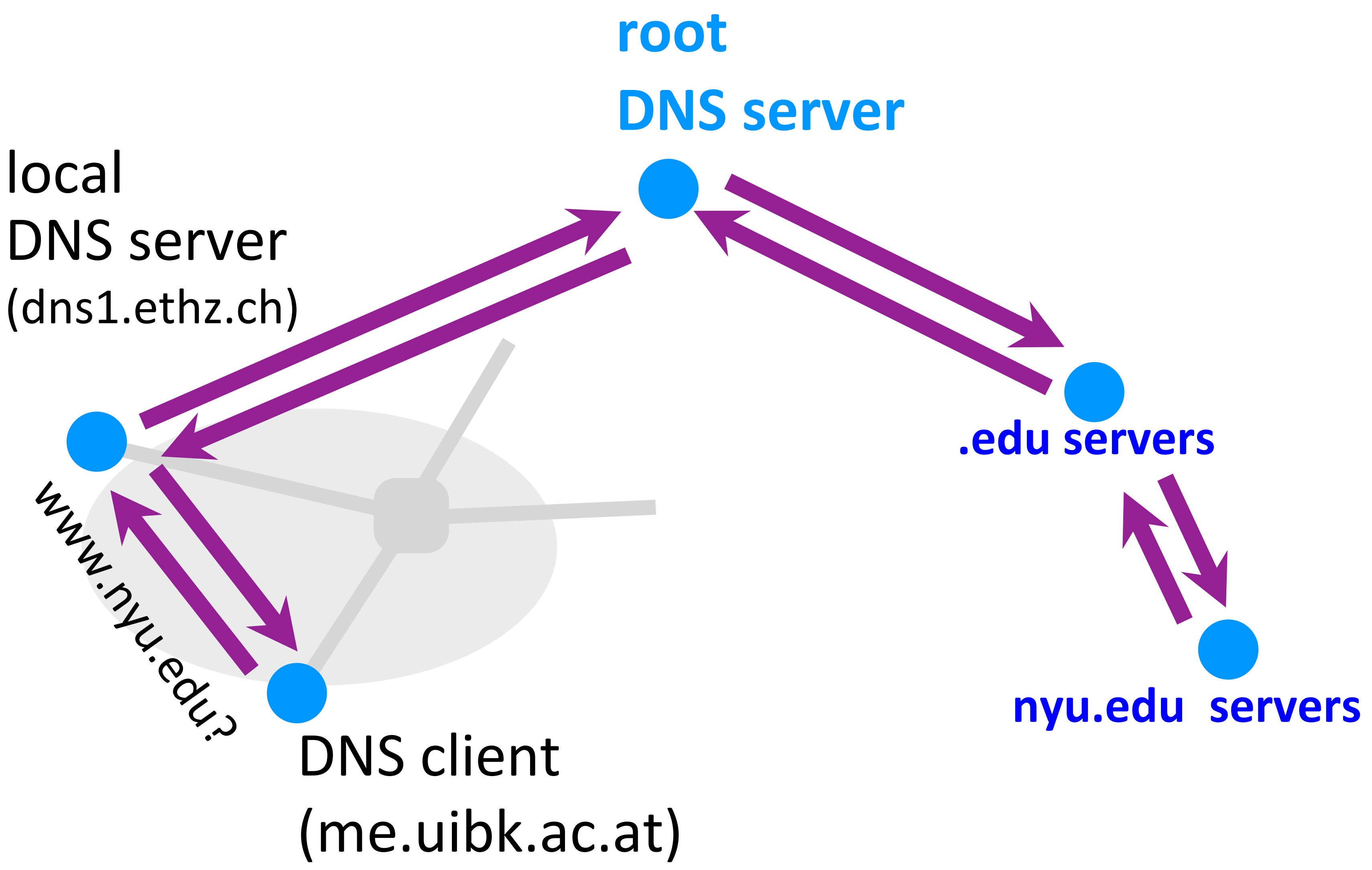
When performing a recursive query,

the client offload the task of resolving to the server

