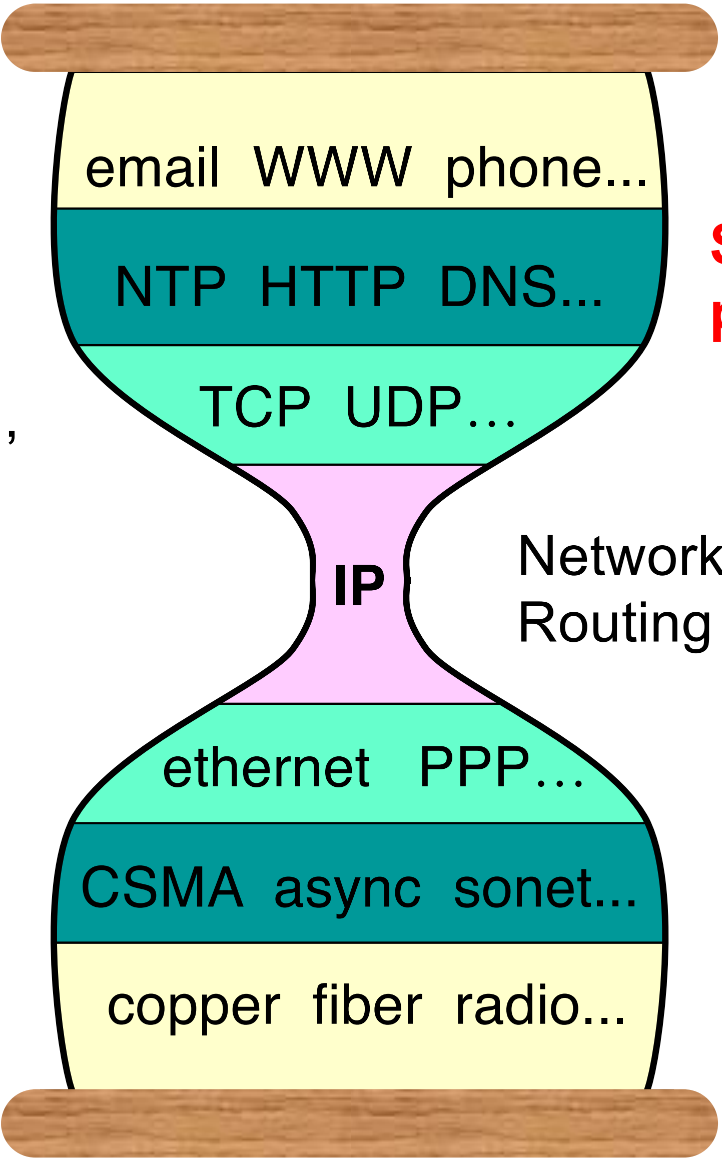
CSC 525:
Computer Networks

The Story So Far



email WWW phone...

NTP HTTP DNS...

TCP UDP…

IP

ethernet PPP…

CSMA async sonet...

copper fiber radio...

Some App layer protocols

Transport layer: End to End communication, Reliability, Congestion control, etc.

Network layer: Addressing, Routing and forwarding.

Data Layer: richly connected network with many types of links

Naming

- Name, a digital identifier
  - The very basic component in any communication.
  - www.cs.arizona.edu, 192.12.69.186, joe@cs.arizona.edu, 00:0d:93:60:5a:bb, etc.
- Name space
  - Flat or hierarchical
  - Variable or fixed length

Mapping

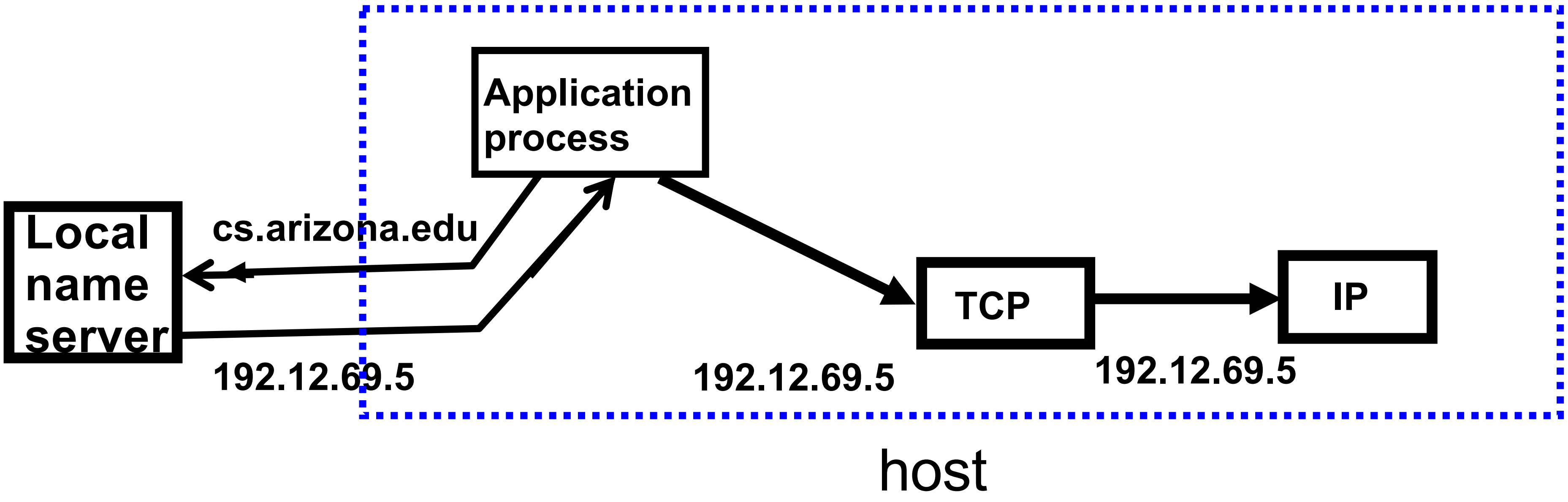
- Different protocols or applications have their own identifiers
  - To best serve the purpose
    - Host names for applications, IP addresses for routers
  - decouple from other protocols or applications.
    - Be able to change IP without changing host name
- Mapping is needed between different name spaces
  - DNS: host names → IP addresses
  - ARP: IP addresses → Link layer addresses

Name Resolution

- The mechanism that resolves a name to another.
- Different designs in different contexts
  - Broadcast: ARP
  - Search a centralized database: LDAP
  - Query a distributed database: DNS
- Resolution scheme often depends on the name space.

Domain Name System (DNS)

- An important piece of Internet infrastructure
- DNS name to address translation
  - DNS name: variable length, mnemonic, tied to organizations
  - IP address: fixed length, tied to network topology
  - API: gethostbyname( ) etc.



