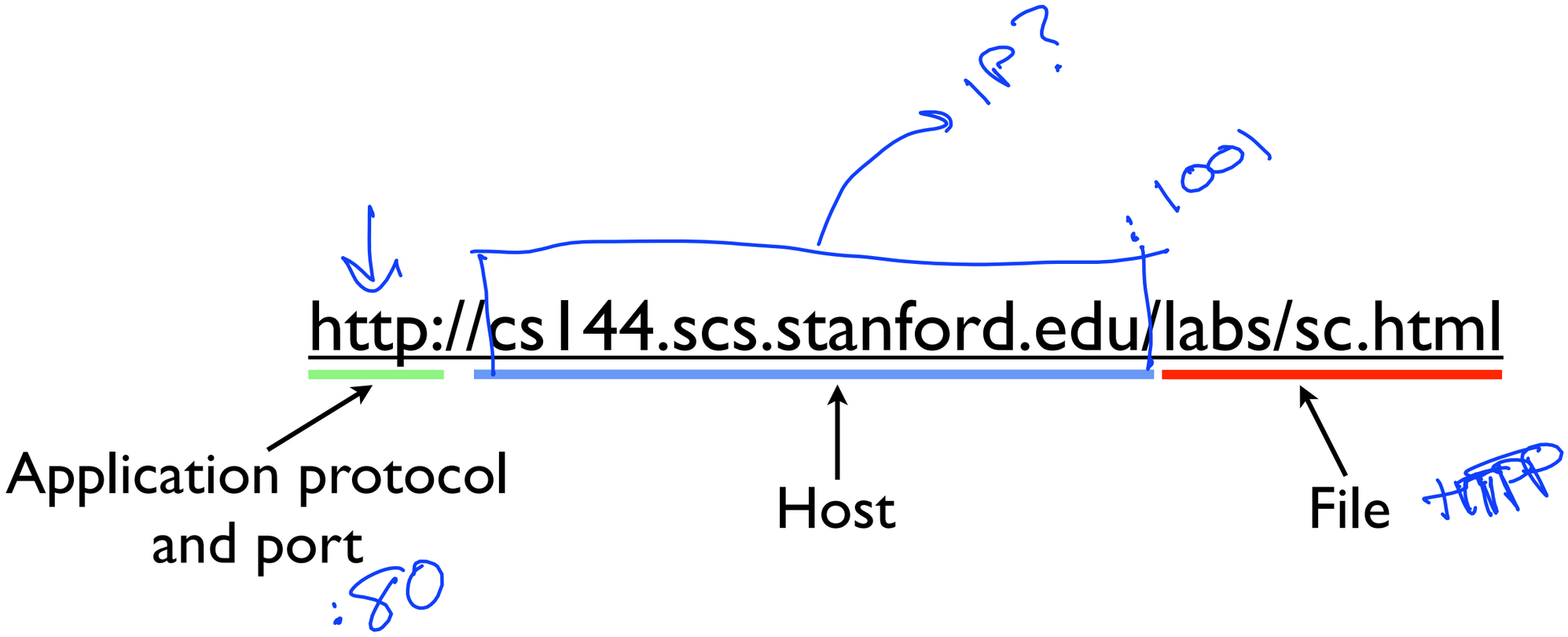
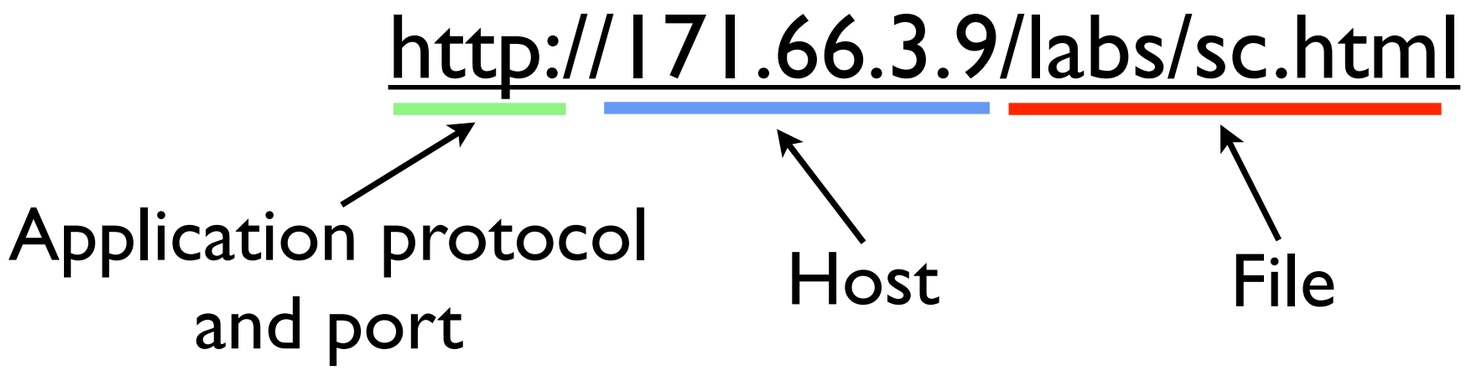
Domain Name System (DNS)

Parsing a URL



Parsing a URL



HOSTS.TXT

- Originally, all hosts were in a file HOSTS.TXT, maintained by Network Information Center
  - Maintained at SR: SRI-NIC.ARPA, 26.0.0.73 (RFC952)
- Hosts periodically used a file transfer protocol to download new version
  - Requires n^2 network capacity, does not scale well