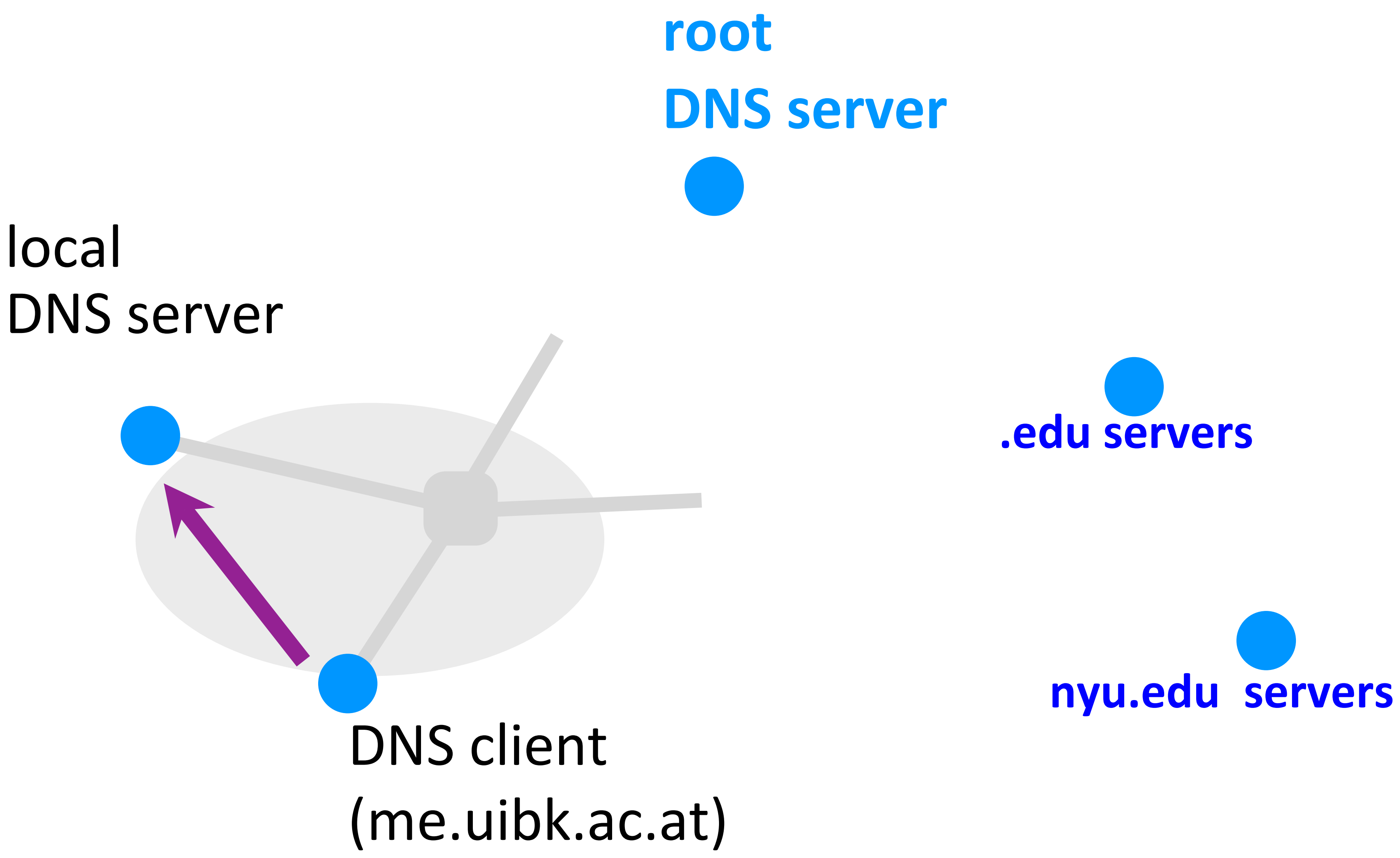

local
DNS server
(dns1.ethz.ch)
root servers
.edu servers
nyu.edu
servers
www.nyu.edu?
DNS client
(me.uibk.ac.at)

