How do you resolve a name into an IP?

In olden times (1980s)

- all host to address mappings were in a file called hosts.txt
- in /etc/hosts
- Had to download regularly
- *still useful for certain situations. /etc/hosts takes precedence
    - https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts

Problem:

- Scalability in terms of
    - Management
    - Availability
    - Consistency

How do you resolve a name into an IP?

In olden times (1980s)

- all host to address mappings were in a file called hosts.txt
- in /etc/hosts
- *still use precedence
  - h ts/master/hosts

What do you do when you need scalability?

Problem:

- Scalabilit
  - Manag
  - Availability
  - Consistency

How do you resolve a name into an IP?

In olden times (1980s)

- all host to address mappings were in a file called hosts.txt
- in /etc/hosts
- *still use precedence
  - h ts/master/hosts

What do you do when you need scalability?
a hierarchical structure

Problem:

- Scalabilit
  - Manag
  - Availability
  - Consistency

To scale, DNS adopt three intertwined hierarchies

naming structure	hierarchy of addresses https://ee.hawaii.edu/home/
Management	hierarchy of authority over names
Infrastructure	hierarchy of DNS servers

To scale, DNS adopt three intertwined hierarchies

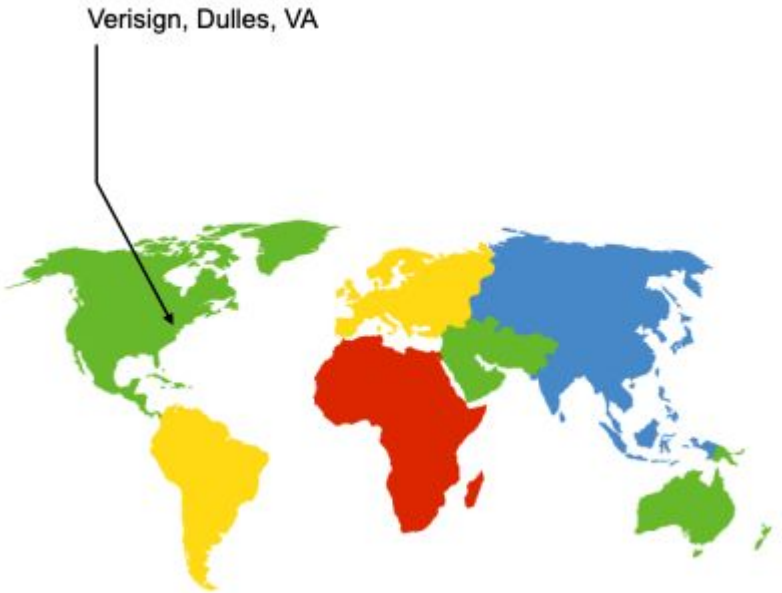
naming structure	hierarchy of addresses https://ee.hawaii.edu/home/
Management	hierarchy of authority over names
Infrastructure	hierarchy of DNS servers

DNS root

Located in Virginia, USA

Every server knows the address of root servers - needed for bootstrap https://www.internic.net/domain/named.root

How do we make the root scale?



DNS root

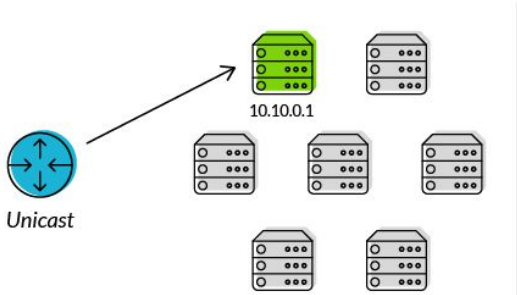
13 root servers (see http://www.root-servers.org/)

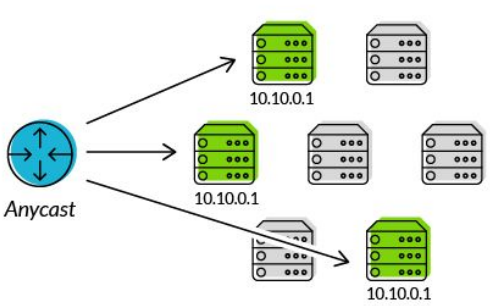
Labeled A through M

- Does this scale?