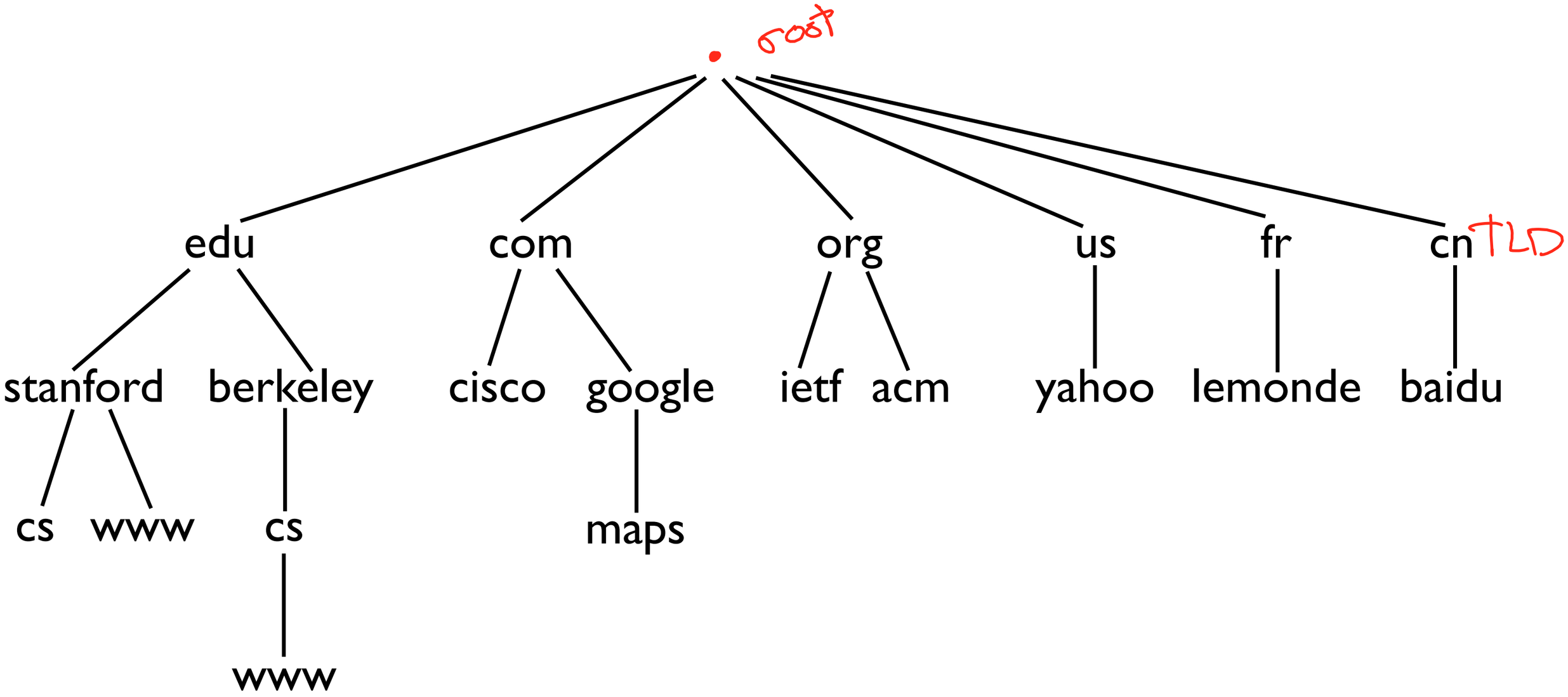
Domain Name System

- Map names to addresses (more generally, values)
- Must be able to handle *huge* number of records
- Must have distributed control: people can control their own names
- Must be robust to individual node failures

Domain Name System Design

- Two properties make DNS design feasible
  - Read-only or read-mostly database: hosts look up names much more often than update them
  - Loose consistency: changes can take a little while to propagate
- Two properties allow extensive caching
  - Look up a name, keep result for a long time