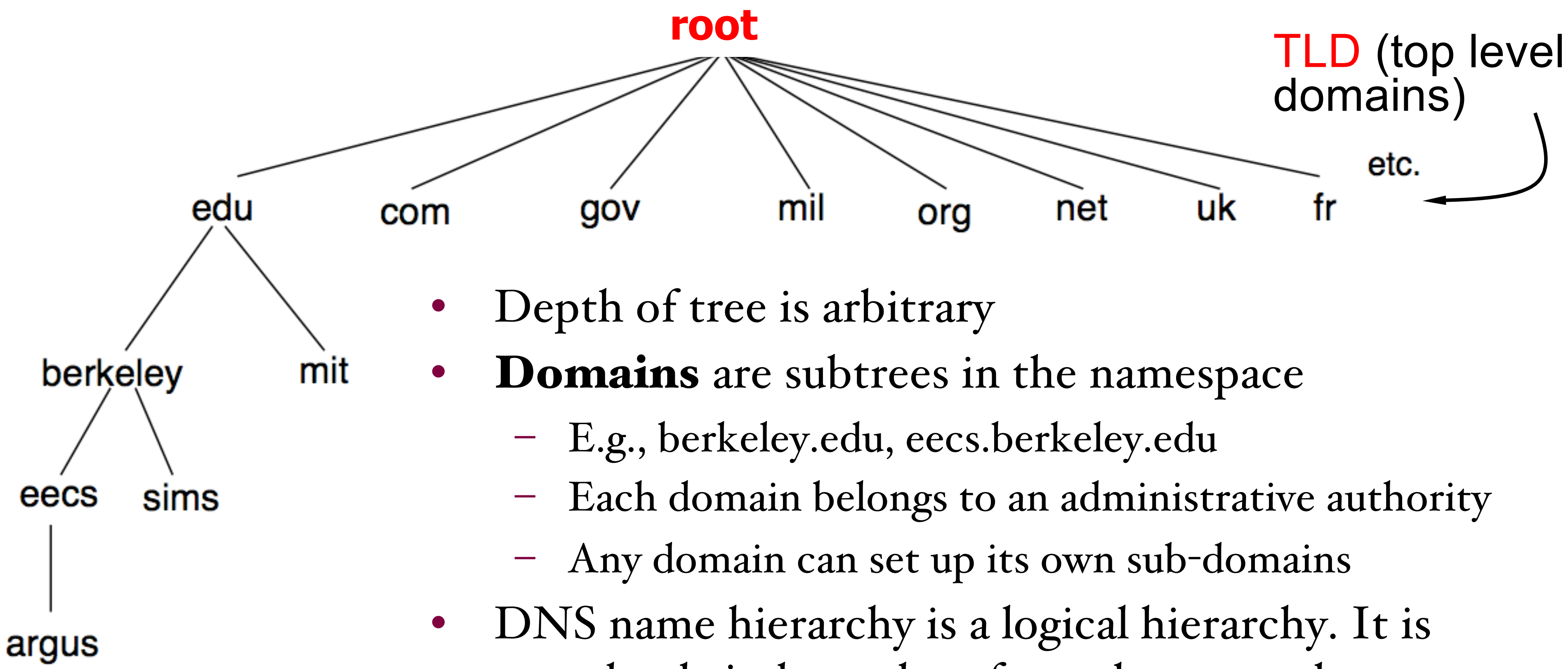
Before DNS

- A centrally managed file HOSTS.TXT (/etc/hosts)
  - Changes were submitted to SRI
  - New files were ftp'd periodically from SRI
  - An administrator could pick names at his discretion.
- It couldn't scale as the Internet grew
  - SRI couldn't handle the load
  - Names were not unique
  - Many hosts have inaccurate copies of hosts.txt
- Internet growth was threatened
  - DNS was born

Basic DNS Components

- A hierarchical name space
  - As opposed to original flat name space
  - Follow the hierarchy of organizations, not the network topology.
- A distributed database, realized by a hierarchy of servers
  - Provided by individual domain owners
  - As opposed to original centralized storage
- Local resolvers
  - Provided by ISPs to serve local users.
- A client-server access protocol
  - on UDP port 53, can use TCP if desired
  - Usually just one round of query-reply between a client and a resolver.

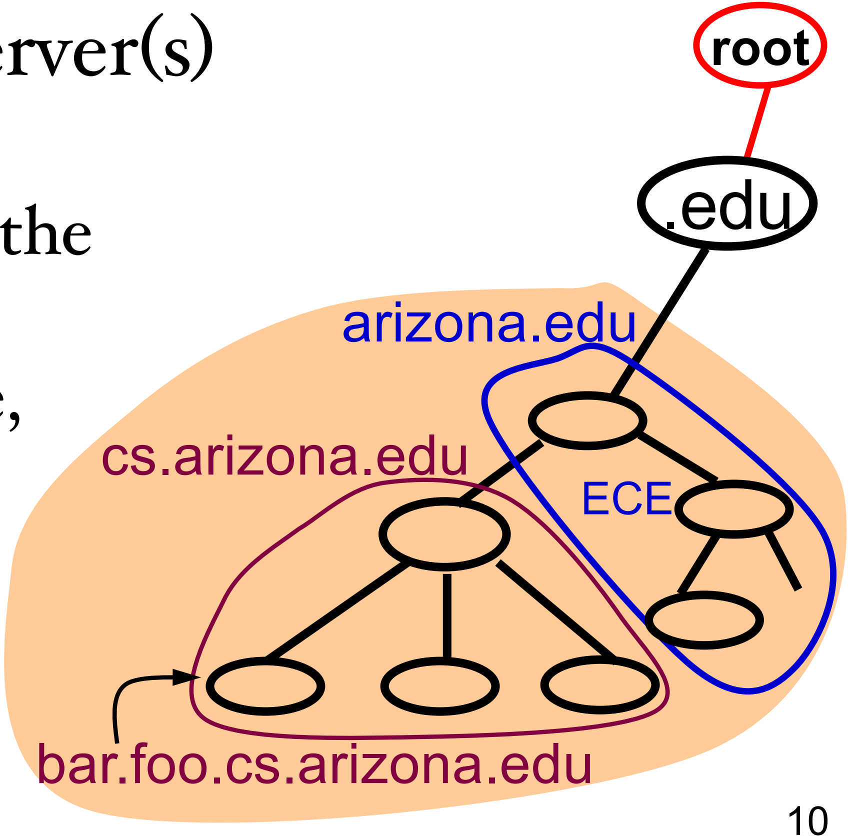
Name Hierarchy



- Depth of tree is arbitrary
- **Domains** are subtrees in the namespace
  - E.g., berkeley.edu, eecs.berkeley.edu
  - Each domain belongs to an administrative authority
  - Any domain can set up its own sub-domains
- DNS name hierarchy is a logical hierarchy. It is completely independent from the network topological structure.

DNS as a Distributed Database

- In implementation, the entire DNS database is divided to a hierarchy of zones
  - zone: a *continuous sub-space* in the DNS name tree
  - a zone may contain domains at different levels
- Each zone is controlled by an administrative authority with its own name server(s)
- Zone vs. Domain
  - A domain is a logical sub-tree in the name space.
  - A zone is a continuous sub-space, which is stored in the same server and managed by the same admin.