To scale root servers, operators rely on BGP anycast

- 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

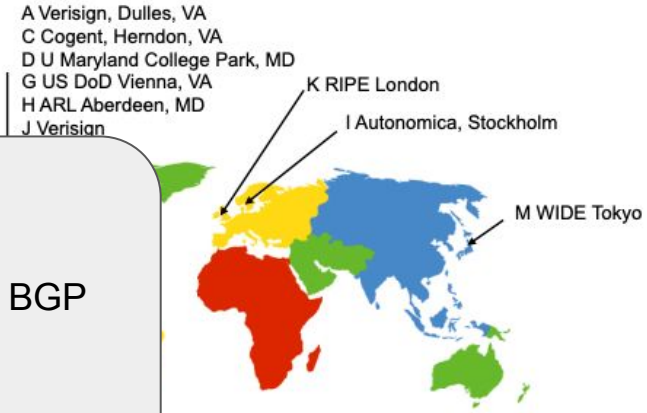


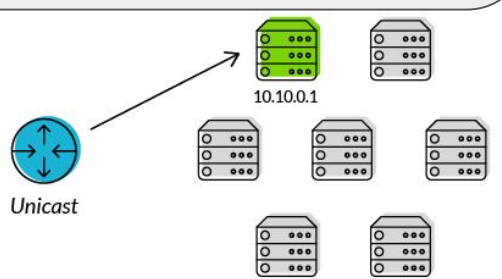


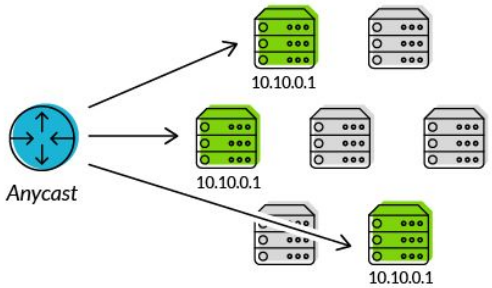
To scale root servers, operators rely on BGP anycast

- Routing finds shortest-paths
- If several location
  the same prefix, t
  will deliver the pa
  “closest” location
- This enables seamless
  replications of resources

Do you see any problems with using BGP anycast for load balancing?

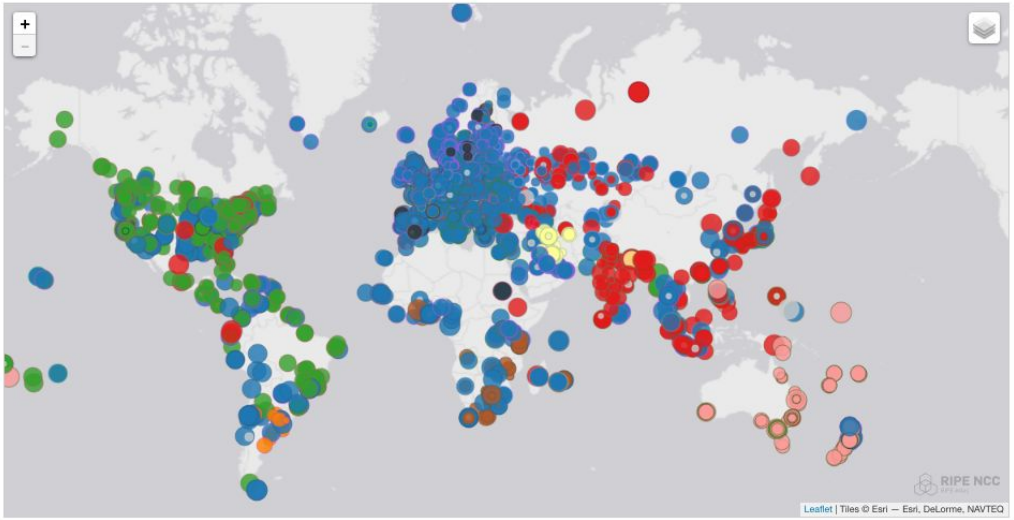






To scale root servers, operators rely on BGP anycast

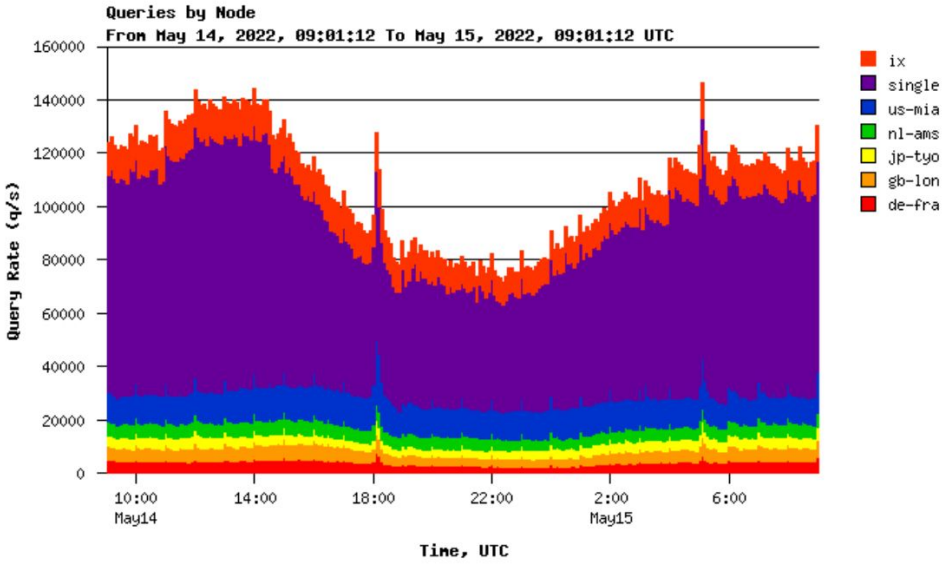
- K root (RIPE) anycast
  - Color == server used
- BGP is mediocre at this!