DNS Name Architecture

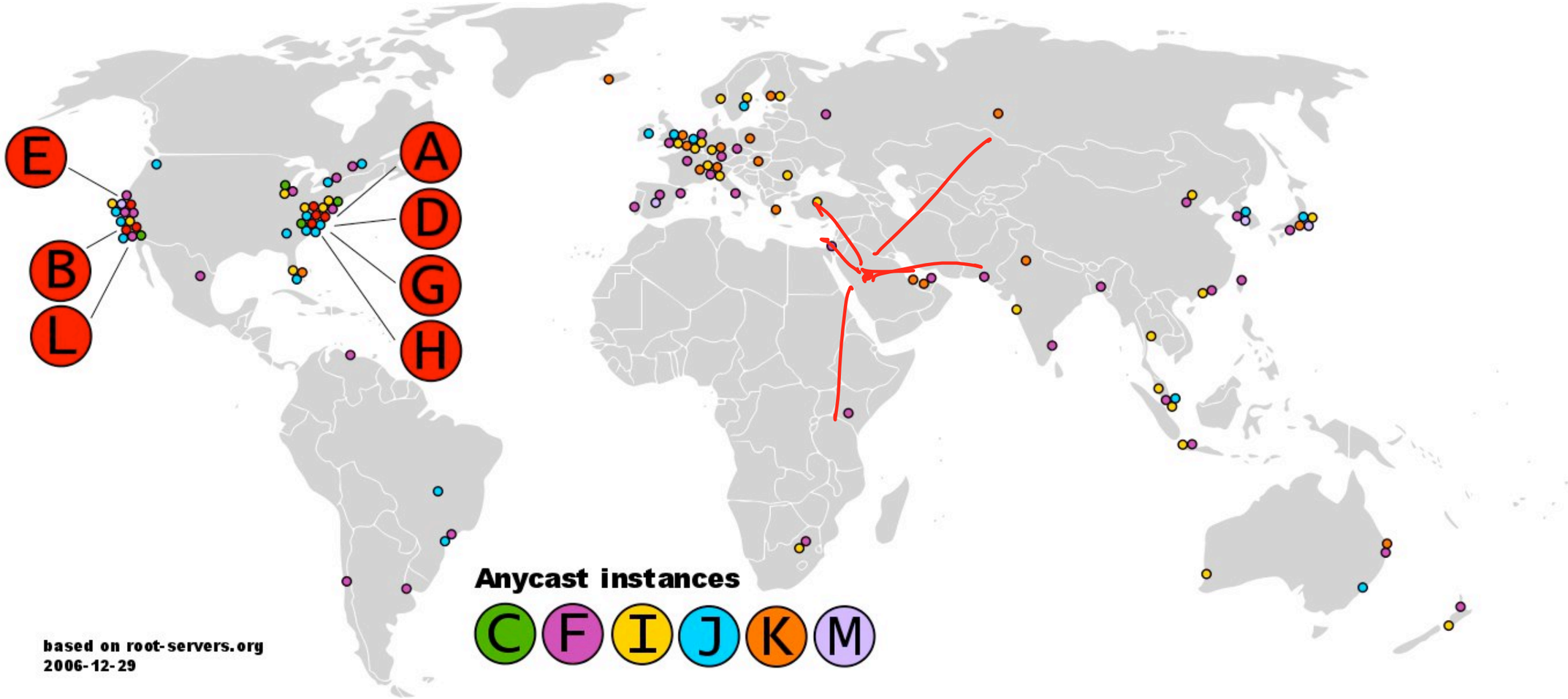


DNS Servers

- Hierarchical zones ("root" zone, edu, stanford, scs)
- Each zone can be separately administered
- Each zone served from several replicated servers
- Root zone: 13 servers, highly replicated (a, b, c, ... m)
  - Bootstrap: root server IPs are stored in a file on name server
  - Replicated through anycast (discussed later in course)

DDoS "root servers"