Server Hierarchy

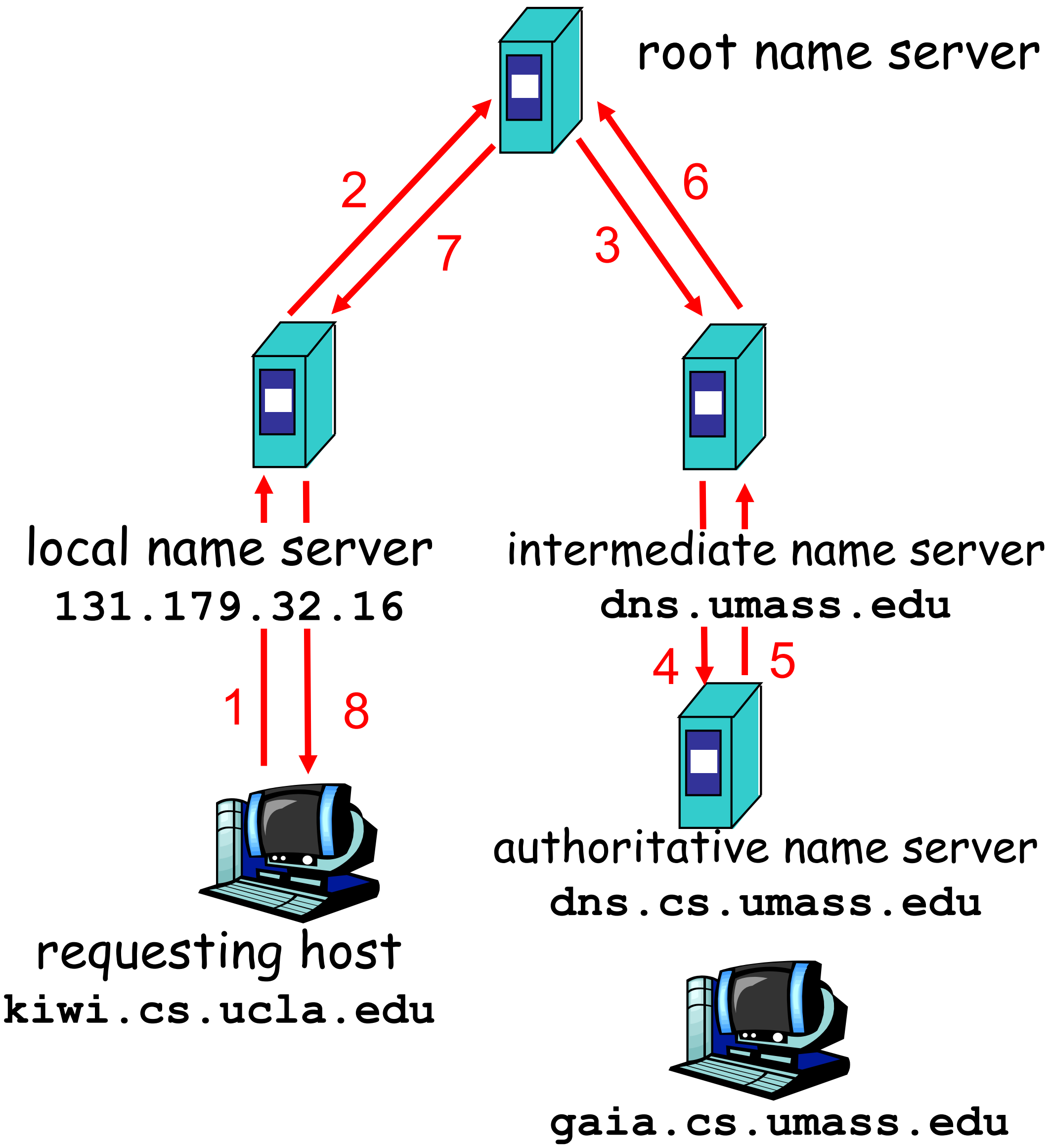
- Each server maintains a subset of the name space
  - A server can support multiple zones
  - A server of a zone doesn't have to be part of the corresponding domain, nor in the corresponding network.

- Each server needs to know some other servers in order to resolve names outside of its own zone
  - All hosts know its local name server (via DHCP or manually configured) and send all queries to it.
  - All servers know the root servers (manually configured)
  - Each server knows the name servers in its sub-zones.
  - Search the hierarchy top-down.

Recursive Query

host **kiwi.cs.ucla.edu** wants IP address of **gaia.cs.umass.edu**

1. Contacts its local DNS server, **131.179.32.16** (dns.cs.ucla.edu)
2. **dns.cs.ucla.edu** contacts root name server, *if necessary*
3. root name server contacts umass name server, **dns.umass.edu**, *if necessary*
4. **dns.umass.edu** contacts the authoritative name server, **dns.cs.umass.edu**, if necessary



