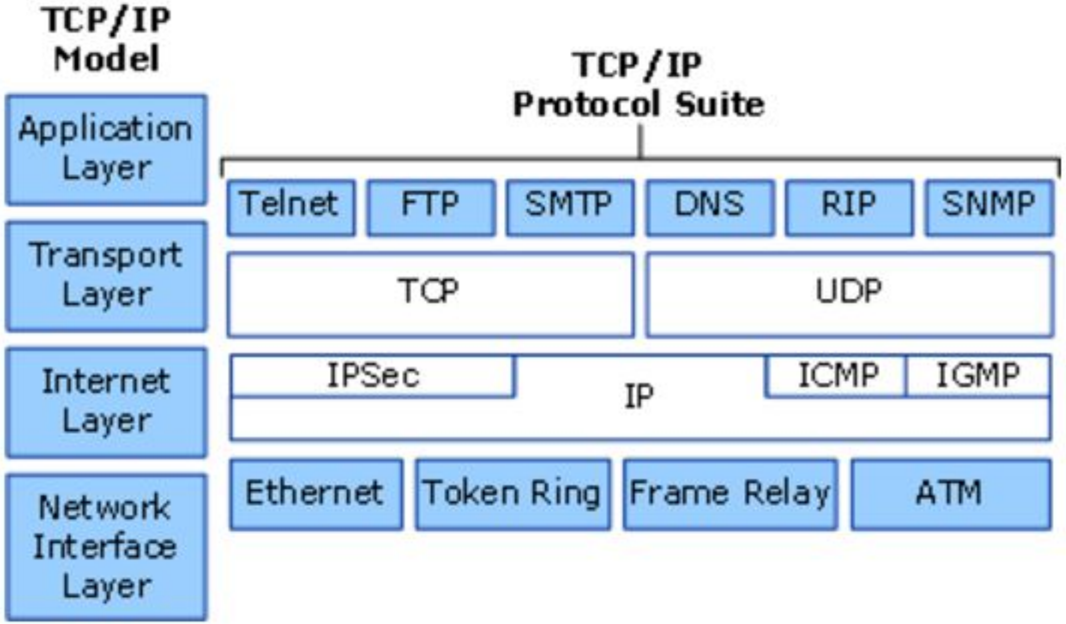
CSE 410/565: Computer Security

Instructor: Dr. Ziming Zhao

Application Layer



Domain Name System

- Users generally prefer names to numbers. Computers prefer numbers to names. DNS provides the mapping between the two

- What Internet users use to reference anything by name on the Internet

- A lookup mechanism by which Internet software translates names to attributes such as addresses

History of DNS

- ARPANET utilized a central file HOSTS.TXT
  - Contains names to addresses mapping
  - Maintained by SRI’s NIC (Stanford-Research-Institute: Network-Information-Center)

- Administrators email changes to NIC
  - NIC updates HOSTS.TXT periodically
- Administrators FTP (download) HOSTS.TXT

History of DNS

- As the system grew, HOSTS.TXT had problems with:
  - Scalability (traffic and load)
  - Name collisions
  - Consistency

- In 1984, Paul Mockapetris released the first version (RFCs 882 and 883, superseded by 1034 and 1035 …)

- Early 1980s, BIND (Berkeley Internet Name Domain) software was developed

Uniform Resource Identifier

```
scheme:[//[user:password@]host[:port]][/]path[?query][#fragment]
```

Examples

- ftp://*username*:*password*@hostname/
- http://www.ietf.org/rfc/rfc2396.txt
- mailto:John.Doe@example.com
- news:comp.infosystems.www.servers.unix
- telnet://melvyl.ucop.edu/

- Host: a registered domain name or IP address

Domain Name System

A globally distributed, scalable, reliable database

Comprised of three components

- A “name space”
- Servers making that name space available
- Resolvers (clients) which query the servers about the name space

How many domain names?

VERISIGN

« Previous Release | Next Release »

Jun 30, 2015

Internet Grows to 294 Million Domain Names in the First Quarter of 2015

- Data is maintained locally, but retrievable globally
  - No single computer has all DNS data
- DNS lookups can be performed by any device
- Remote DNS data is locally cacheable to improve performance

Scalability

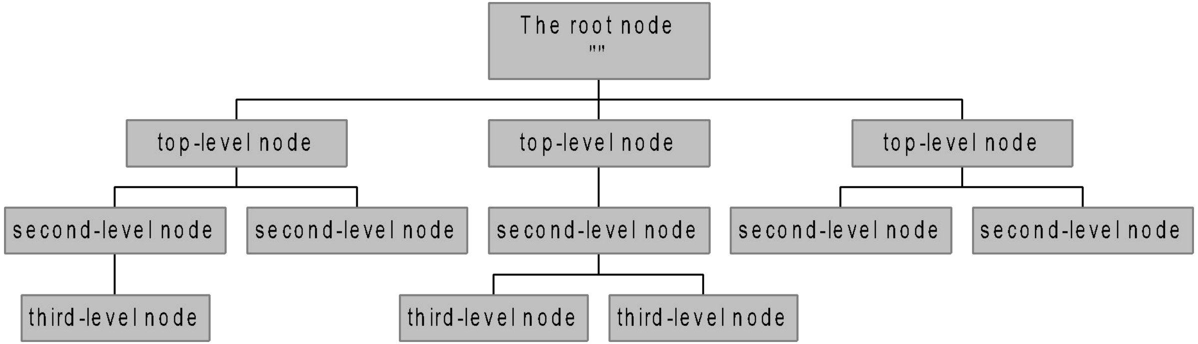
- No limit to the size of the database
- No limit to the number of queries
    - Tens of thousands of queries handled easily every second
- Queries distributed among masters, slaves, and caches

Reliability

- Data is replicated
  - Data from master is copied to multiple slaves
- Clients can query
  - Master server
  - Any of the copies at slave servers
- Clients will typically query local caches
- DNS protocols can use either UDP or TCP
  - If UDP, DNS protocol can handle retransmission, sequencing, etc.