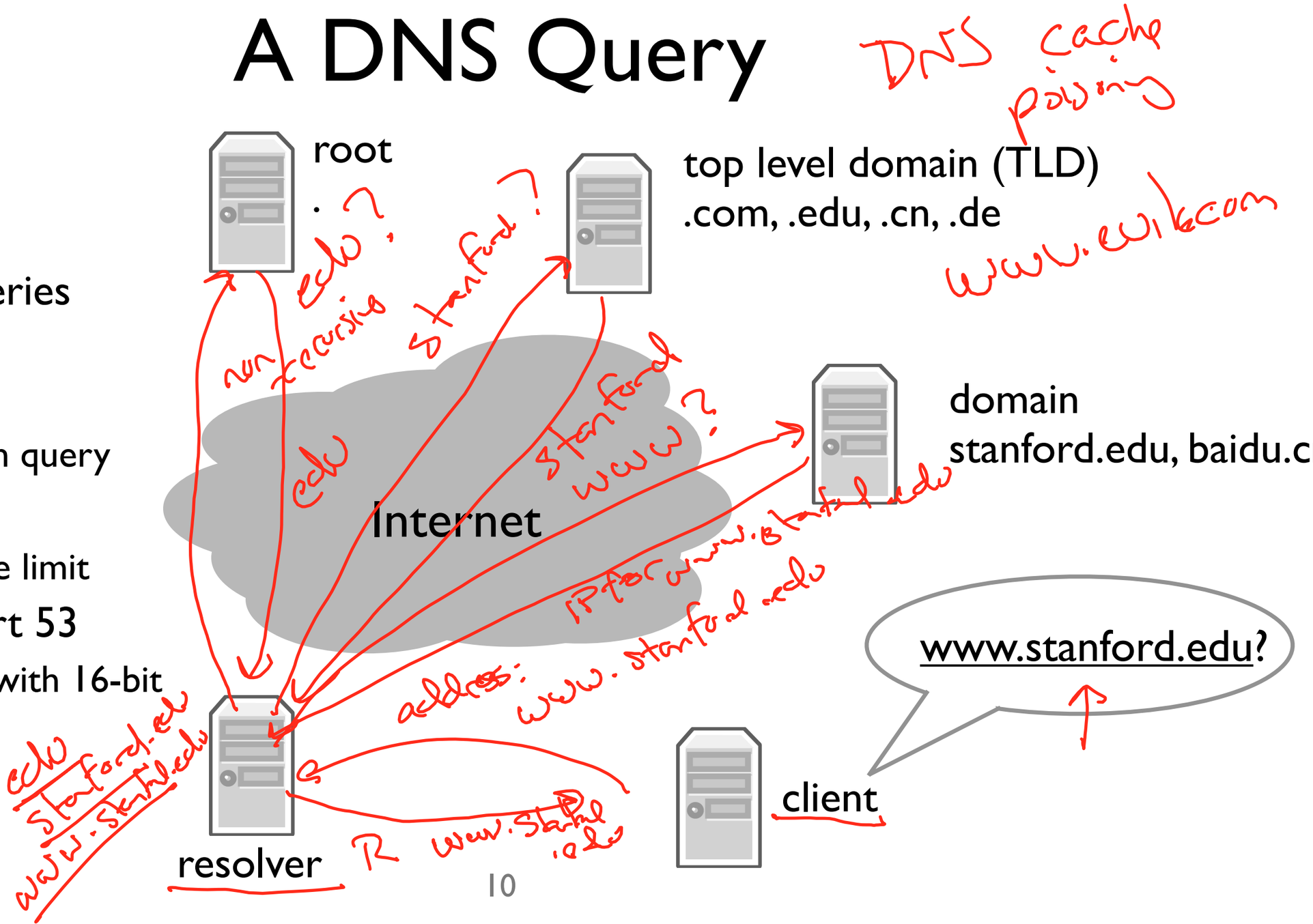
DNS Root Servers



http://en.wikipedia.org/wiki/File:Root-current.svg

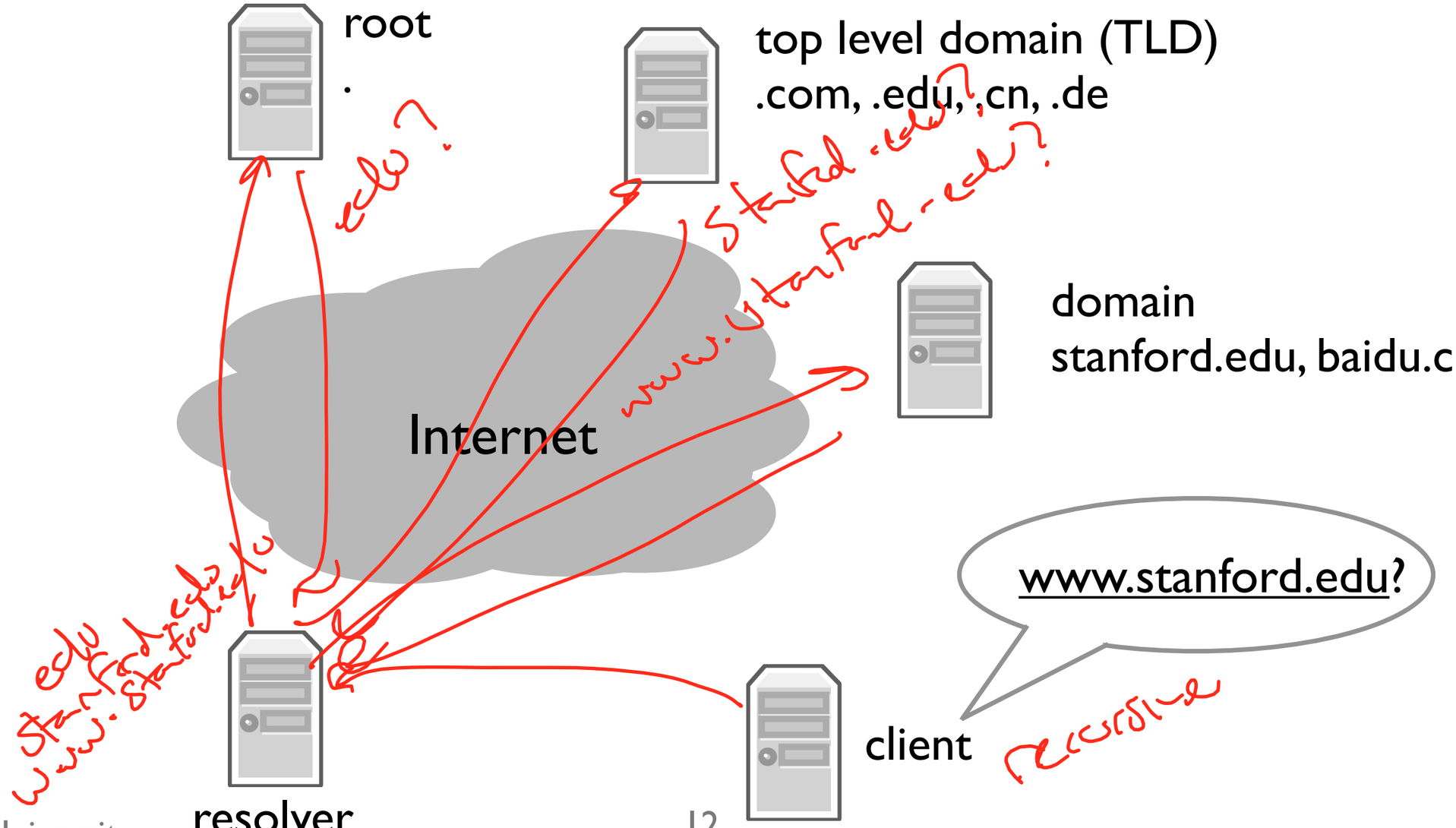
A DNS Query

- Two types of queries
  - Recursive
  - Non-recursive
  - Specified by bit in query
- UDP port 53
  - 512 byte message limit
- Can use TCP port 53
  - Prefix messages with 16-bit length field



DNS: Queries and Resource Records

A DNS Query


root
.
top level domain (TLD)
.com, .edu, .cn, .de
domain
stanford.edu, baidu.cn
Internet
www.stanford.edu?
client
resolver
recursive