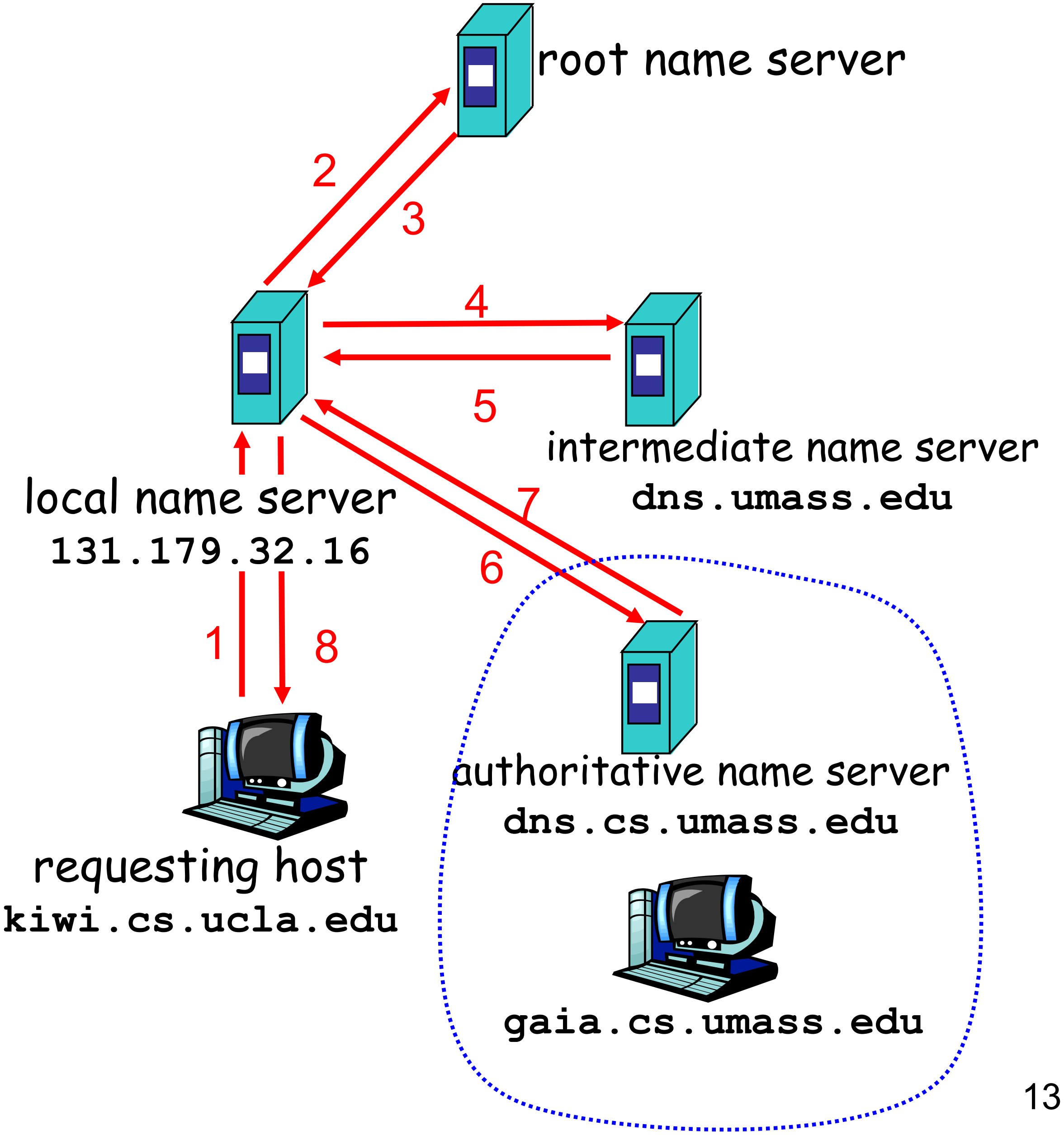
Iterative Query

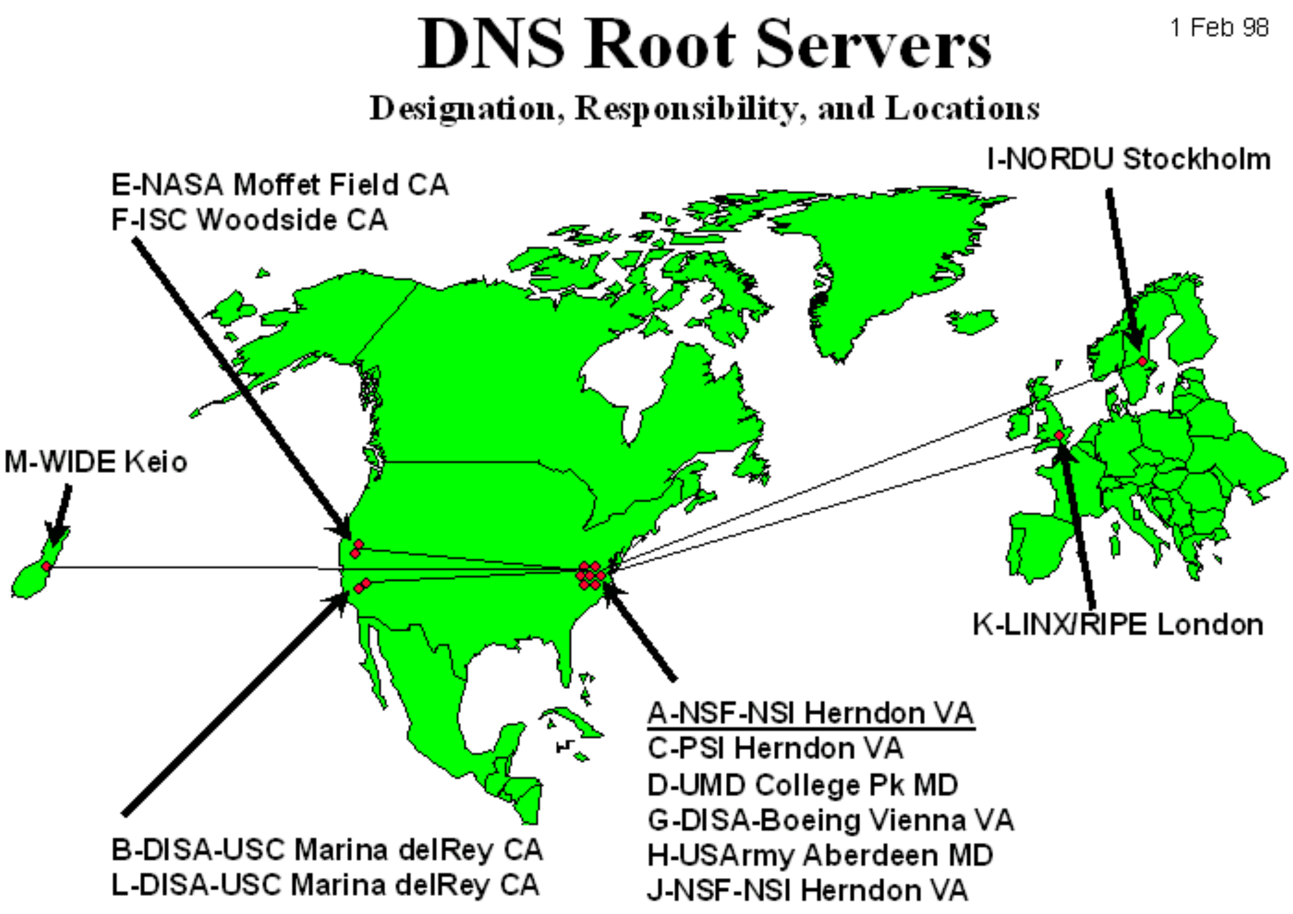
recursive query:

- puts burden of name resolution on contacted name server
- heavy load?

iterative query:

- contacted server replies with name of server to contact
- “I don’ t know this name, but ask this server”
- Most queries are done this way.





DNS Root Servers

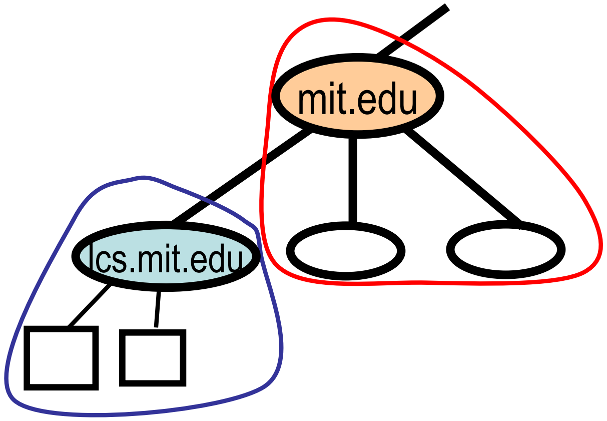
Designation, Responsibility, and Locations

1 Feb 98

- Originally 13 root name servers worldwide
- IP addresses of root servers are stored in a local file
- Now many more replicas are established around the world.

What’s in the Zone’s Master File

- Data that define the top node of the zone
  - including a list of all the servers for the zone
- Authoritative data for all hosts in the zone
- Data that describe delegated sub-zones
  - Domain name, owner, etc.
- “glue data”: For each sub-zone, list the names of its name servers, and their IP addresses.
  - To be able to direct queries to the sub-zone servers.



DNS Data Format

- Each name server maintains a collection of resource records (RR)
- Each RR contains 5 types of components
  - Name
  - Value
  - TTL: life time for cached data
  - Type: how to interpret the value
  - Class: protocol family, almost always IN
- New types of values are being invented and added into DNS over time.
  - IP address is just the first type.
  - Use separate RRs for multiple values of the same name

DNS Records

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

name is an alias name for some "canonical" (the real) name

value is the canonical name

Type=MX

name is hostname or domain

value is hostname of the mail server for the host or domain.