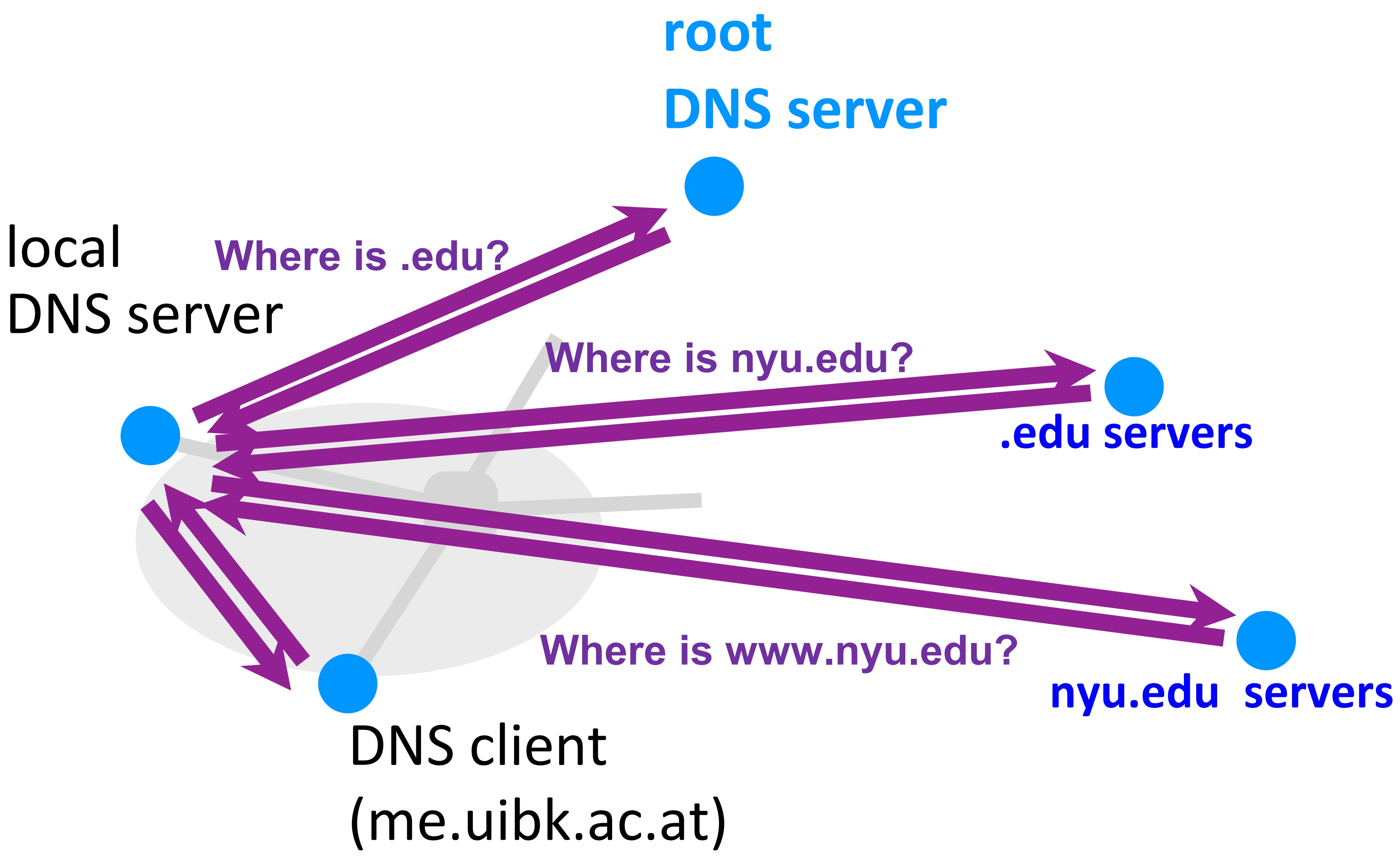

root
DNS server
local
DNS server
(dns1.ethz.ch)
.edu servers
www.nyu.edu?
nyu.edu
servers
DNS client
(me.uibk.ac.at)


root
DNS server
local
DNS server
(dns1.ethz.ch)
.edu servers
www.nyu.edu?
nyu.edu
servers
DNS client
(me.uibk.ac.at)


root
DNS server
local
DNS server
(dns1.ethz.ch)
www.nyu.edu?
DNS client
(me.uibk.ac.at)
.edu servers
nyu.edu servers

When performing a iterative query,
the server only returns the address of the "next server"


root
DNS server
local
DNS server
.edu servers
nyu.edu servers
DNS client
(me.uibk.ac.at)


root
DNS server
local
DNS server
Where is .edu?
Where is nyu.edu?
.edu servers
Where is www.nyu.edu?
nyu.edu servers
DNS client
(me.uibk.ac.at)

What about resolving speeds?

Waiting for servers all over the globe is not fast…

To reduce resolution times, DNS relies on caching

DNS servers cache responses to former queries

and your client *and* the applications (!)

Authoritative servers associate a lifetime to each record

Time-To-Live (TTL)

DNS records can only be cached for TTL seconds

after which they must be cleared

As top-level servers rarely change & popular website visited often, caching is very effective (*)