Resource Records

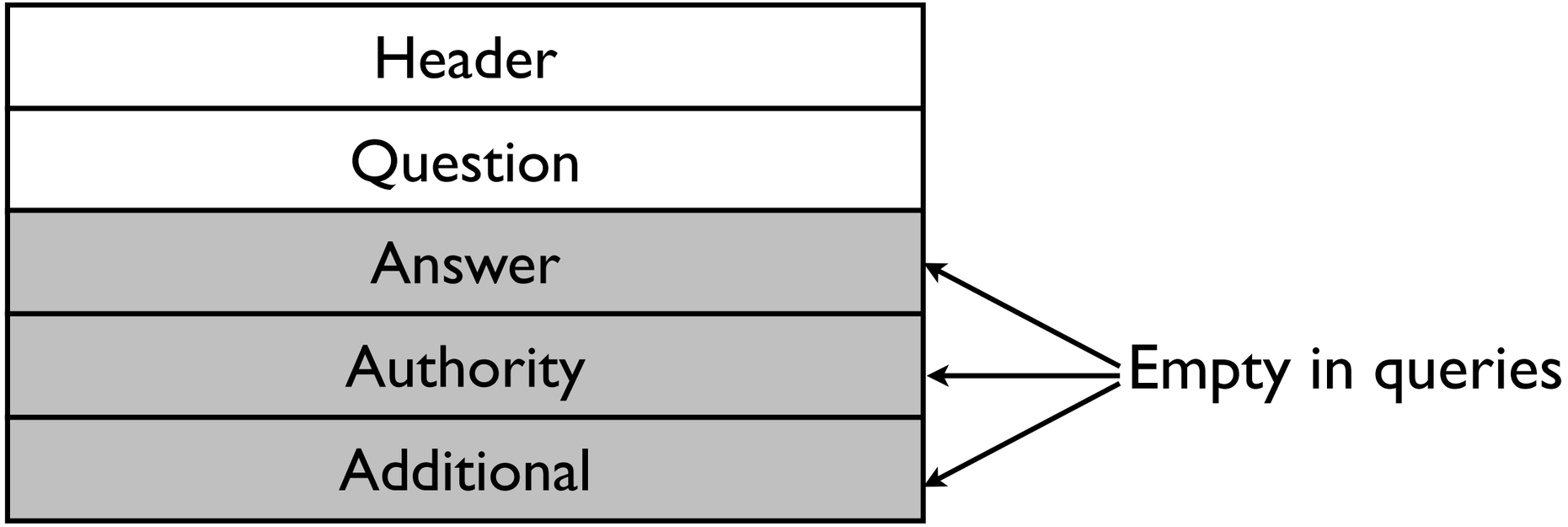
- All DNS information represented in Resource Records (RRs):

  name [TTL] [class] type rdata

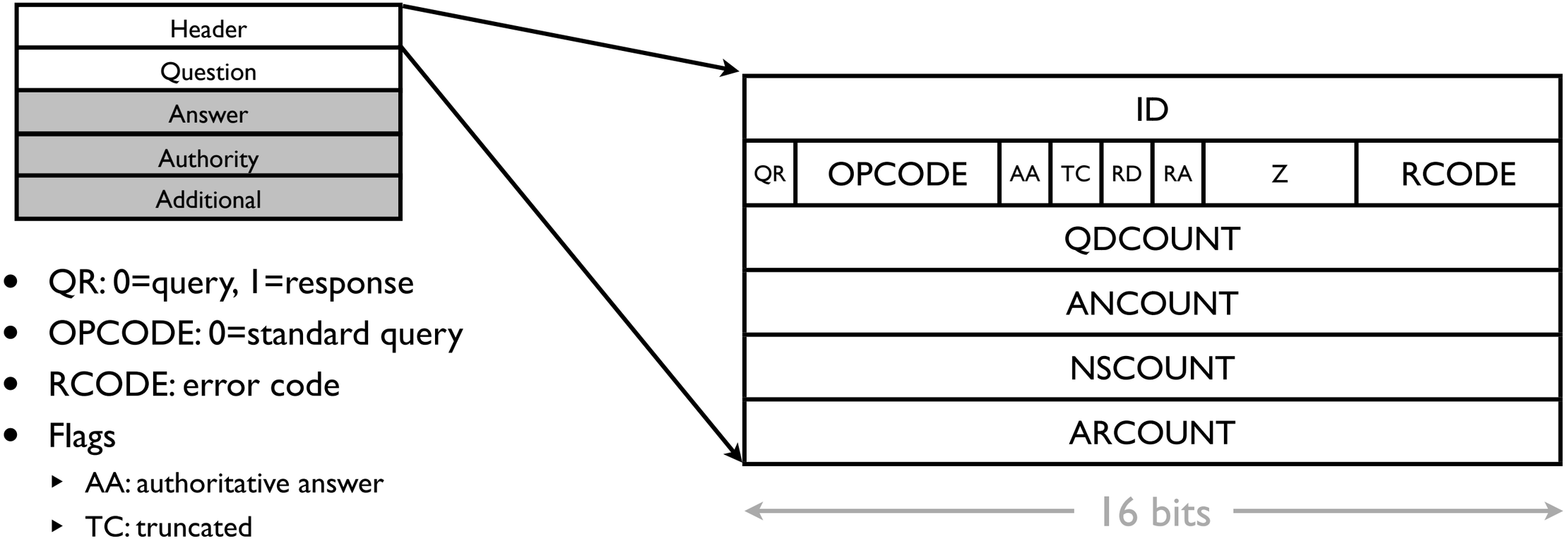
  - *name*: domain name (e.g., www.stanford.edu)
  - *TTL*: time to live (in seconds)
  - *class*: for extensibility, usually `IN 1` (Internet)
  - *type*: type of the record
  - *rdata*: resource data dependent on *type*

- Two critical RR types: A (IPv4 address) and NS (name server) records
- `dig` tool

DNS Message Structure (RFC1035)