The Name Space

- The *name space* is the structure of the DNS database
  - An inverted tree with the root node at the top
- Each node has a label
  - The root node has a null label, written as “”

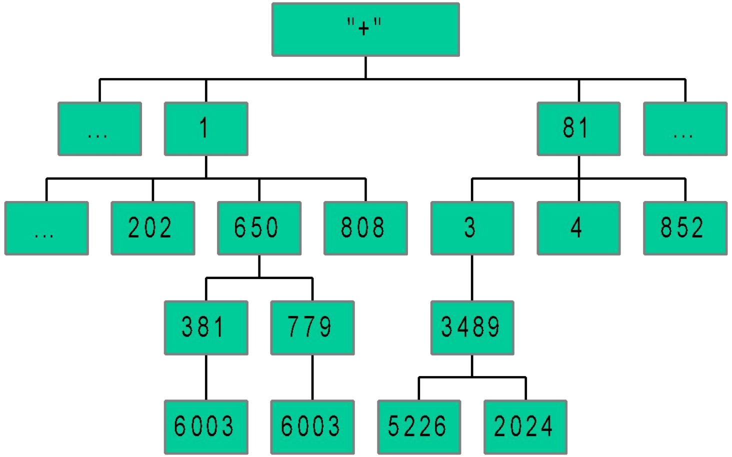


An Analogy – E.164

Root node maintained by the ITU (call it "+")

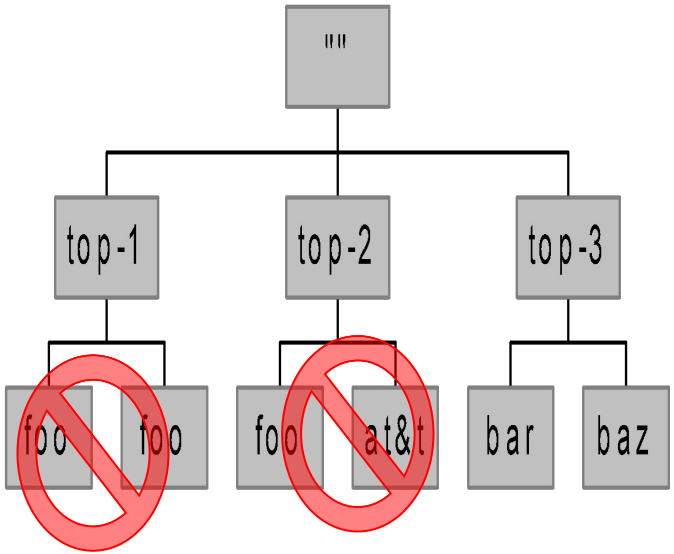
Top level nodes = country codes (1, 81, etc)

Second level nodes = regional codes (1-402, 81-3, etc.)



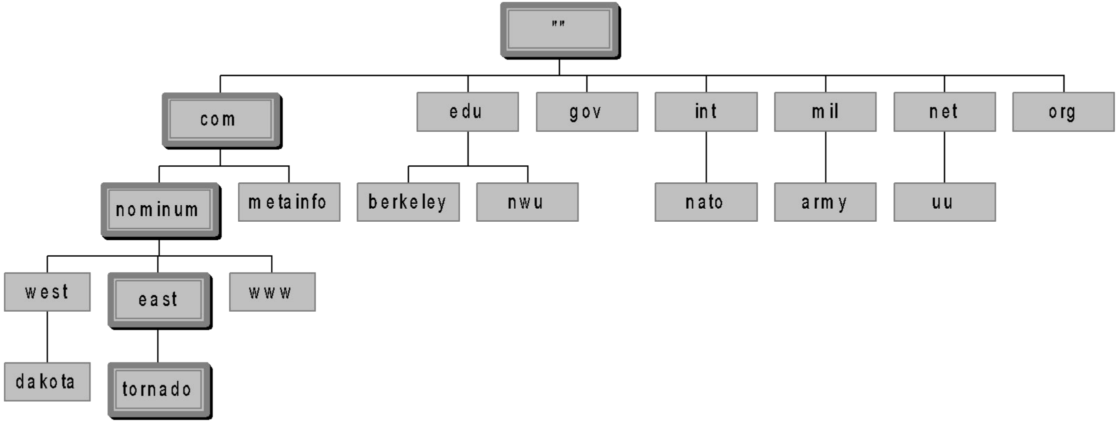
Labels

- Each node in the tree must have a label
  - A string of up to 63 bytes
  - RFCs 852 and 1123 define legal characters for “hostnames”
    - A-Z, 0-9, and “-” only with a-z and A-Z treated as the same
- Sibling nodes must have unique labels
- The null label is reserved for the root node

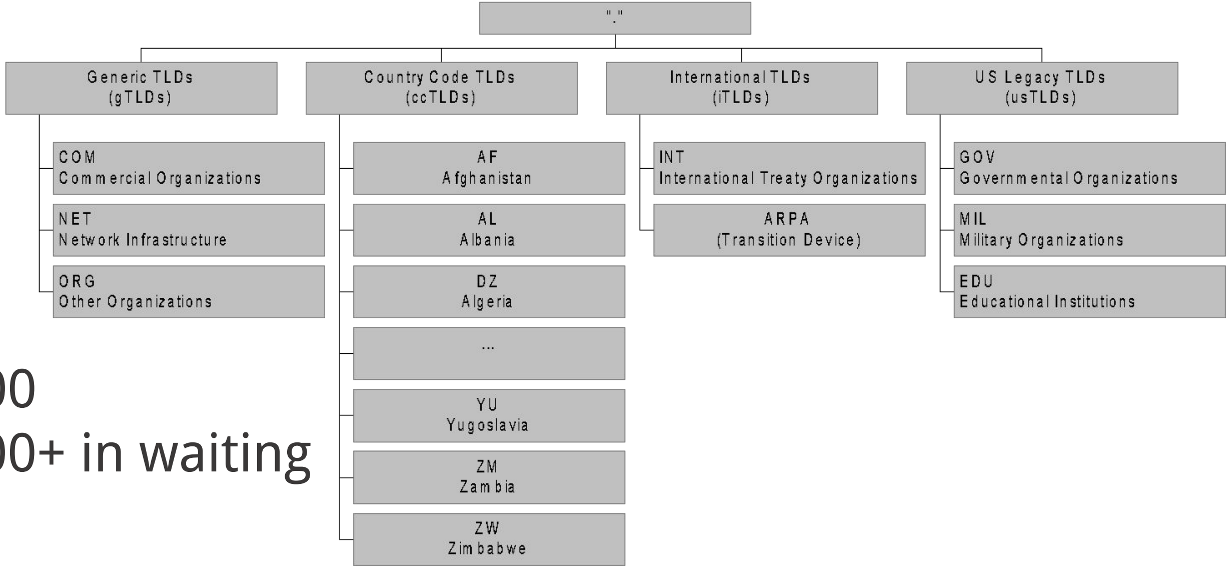


Domain Names

- A domain name is the sequence of labels from a node to the root, separated by dots (“.”s), read left to right
  - The name space has a maximum depth of 127 levels
  - Domain names are limited to 255 characters in length
- A node’s domain name identifies its position in the name space



Top-level Domain



>700
1900+ in waiting list

Subdomains

One domain is a subdomain of another if its domain name ends in the other's domain name

- So *cse.ub.edu* is a subdomain of
  - *ub.edu & edu*
- *ub.edu* is a subdomain of *edu*

Delegation

Administrators can create subdomains to group hosts

- According to geography, organizational affiliation etc.