Top 10% of names account for 70% of lookups

9% of lookups are unique

Limit cache hit rate to 91%

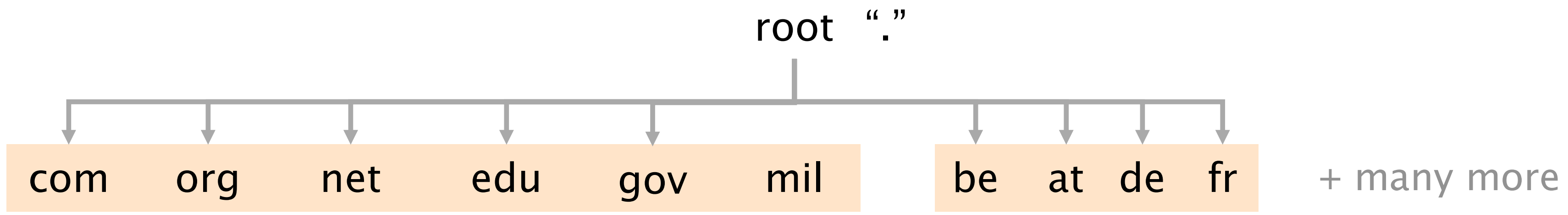
Practical cache hit rates ~75%

(*) see https://pdos.csail.mit.edu/papers/dns:ton.pdf

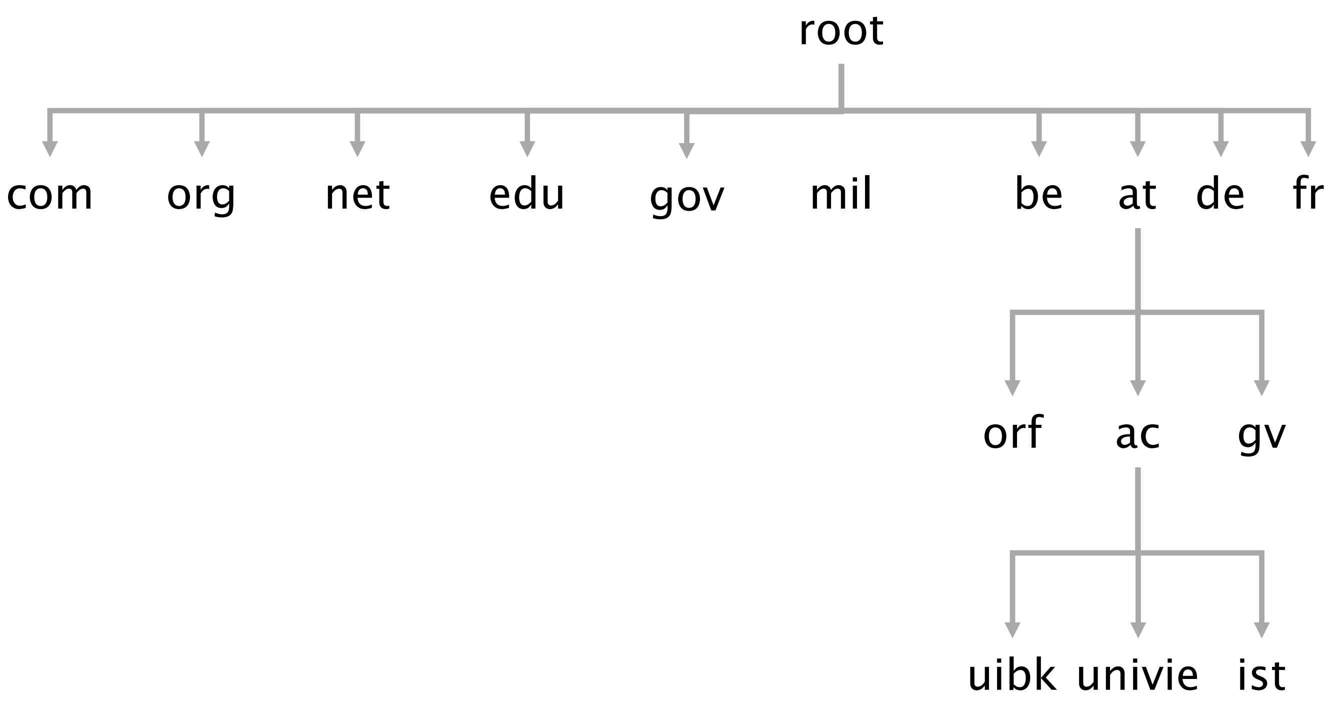
Communication Networks and Internet Technology