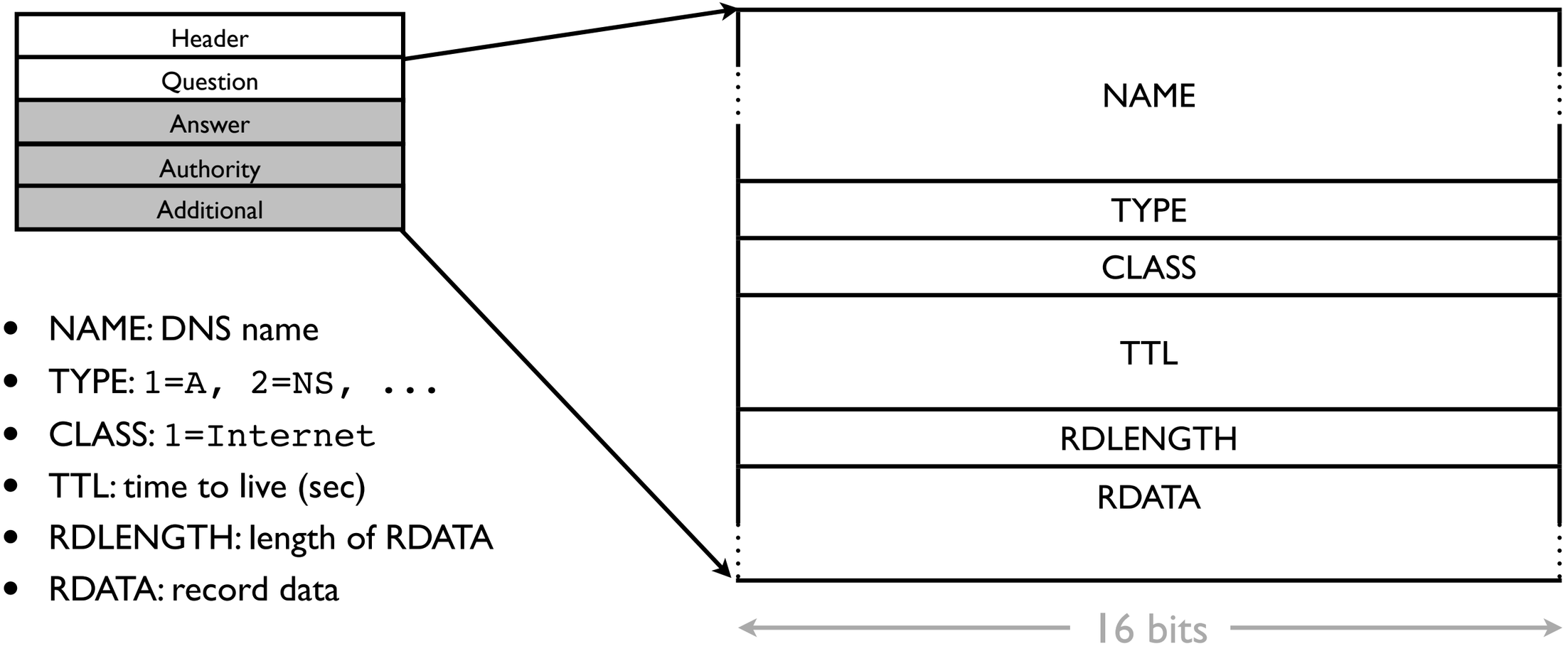


DNS Header Structure (RFC1035)



- QR: 0=query, 1=response
- OPCODE: 0=standard query
- RCODE: error code
- Flags
  - AA: authoritative answer
  - TC: truncated
  - RD: recursion desired
  - RA: recursion available

DNS RR Structure (RFC1035)