An administrator of a domain can delegate responsibility for managing a subdomain to someone else

The parent domain retains links to the delegated subdomains

Delegation Creates Zones

Each time an administrator delegates a subdomain, a new unit of administration is created

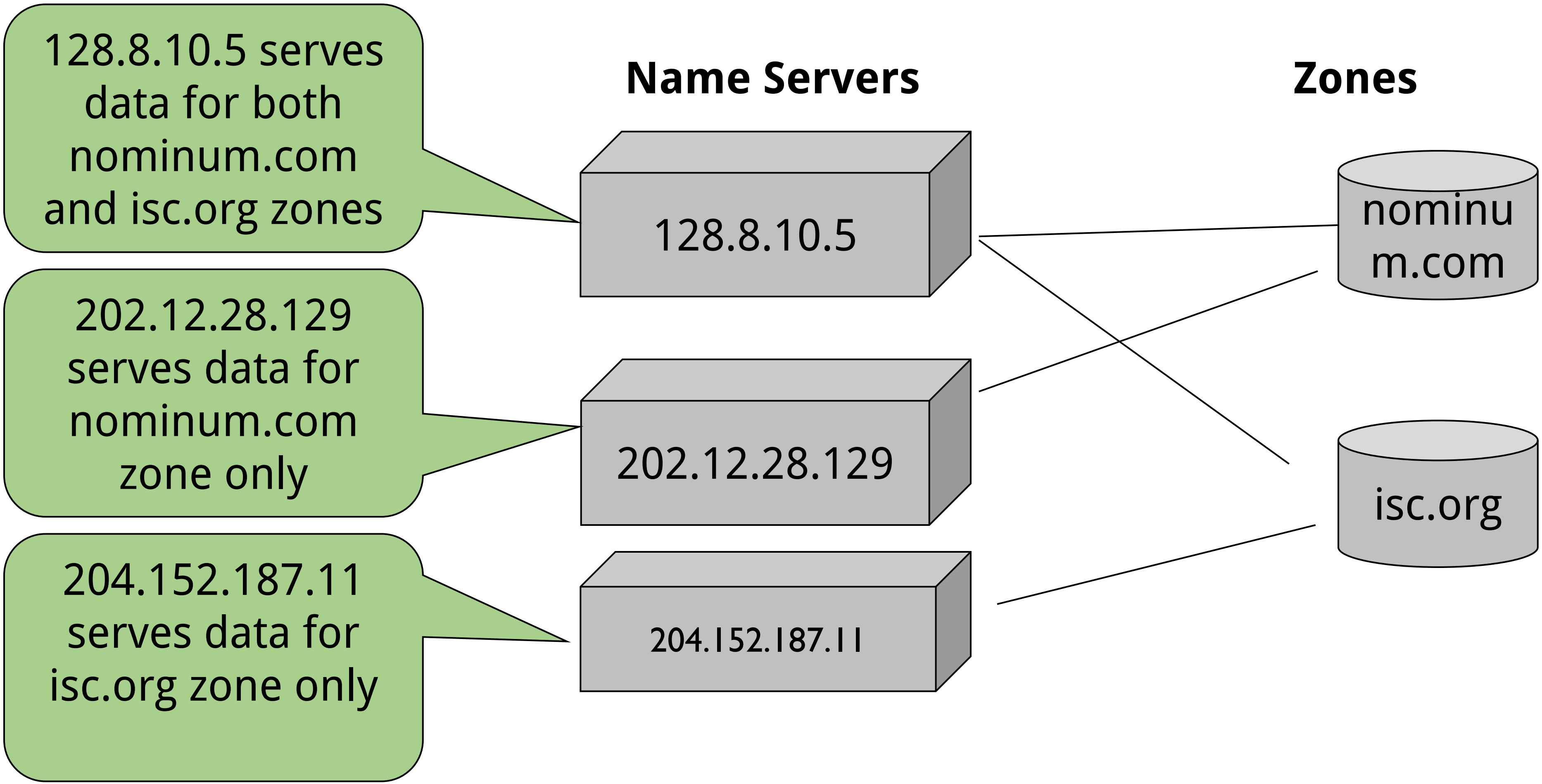
- The subdomain and its parent domain can now be administered independently
- These units are called *zones*

Delegation is good: it is the key to scalability

Name Servers

- Name servers store information about the name space in units called "zones"
  - The name servers that load a complete zone are said to "have authority for" or "be authoritative for" the zone
- Usually, more than one name server are authoritative for the same zone
  - This ensures redundancy and spreads the load
- Also, a single name server may be authoritative for many zones

Name Servers



Type of Name Server

- Two main types of servers
  - Authoritative – maintains the data
    - Master – where the data is edited
    - Slave – where data is replicated to
  - Caching – stores data obtained from an authoritative server

Root Servers

The root zone file lists the names and IP addresses of the authoritative DNS servers for all top-level domains (TLDs)

The root zone file is published on 13 servers, “A” through “M”, around the Internet

Root name server operations currently provided by volunteer efforts by a very diverse set of organizations

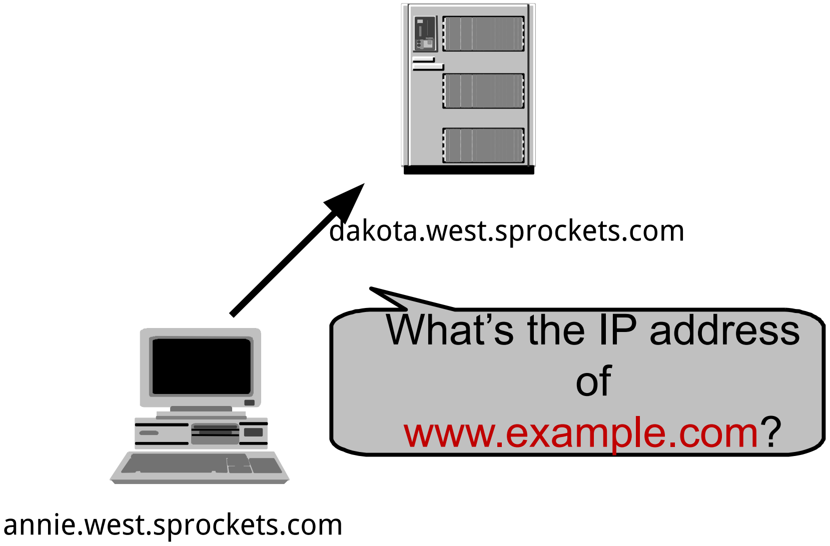
Root Servers

The authoritative name servers that serve the DNS root zone, commonly known as the “root servers”, are a network of hundreds of servers in many countries around the world. They are configured in the DNS root zone as 13 named authorities, as follows.