Short Recap on this weeks lecture

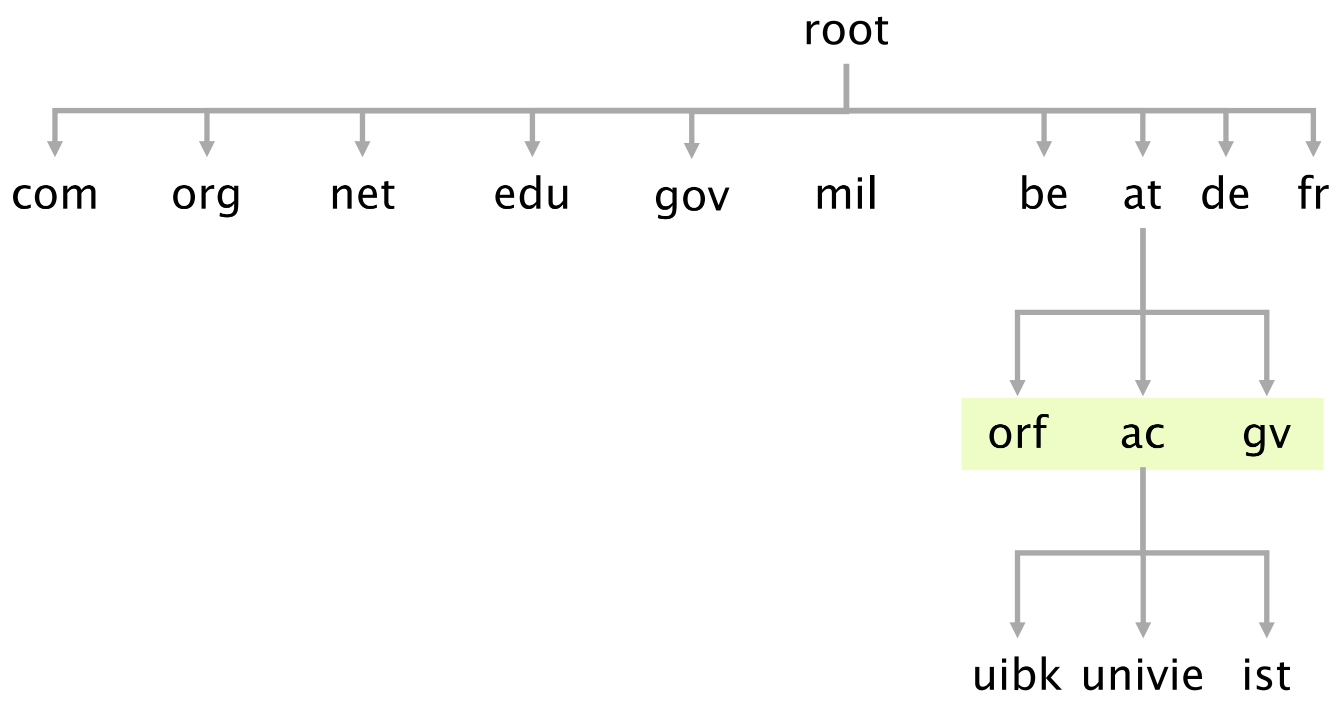
Top Level Domain (TLDs) sit at the top

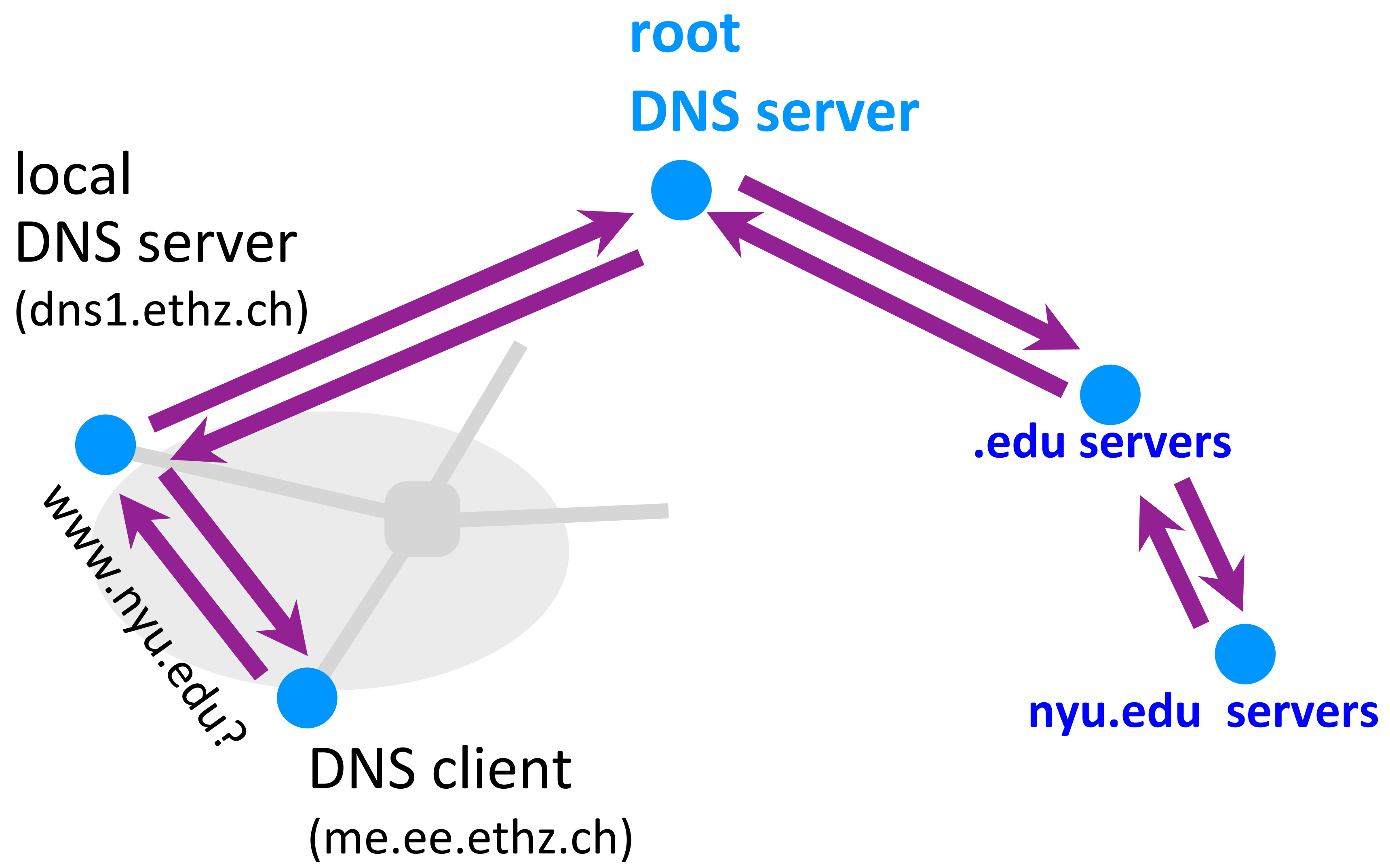


The DNS system is hierarchically administered

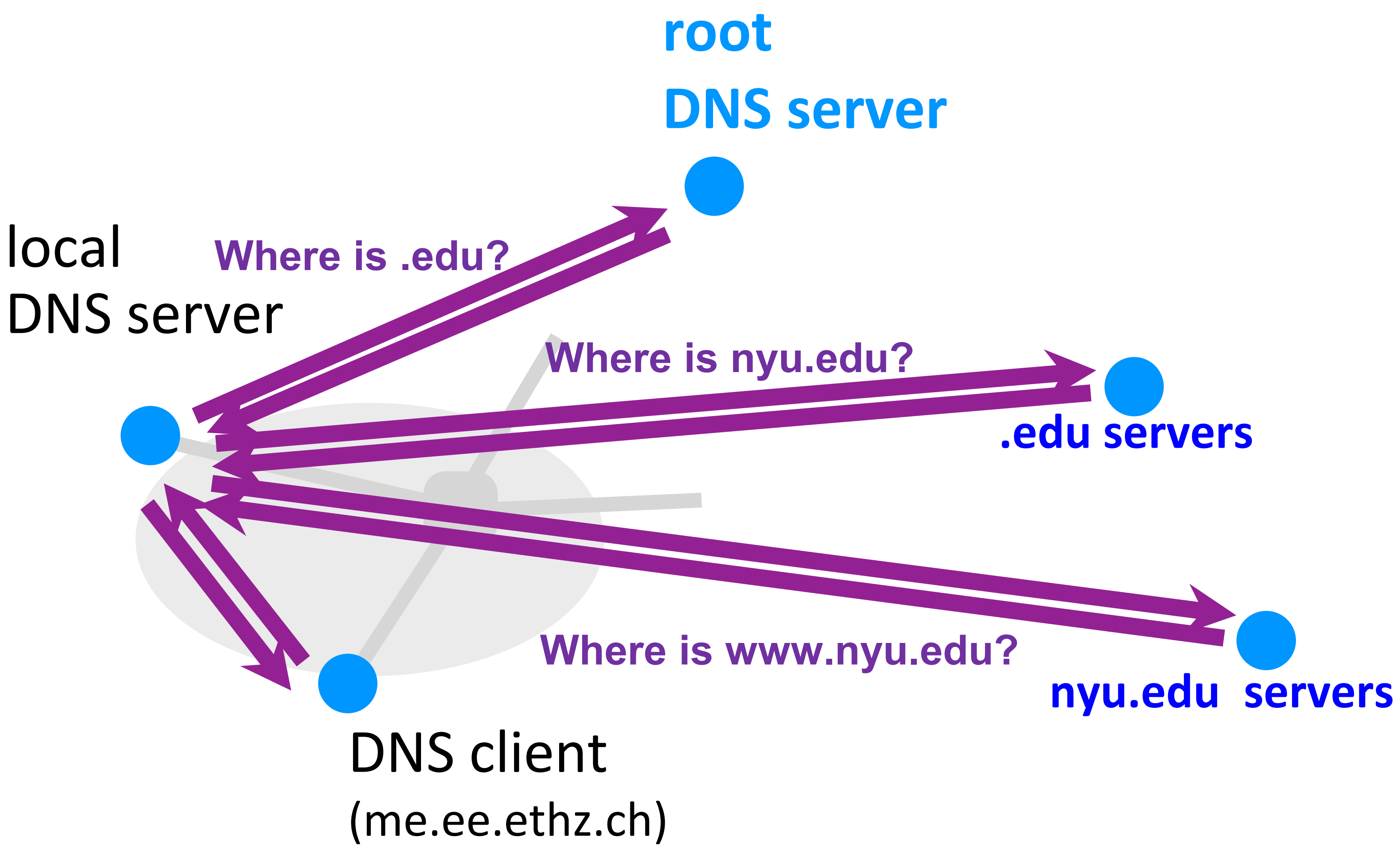


The bottom (and bulk) of the hierarchy is managed by Internet Service Provider or locally



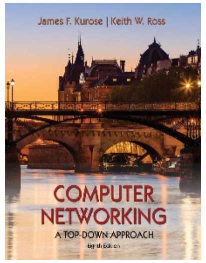


root
DNS server
local
DNS server