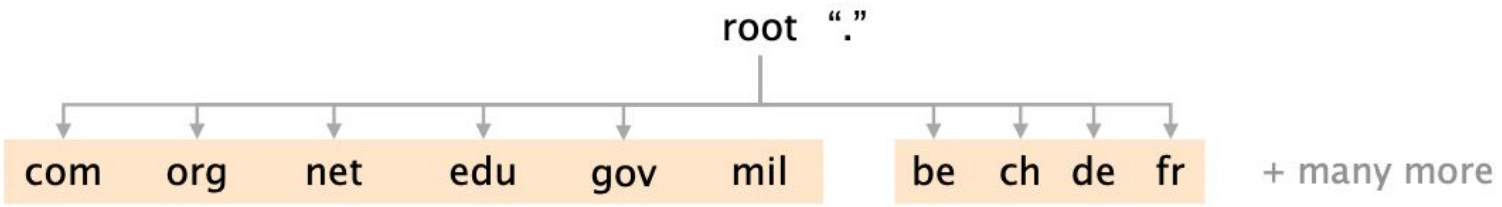
DNS scale

Each instance receives up to 80k queries per second

summing up to a few billions of queries per day



Top Level Domain (TLDs) sit below the root

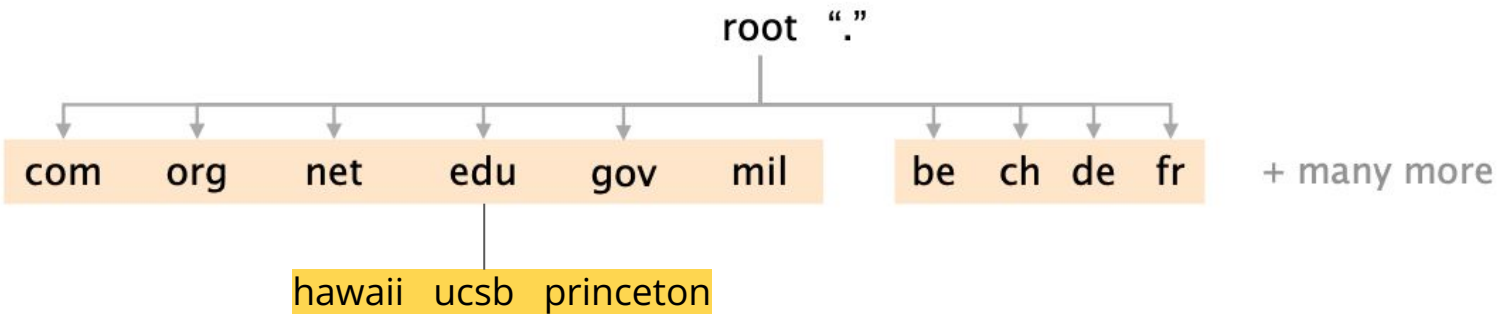


Each root knows the address of all TLD servers

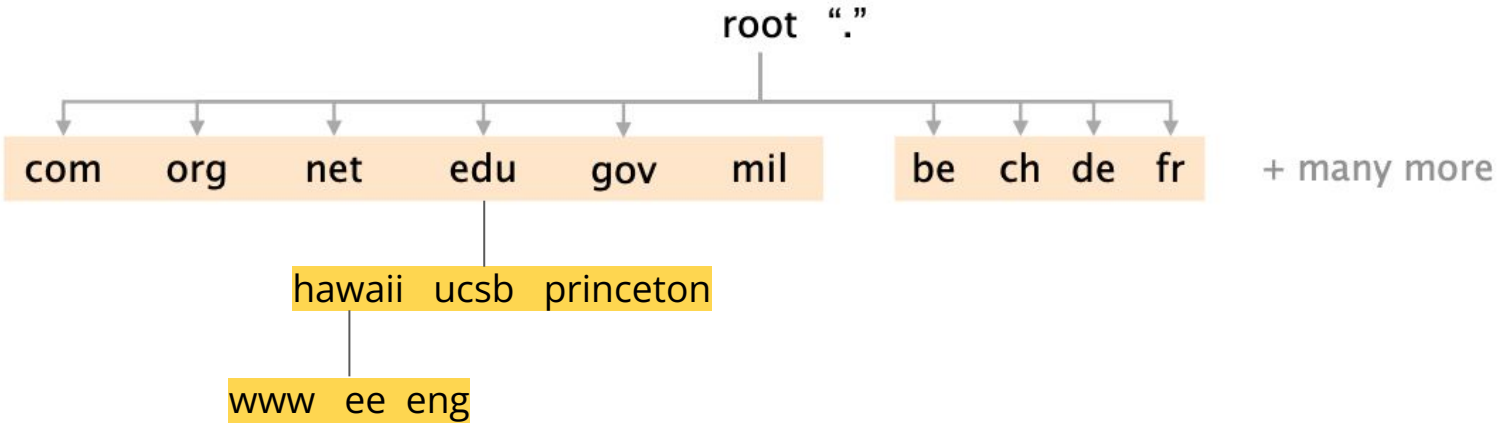
TLD and Authoritative DNS servers

- Top-level domain (TLD) servers
  - Generic domains (e.g., com, org, edu)
  - Country domains (e.g., uk, fr, cn, jp)
  - Special domains (e.g., arpa)