DNS Performance

- Virtually all Internet applications invoke DNS lookup, therefore performance is very important.
  - A significant portion of delay during web browsing comes from DNS lookup.
- Replication
  - Each zone has one or more secondary servers
  - Should be placed at diverse locations to increase system robustness.
- Caching
  - resolvers cache recent query results, till their TTLs expire.
  - Not just the end query result, but also the intermediate result
    - If a resolver caches the IP address of Google's name server, then all the queries about any host within google.com can be sent directly to Google's name server, instead of searching it from the Root server.
  - Only cache misses go to the root servers.

New Development

- Load balancing app (e.g., web) servers
  - Return the server address that's the least loaded, or closest to the client. (e.g., Akamai.com)
- Dynamic updates by end hosts
  - e.g., broadband home users (dyndns.org)
- DNS security extensions (DNSSEC)
- People use DNS for all kinds of purposes
  - Tens of applications and data types are added
    - e.g., real-time blacklist for anti-spam
  - DNS is the only Internet-scale database ubiquitously available
- Is DNS overloaded?

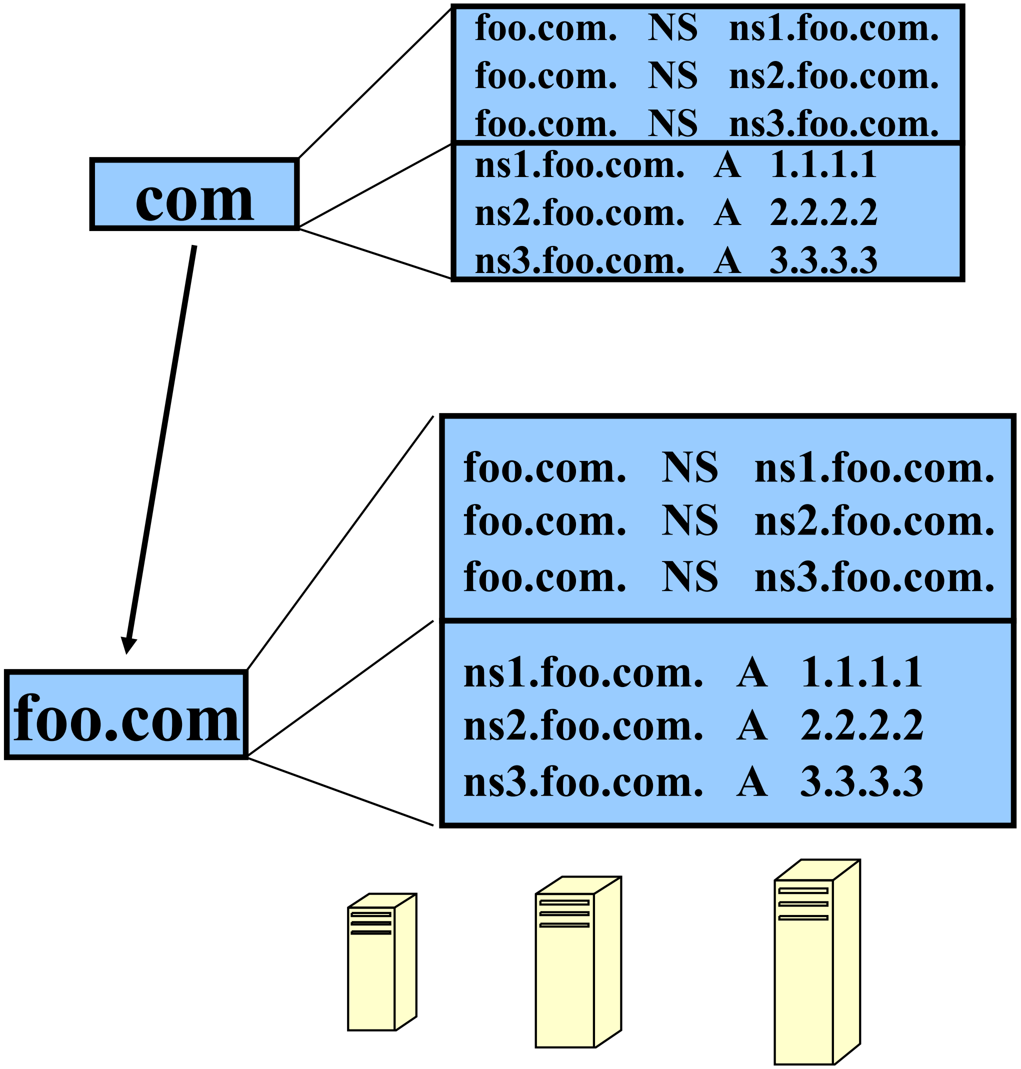
- Try user tools such as dig, nslookup, host.

Study DNS Robustness

- Classify DNS operational errors:
  - Study known errors
  - Identify new types of errors
- Measure their pervasiveness
- Quantify their impacts on DNS
  - availability
  - performance

Infrastructure RRs

- ***NS Resource Record***:
  - –Provides the names of a zone’ s authoritative servers
  - –Stored both at the parent and at the child zone
- ***A Resource Record***
  - –Associated with a NS resource record
  - –Also stored at the parent zone (glue A record)



```
foo.com.   NS   ns1.foo.com.
foo.com.   NS   ns2.foo.com.
foo.com.   NS   ns3.foo.com.
ns1.foo.com.   A   1.1.1.1
ns2.foo.com.   A   2.2.2.2
ns3.foo.com.   A   3.3.3.3
```