Hostname	IP Addresses	Manager
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	VeriSign, Inc.
b.root-servers.net	192.228.79.201, 2001:500:84::b	University of Southern California (ISI)
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	VeriSign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

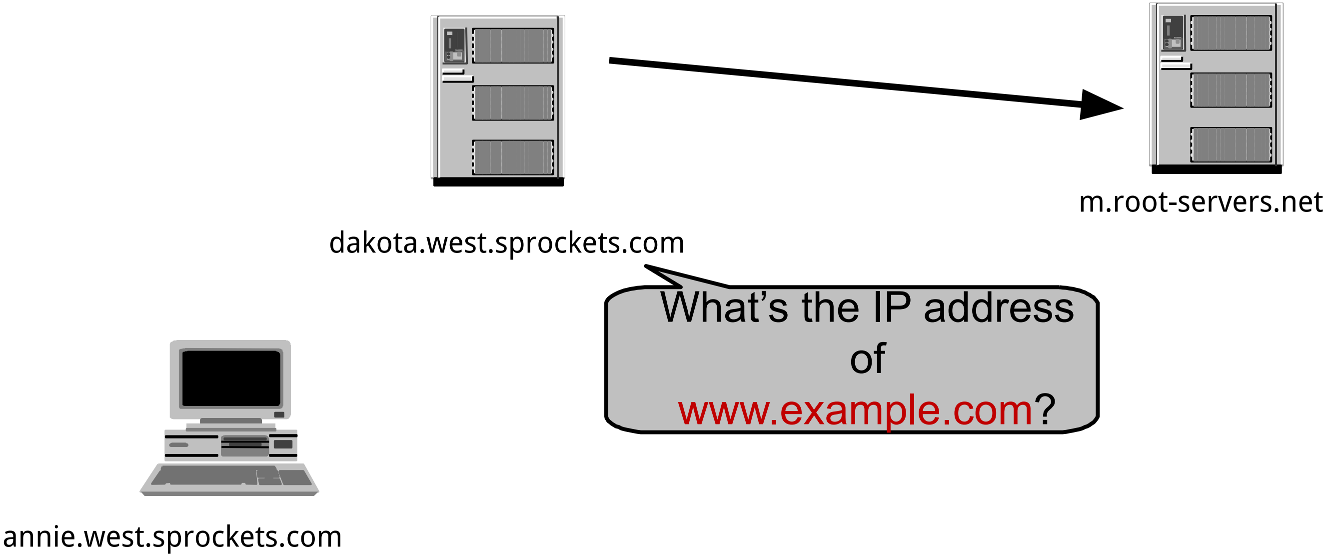
Resolution Process

The workstation *annie* asks its configured name server, *dakota,* for *www.example.com's* address



Resolution Process

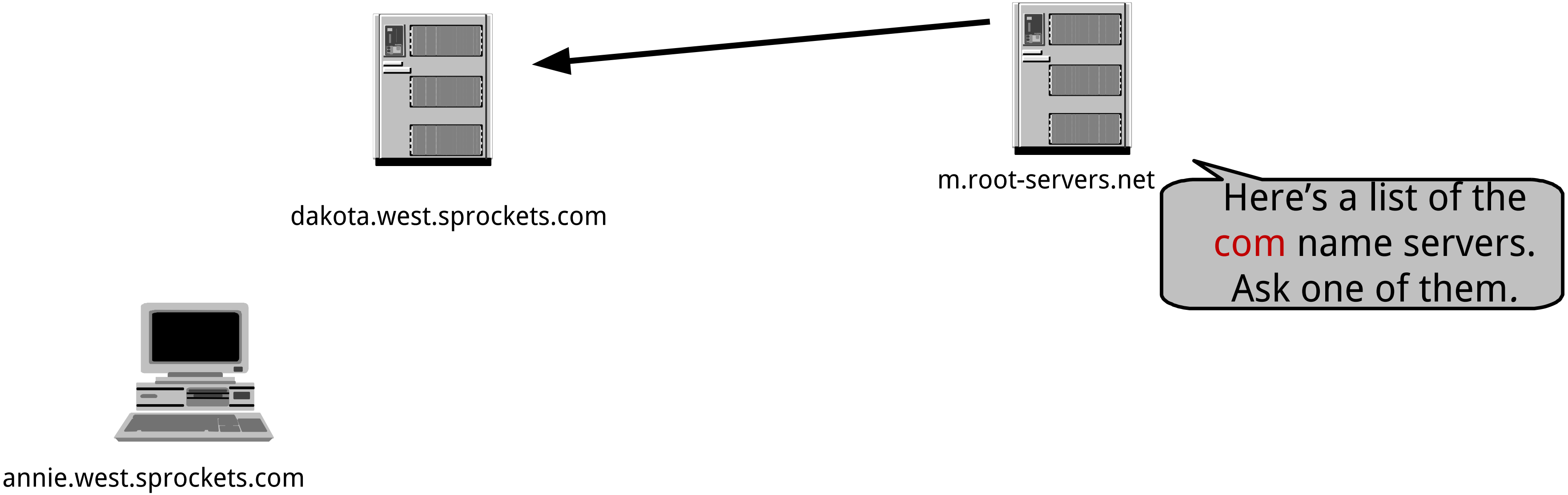
The name server *dakota* asks a root name server, *m*, for *www.example.com's* address



Resolution Process

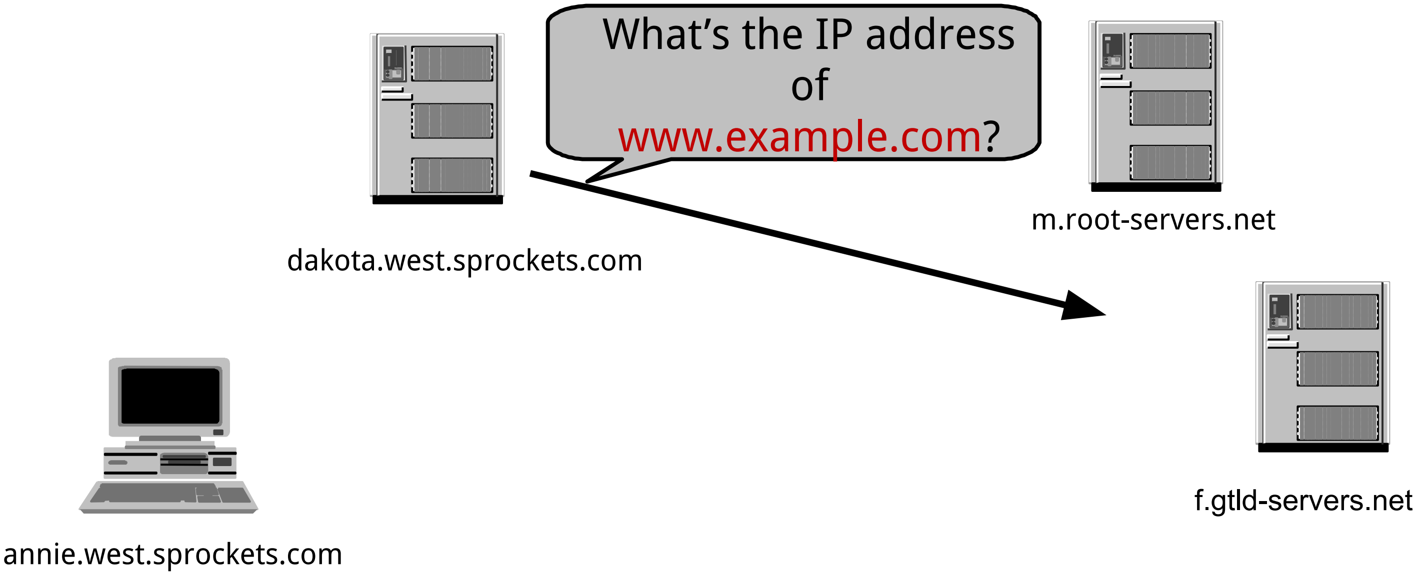
The root server *m* refers *dakota* to the *com* nameservers