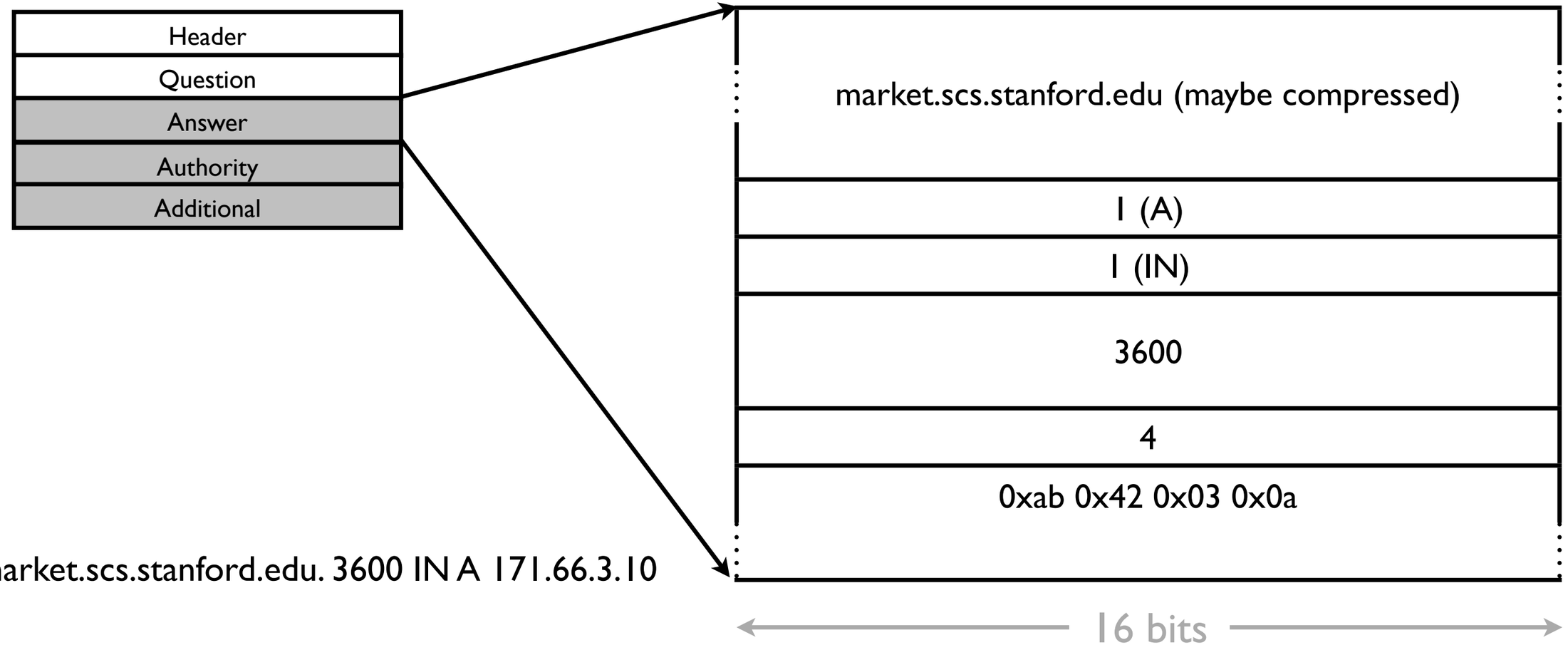


- NAME: DNS name
- TYPE: `1=A, 2=NS, ...`
- CLASS: `1=Internet`
- TTL: time to live (sec)
- RDLENGTH: length of RDATA
- RDATA: record data

DNS Name Compression

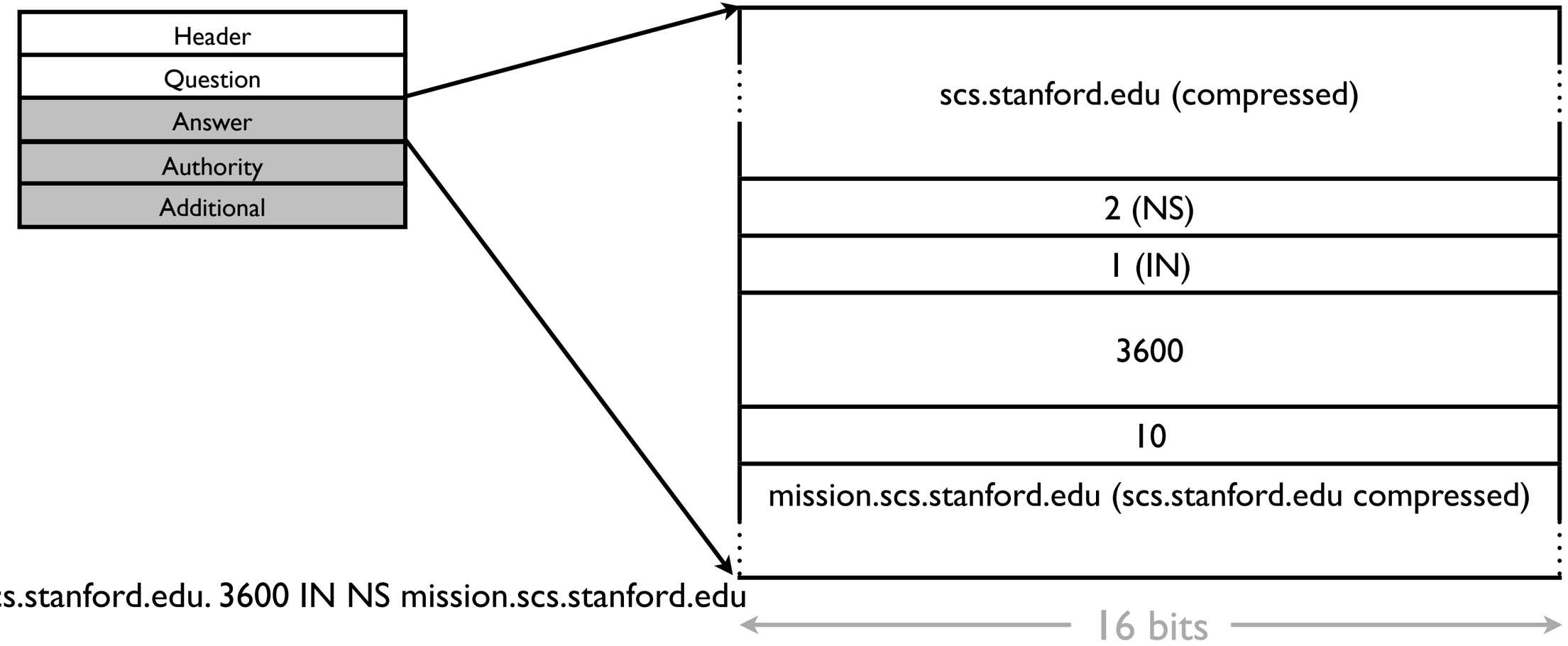
- Names can be long and repeated several times in a packet
  - Query/answer
  - NS record/A record
- Break names into labels: www.stanford.edu is www, stanford, and edu
- Each label is encoded as length, text: 3www, 8stanford, 3edu
  - Length is binary
  - Text is ASCII: 3www is 0x0377 0x7777
- If length >= 192, next 14 bits specifies offset in packet of name
  - 0xc00c means name is at offset 0xc00c-0xc000 = 0x0c = 12