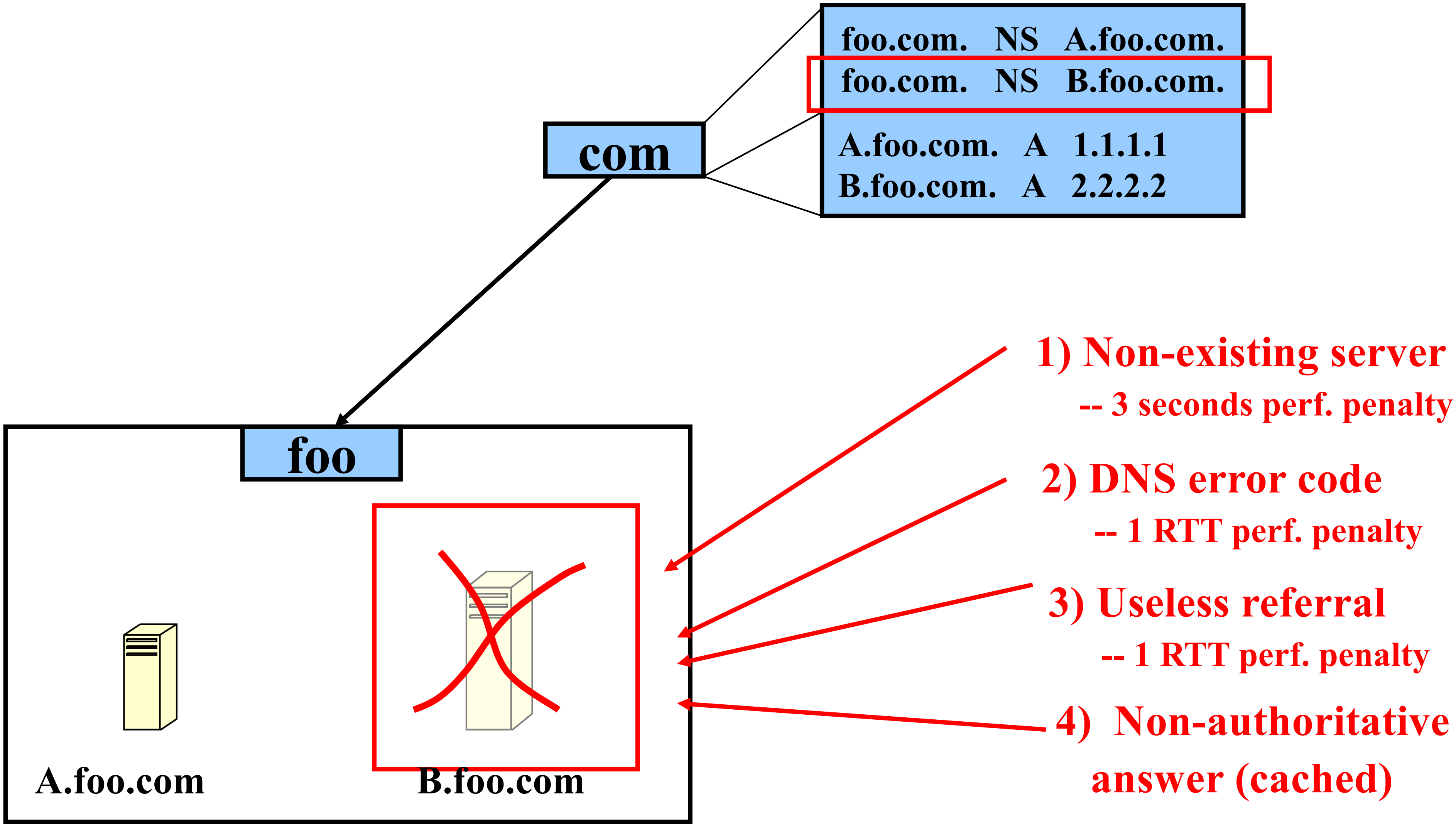
```
foo.com.   NS   ns1.foo.com.
foo.com.   NS   ns2.foo.com.
foo.com.   NS   ns3.foo.com.
ns1.foo.com.   A   1.1.1.1
ns2.foo.com.   A   2.2.2.2
ns3.foo.com.   A   3.3.3.3
```

What Affects DNS Availability

- Name Servers:
  - Software failures
  - Network failures
  - Scheduled maintenance tasks
- Infrastructure Resource Records:
  - Availability of these records
  - Configuration errors

focus of this paper

Lame Delegation



com

foo.com. NS A.foo.com.
foo.com. NS B.foo.com.

A.foo.com. A 1.1.1.1
B.foo.com. A 2.2.2.2

foo

A.foo.com

B.foo.com

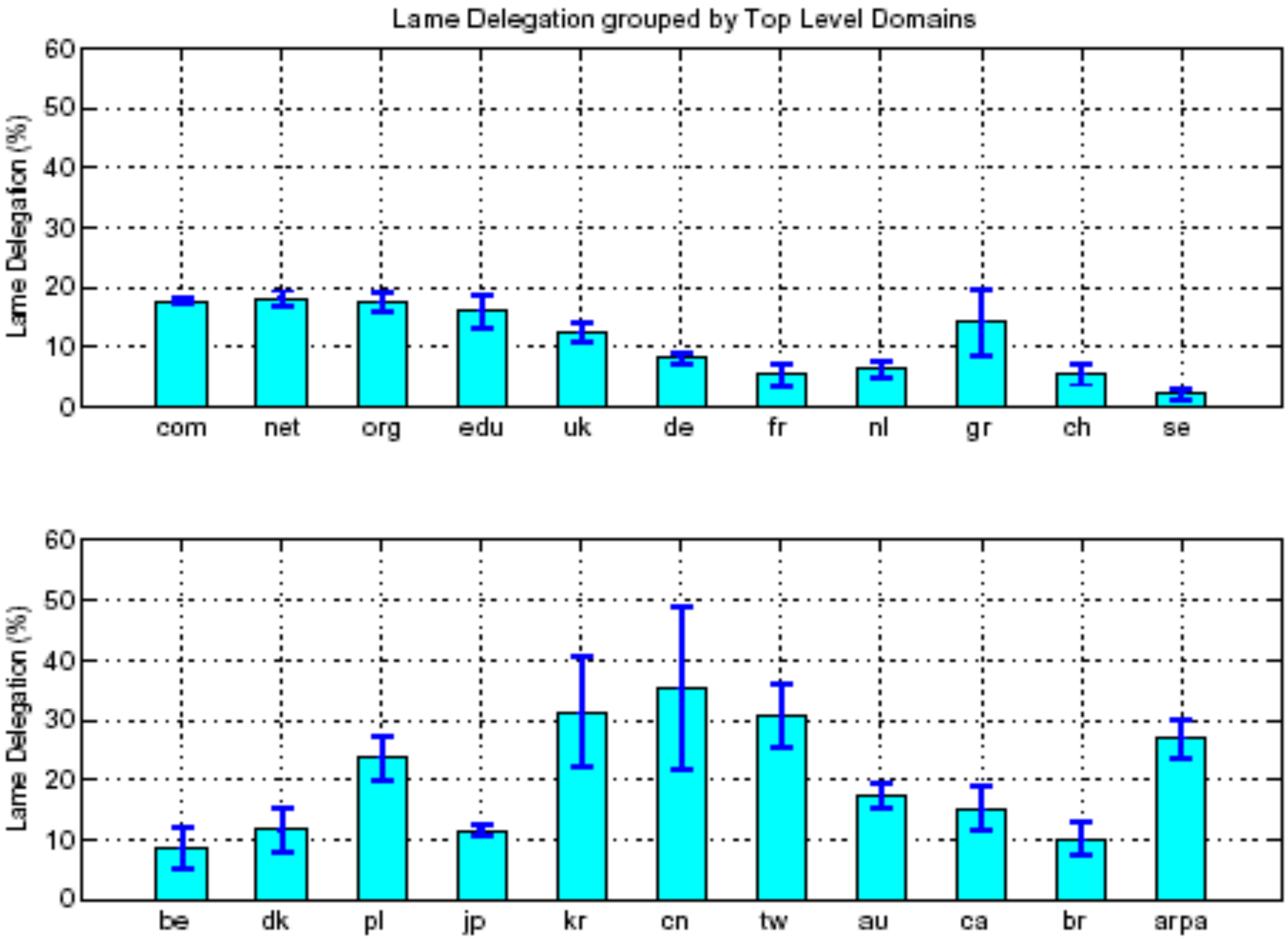
1) **Non-existing server**
-- 3 seconds perf. penalty