This type of response is called a “referral”



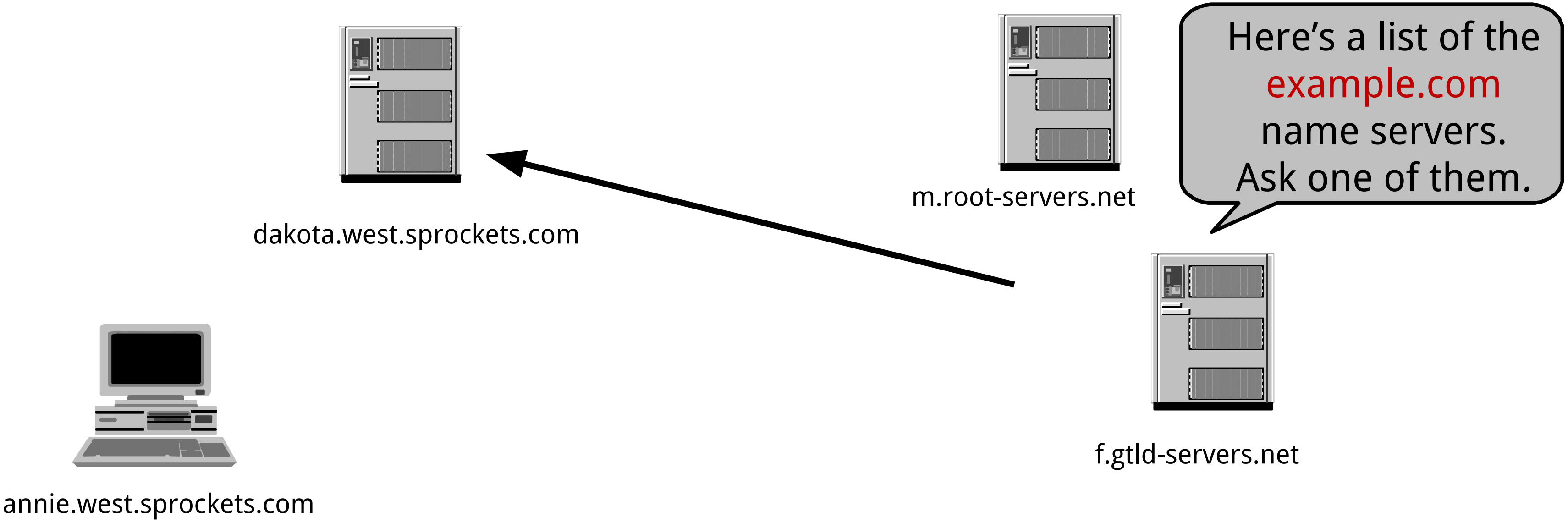
Resolution Process

The name server *dakota* asks a *com* name server, *f*, for *www.example.com's* address