DNS A Record



market.scs.stanford.edu. 3600 IN A 171.66.3.10

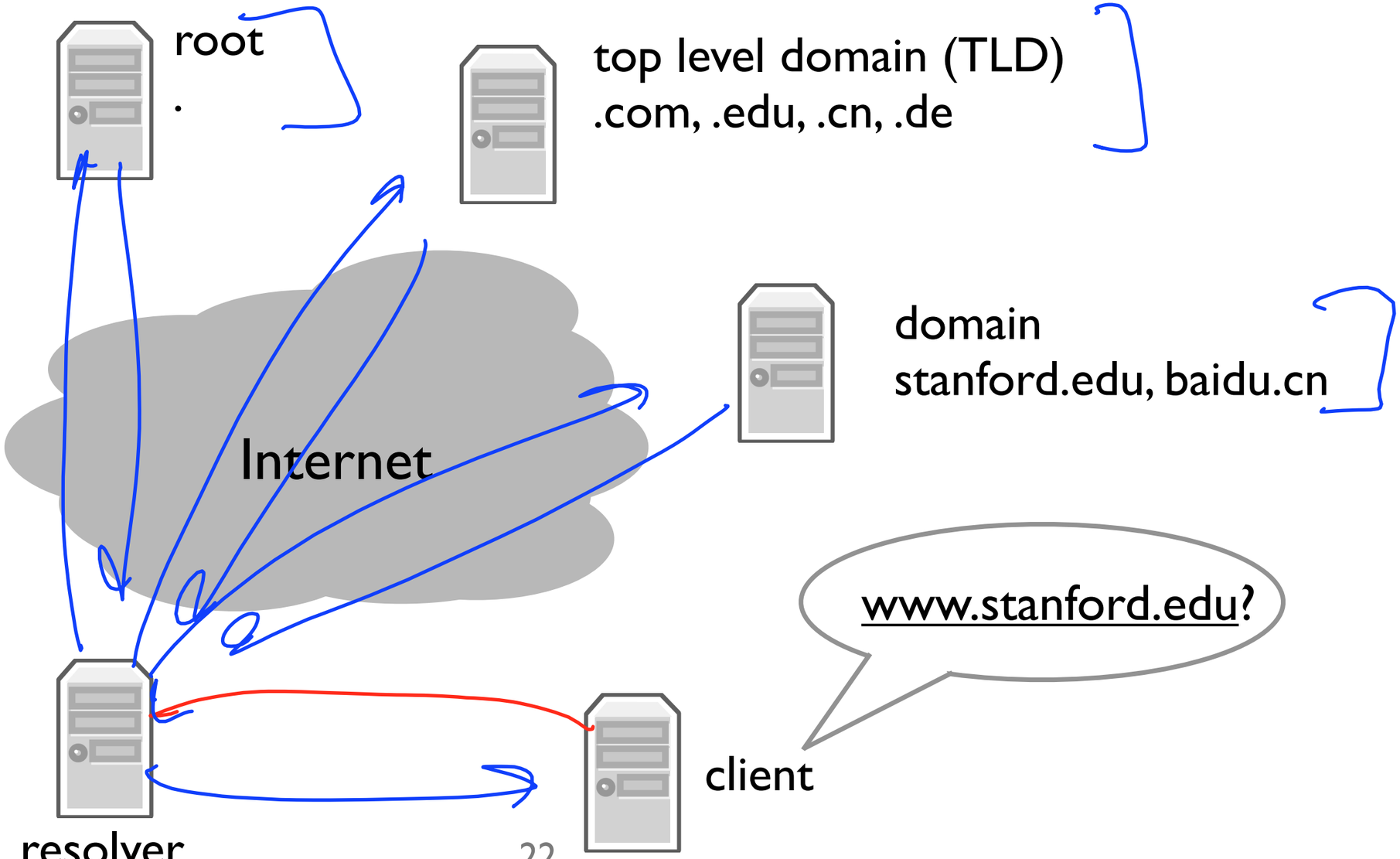
DNS NS Record



DNS Wireshark Example

DNS Details

Details


root
.
top level domain (TLD)
.com, .edu, .cn, .de
domain
stanford.edu, baidu.cn
Internet
www.stanford.edu?
client
resolver

Traversing Zones

- Bootstrap: local name server has a *root cache* file (specifies root servers)
  - Starting set of name/address mappings to query
- Recursion: how do you get the address of name servers?
  - Remember, NS records have host names
  - argus.stanford.edu is the name server for stanford.edu
  - How do you contact argus?
- Solution to recursion: *glue records*, A records in parent zone
  - The .edu name servers have NS records for stanford.edu
  - The .edu name servers also have A records for argus.stanford.edu

Glue Record Example

- Look up www.scs.stanford.edu assuming no cache
  - `dig +norec www.scs.stanford.edu @a.root-servers.net`
  - `dig +norec www.scs.stanford.edu @a.edu-servers.net`
  - `dig +norec www.scs.stanford.edu @argus.stanford.edu`
  - `dig +norec www.scs.stanford.edu @ns1.fs.net`

CNAME Record

- Canonical name record -- tells you a name is an alias

name [TTL] [class] CNAME canonical-name

  - Any record for canonical name can also be associated with name
  - Example: dig www.stanford.edu
- CNAME precludes any other RRs for name
- Answer can have other records for canonical name

MX Records

- Mail eXchange record -- tells you mail server for a domain

  name [TTL] [class] MX preference mail-server-name

- Can't ping scs.stanford.edu, but you can send email to scs.stanford.edu
- MX records cause A record processing for *mail-server-name*
- Example: `dig mx scs.stanford.edu`
- What if *mail-server-name* does not have an A record?
  - `dig mx bad-mx.scs.stanford.edu`

Many Other Kinds of Records

- SOA: Start of Authority
- TXT: arbitrary text (great for extensions)
- PTR: map address to name
- AAAA: IPv6 address records