(dns1.ethz.ch)
www.nyu.edu?
DNS client
(me.ee.ethz.ch)
.edu servers
nyu.edu servers


root
DNS server
local
DNS server
Where is .edu?
Where is nyu.edu?
.edu servers
Where is www.nyu.edu?
nyu.edu servers
DNS client
(me.ee.ethz.ch)

Reading: Book Kurose & Ross

Class textbook:
Computer Networking: A Top-Down Approach (8th ed.)
J.F. Kurose, K.W. Ross
Pearson, 2020
http://gaia.cs.umass.edu/kurose_ross



- Week 11
  - 2.5 (DNS. The Internet's Directory Service)
  - 2.2 (The Web and HTTP)

Check Your Knowledge

PROBLEM SOLVING HOME

TRY A RANDOM PROBLEM

INTERACTIVE END-OF-CHAPTER EXERCISES

Supplement to Computer Networking: A Top Down Approach 8th Edition

CHAPTER 2: APPLICATION LAYER

- DNS - Basics
- DNS - Iterative vs Recursive Query
- DNS and HTTP delays (similar to Chapter 2, P7,P8)
- HTTP GET (similar to Chapter 2, P4)
- HTTP RESPONSE (similar to Chapter 2, P5)
- Browser Caching
- Electronic Mail and SMTP
- A comparison of client-server and P2P file distribution delays (similar to Chapter 2, P22)

http://gaia.cs.umass.edu/kurose_ross/interactive/

Final Exam

- 02.02.2022 – 10:15h
- Online exam
- Duration **120 min**
- Open book
- Individual problem solving
- No (electronic) communication allowed
- Random spot checks

- Register online via lfuonline