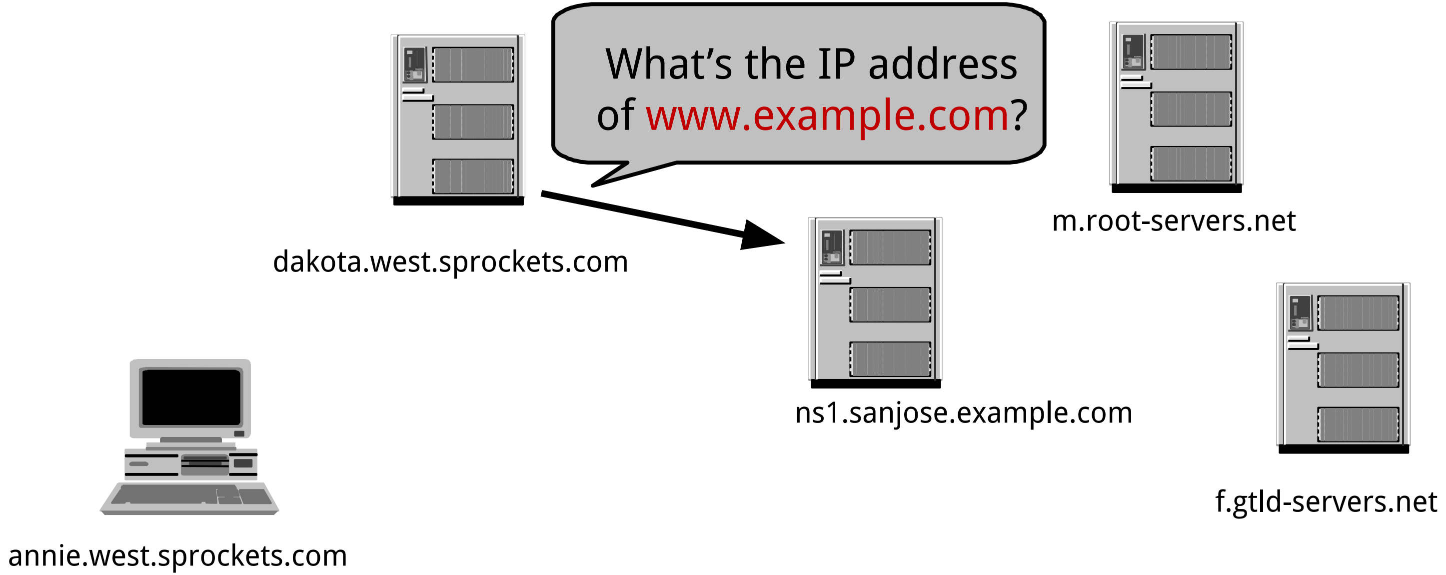
Resolution Process

The *com* name server *f* refers *dakota* to the *example.com* name servers



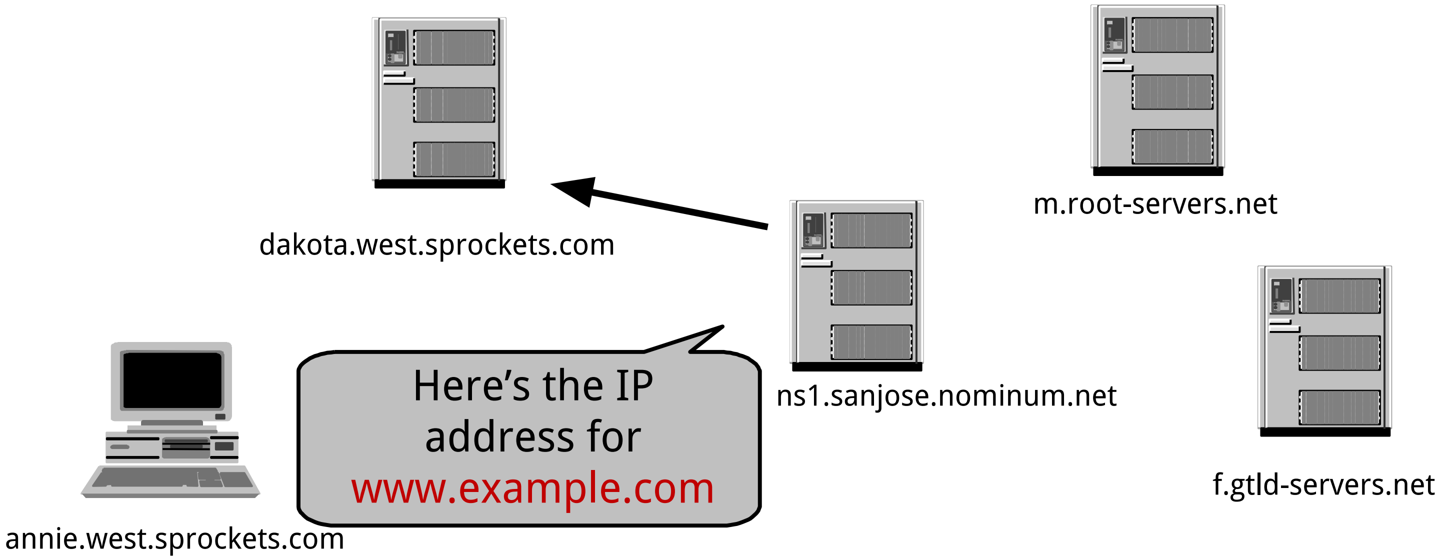
Resolution Process

The name server *dakota* asks a *example.com* name server, *ns1.sanjose*, for *www.example.com's* address



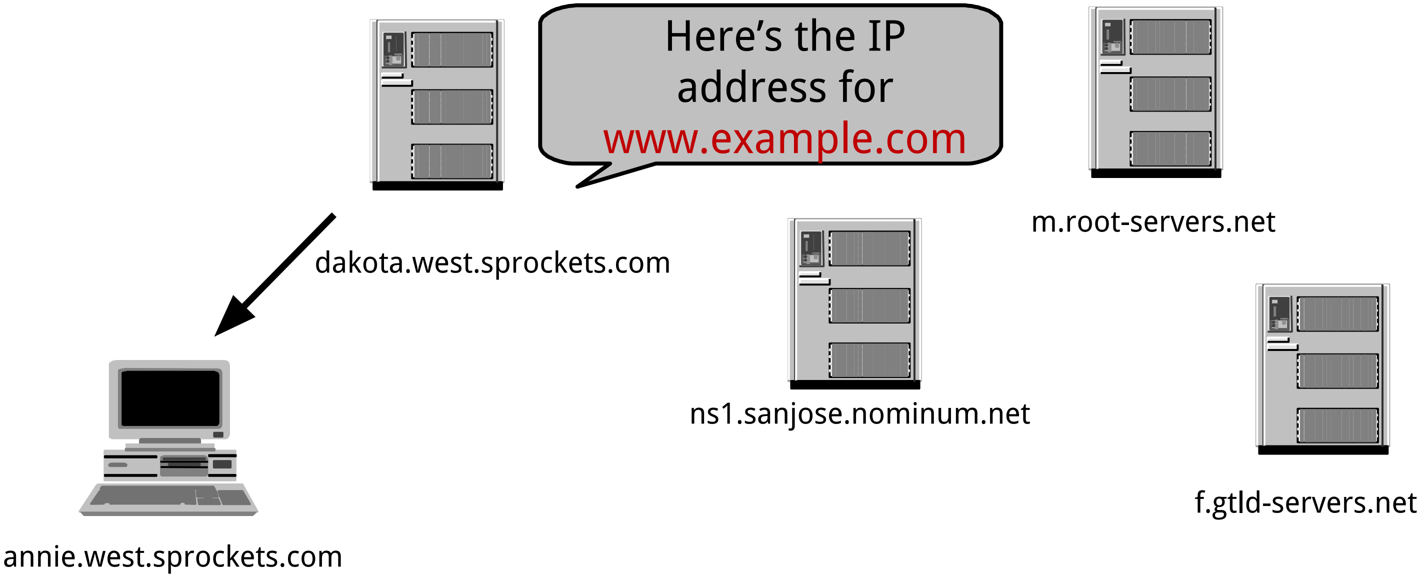
Resolution Process

The *example.com* name server *ns1.sanjose* responds with *www.example.com's* address