  - Typically managed professionally
    - Network Solutions maintains servers for “com”
    - Educause maintains servers for “edu”
- Authoritative DNS servers
  - Provide public records for hosts at an organization
  - For the organization’s servers (e.g., Web and mail)
  - Can be maintained locally or by a service provider

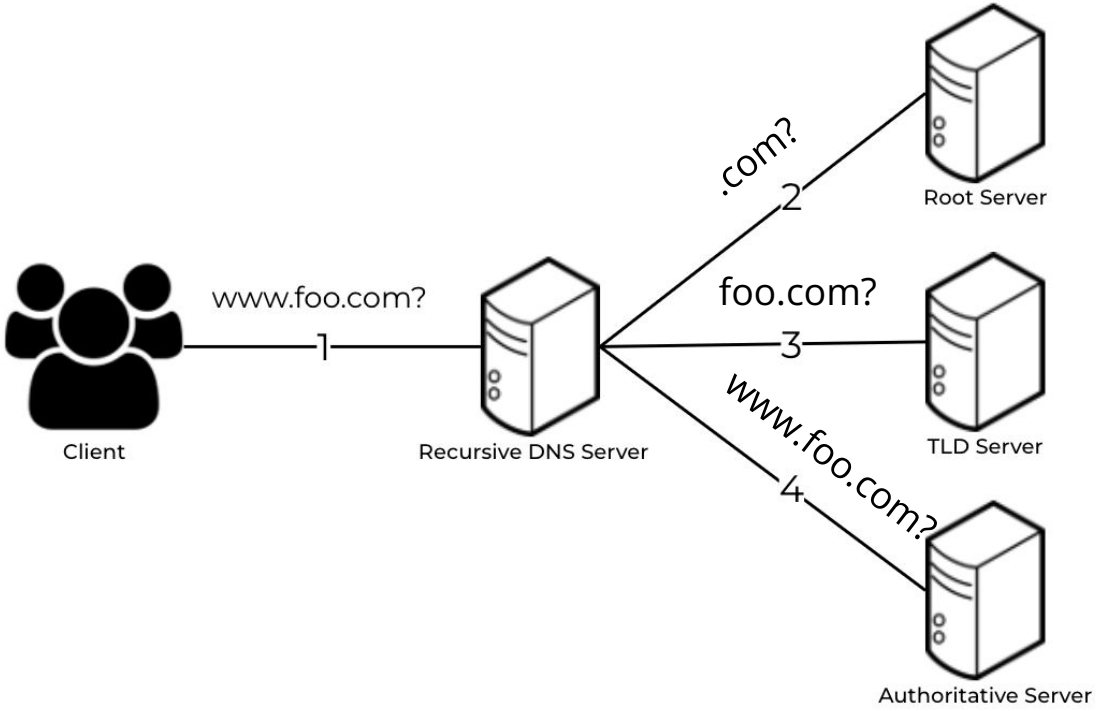
Domains are subtrees



A name, e.g. ee.hawaii.edu, represents a leaf-to-root path in the hierarchy



DNS Hierarchy



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