2) **DNS error code**
-- 1 RTT perf. penalty

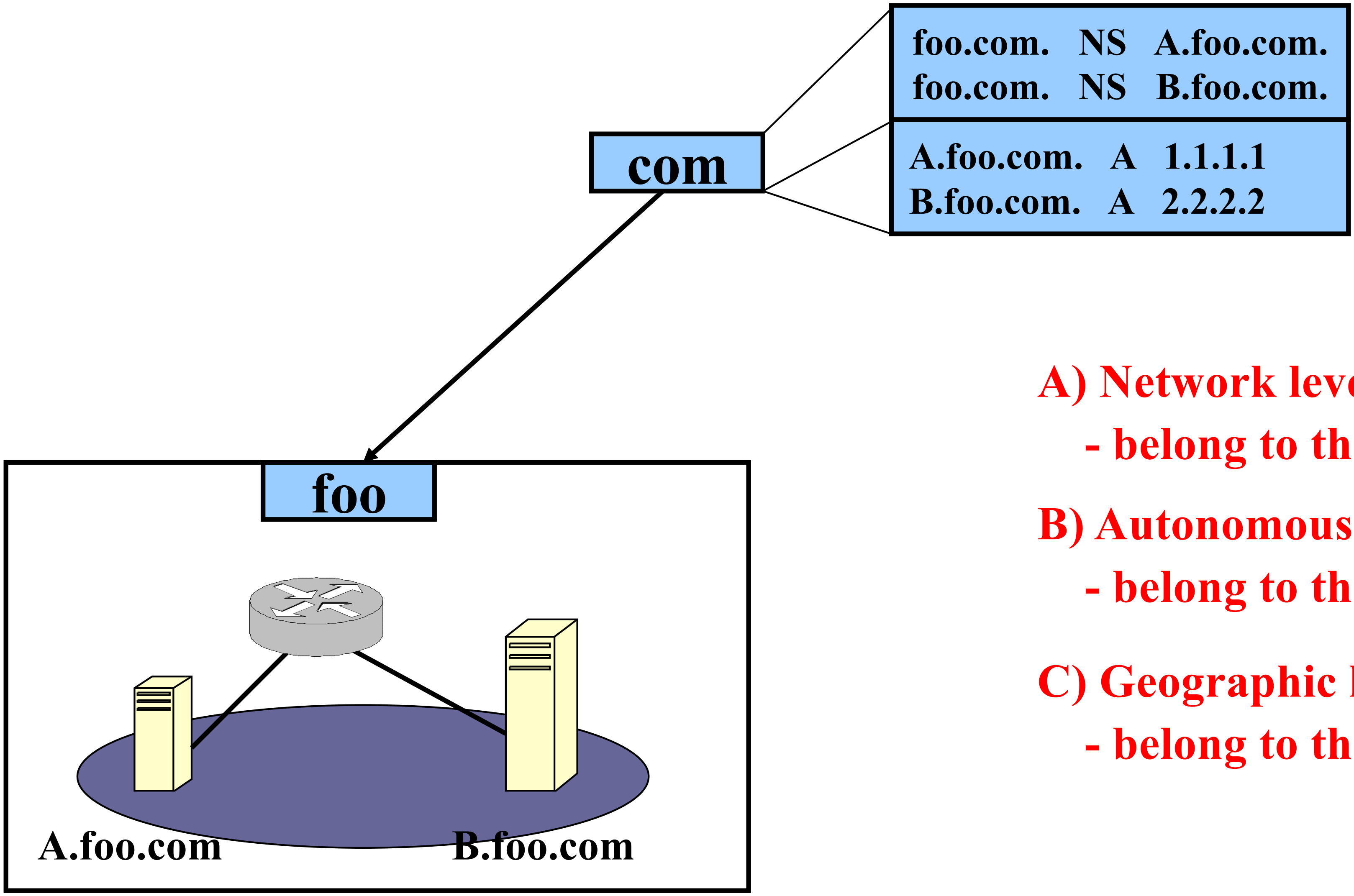
3) **Useless referral**
-- 1 RTT perf. penalty

4) **Non-authoritative answer (cached)**

Lame Delegation Results



Diminished Server Redundancy



```
foo.com.   NS   A.foo.com.
foo.com.   NS   B.foo.com.

A.foo.com.   A   1.1.1.1
B.foo.com.   A   2.2.2.2
```

A) Network level:

- belong to the same subnet

B) Autonomous system level:

- belong to the same AS

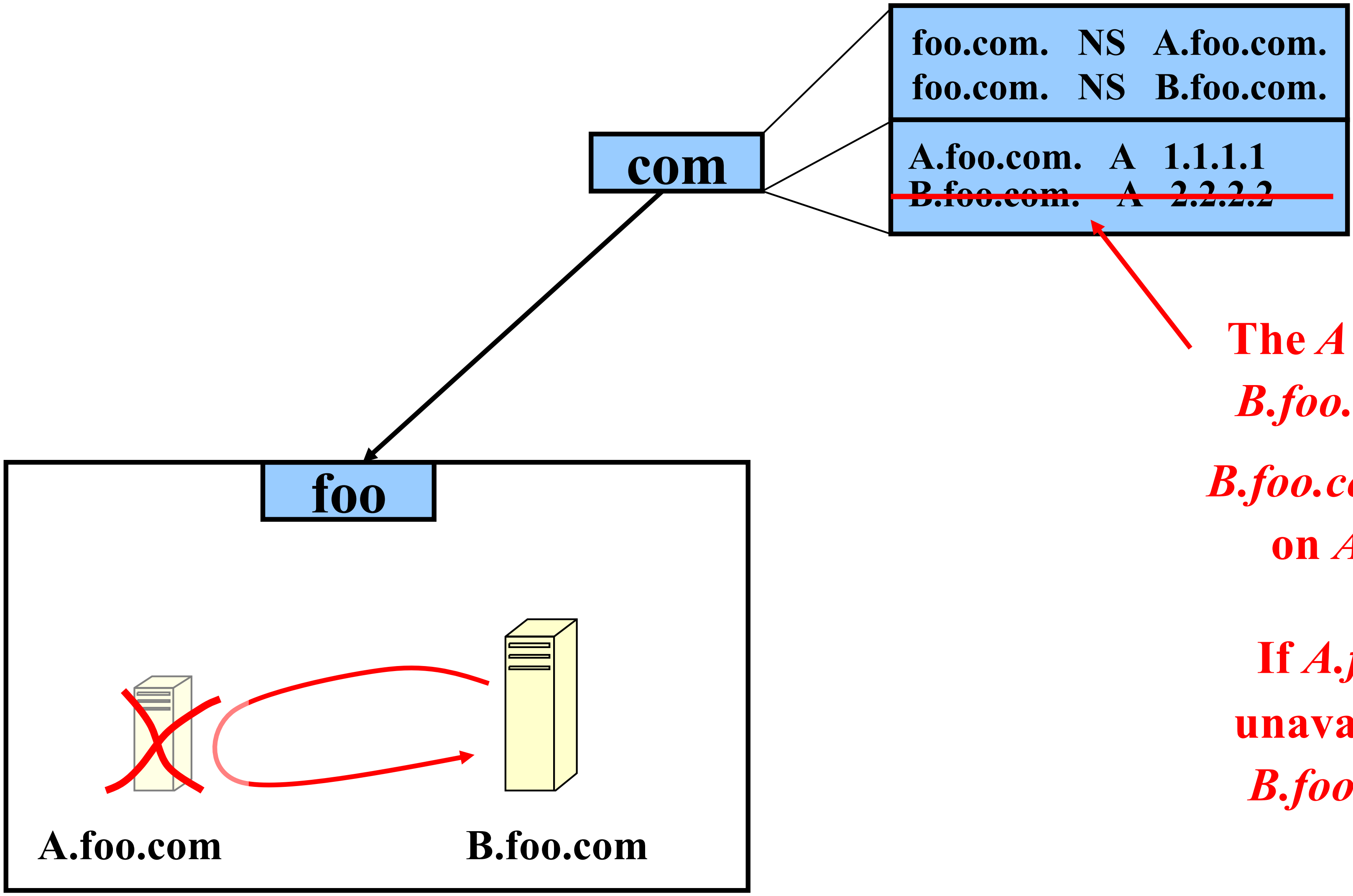
C) Geographic location level:

- belong to the same city

Diminished Server Redundancy Results

- Error Frequency:
  - 45% of all zones have all servers in the same /24 subnet
  - 75% of all zones have servers in the same AS
  - large & popular zones: better AS and geo diversity
- Impact:
  - ***less*** than **99.9**% availability: all servers in the same /24 subnet
  - ***more*** than **99.99**% availability: 3 servers at different ASes or different cities

Cyclic Zone Dependency (1)



foo.com. NS A.foo.com.
foo.com. NS B.foo.com.

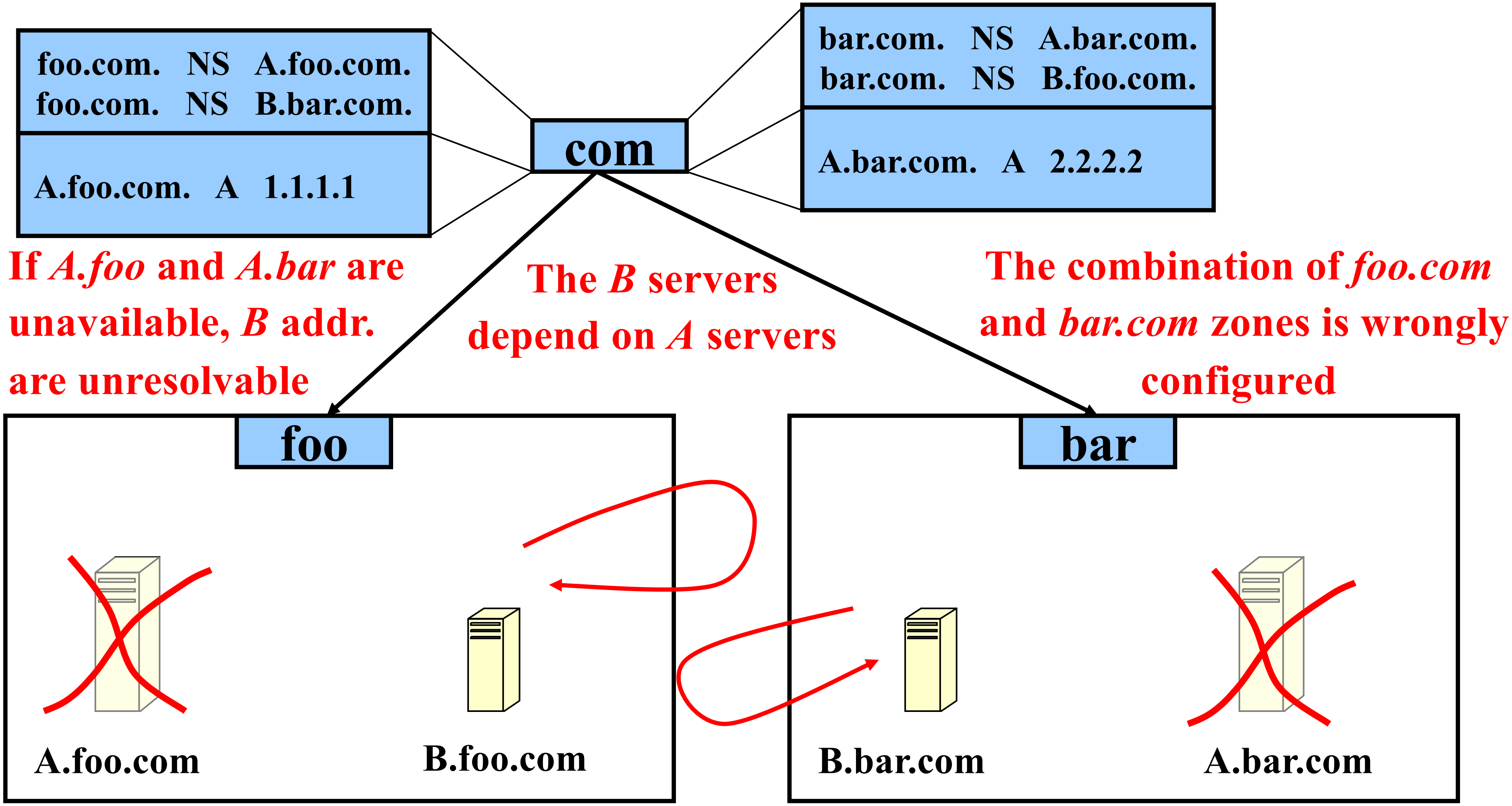
A.foo.com. A 1.1.1.1
~~B.foo.com. A 2.2.2.2~~

The *A glue* RR for *B.foo.com missing*

B.foo.com* depends on *A.foo.com

If *A.foo.com* is unavailable then *B.foo.com* is too

Cyclic Zone Dependency (2)



foo.com. NS A.foo.com.
foo.com. NS B.bar.com.

A.foo.com. A 1.1.1.1

com

bar.com. NS A.bar.com.
bar.com. NS B.foo.com.

A.bar.com. A 2.2.2.2

If *A.foo* and *A.bar* are unavailable, *B* addr. are unresolvable

The *B* servers depend on *A* servers

The combination of *foo.com* and *bar.com* zones is wrongly configured

Cyclic Zone Dependency Results

- Error Frequency:
  - 2% of the zones
  - None of the 500 most popular zones
- Impact:
  - 90% of the zones with cyclic dependency errors lose 25% (or even more) of their servers
  - 2 or 4 zones are involved in most errors

Robustness

- User-perceived robustness:
  - Data replication: *only one server is needed*
  - Data caching: *temporarily masks infrastructure failures*
  - Popular zones: *fewer configuration errors*
- System robustness:
  - Fewer available servers: *due to inconsistency errors*
  - Fewer redundant servers: *due to dependency errors*

Summary on DNS Misconfiguration

- DNS operational errors are *widespread*
- DNS operational errors affect *availability:*
  - 50% of the servers lost
  - less than 99.9% availability
- DNS operational errors affect *performance:*
  - one or even two orders of magnitude
- DNS system robustness lower than user perception
  - Due to protocol design, not just due to operator errors