Resolution Process

The name server *dakota* responds to *annie* with *www.example.com's* address

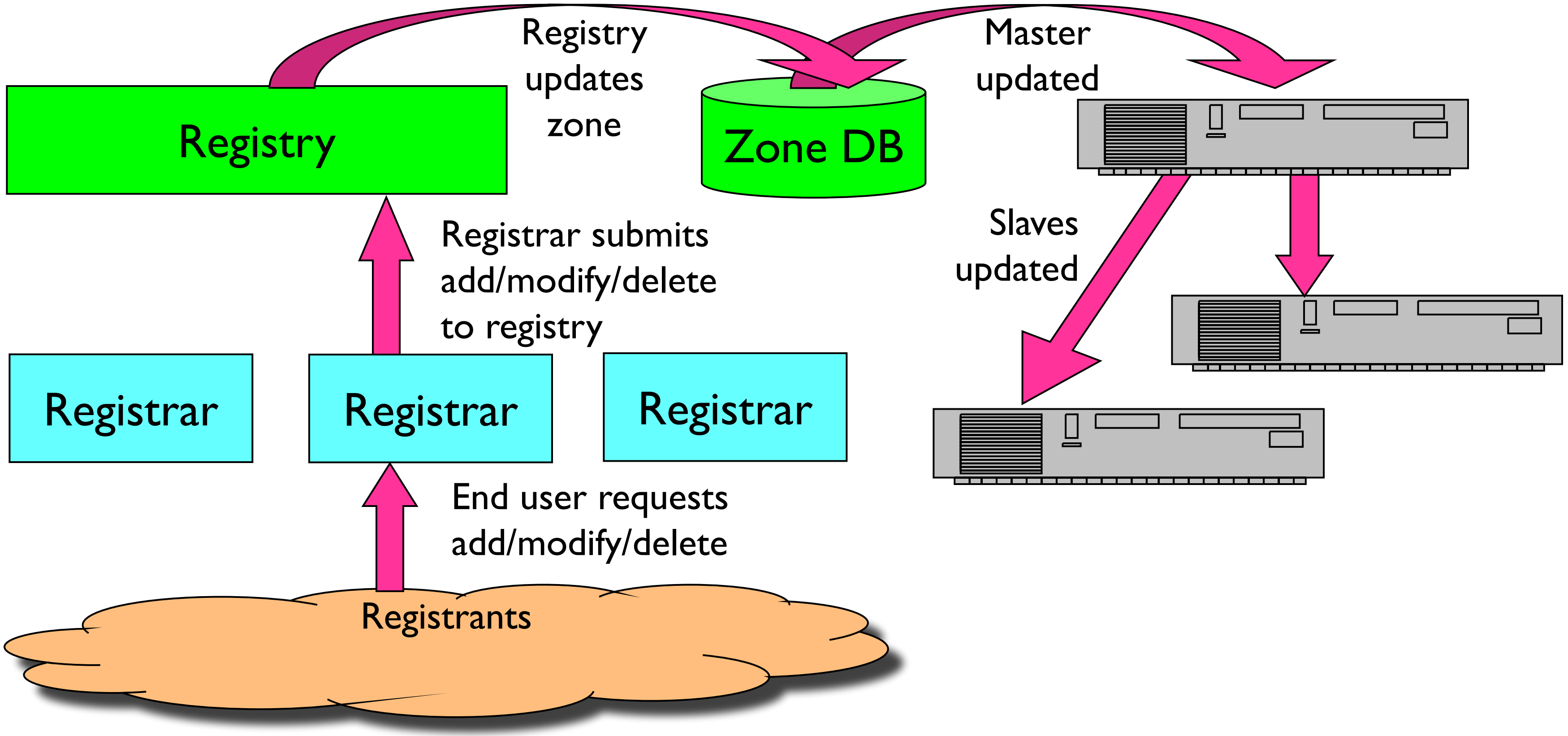


Registries, Registrars, and Registrants

A classification of roles in the operation of a domain name space

- Registry
  - the name space’s database
  - the organization which has edit control of that database
  - the organization which runs the authoritative name servers for that name space
- Registrar
  - the agent which submits change requests to the registry on behalf of the registrant
- Registrant
  - the entity which makes use of the domain name

Registries, Registrars, and Registrants



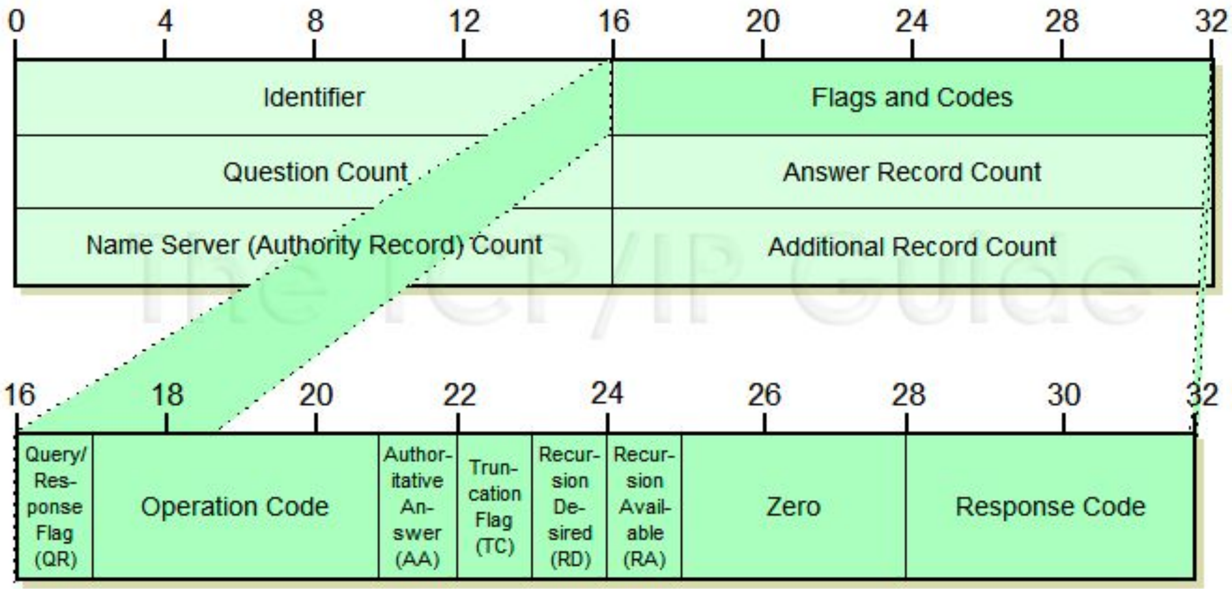
DNS Protocol

DNS primarily uses the UDP on port number 53 to serve requests. DNS queries consist of a single UDP request from the client followed by a single UDP reply from the server.

The DNS protocol uses two types of DNS messages, queries and replies, and they both have the same format.

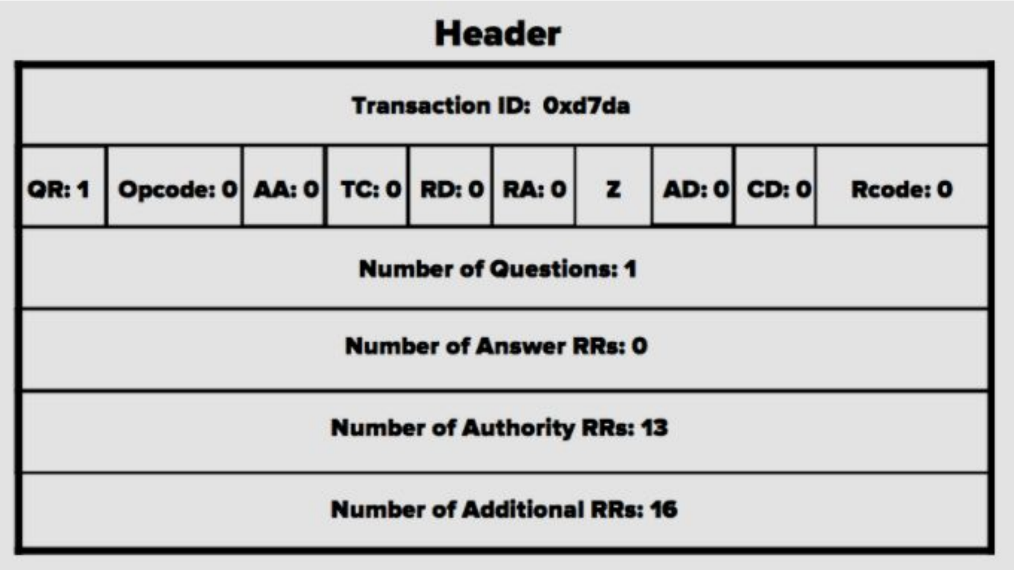
Each message consists of a header and four sections: question, answer, authority, and an additional space. A header field (*flags*) controls the content of these four sections

DNS Protocol



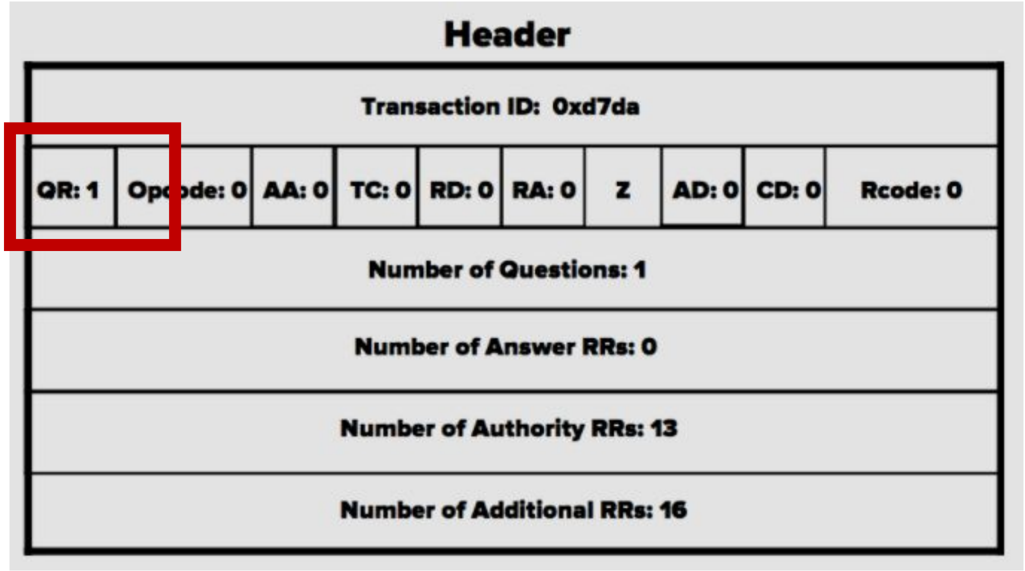
DNS Protocol

- DNS is an application layer protocol.
- DNS protocol relies on UDP by default, but can also work over TCP



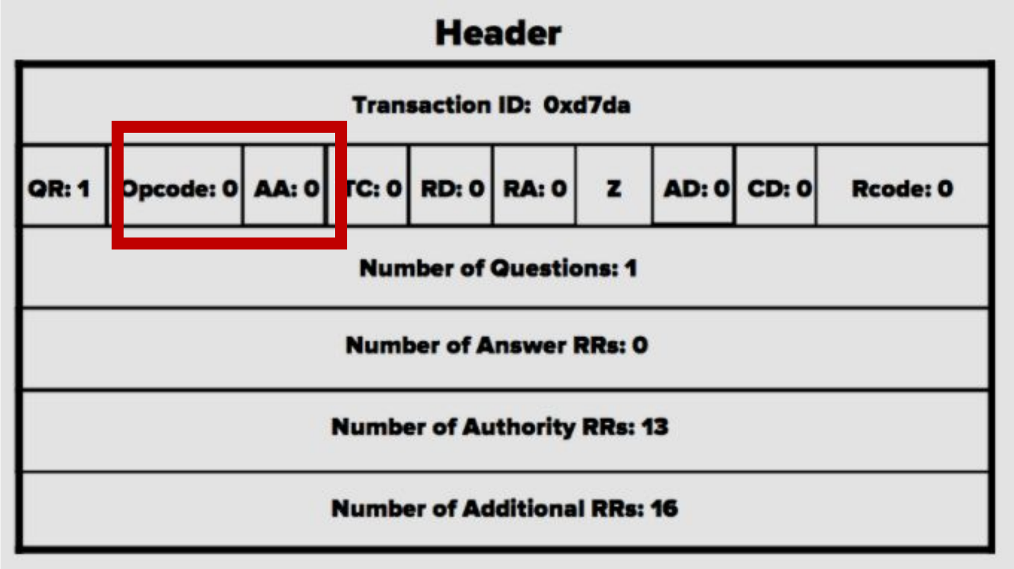
DNS Protocol

Bit 1: QR, query/response flag. When 0, message is a query. When 1, message is response.



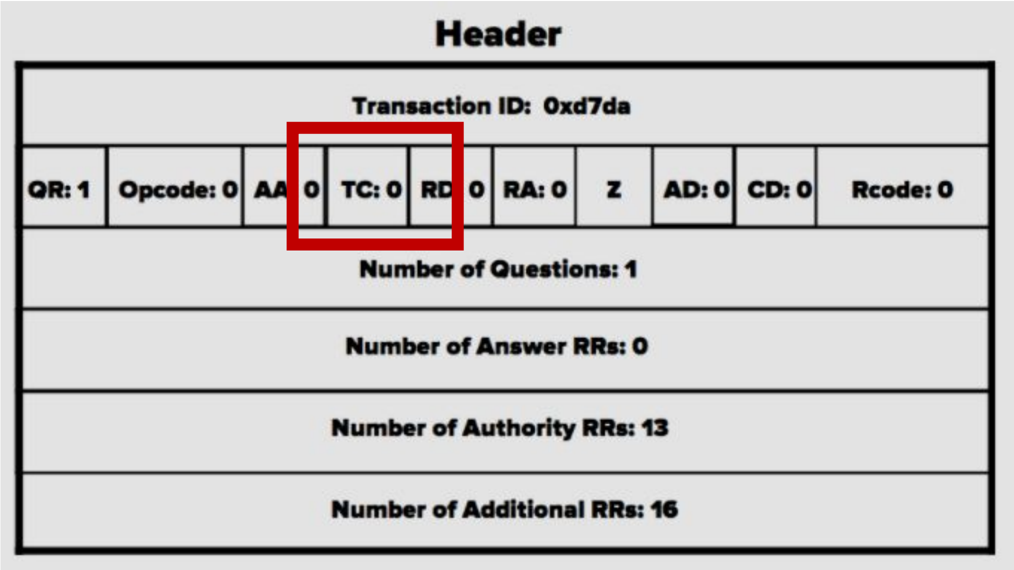
DNS Protocol

Bits 2-5: Opcode, operation code. Tells receiving machine the intent of the message. Generally 0 meaning normal query, however there are other valid options such as 1 for reverse query and 2 for server status.



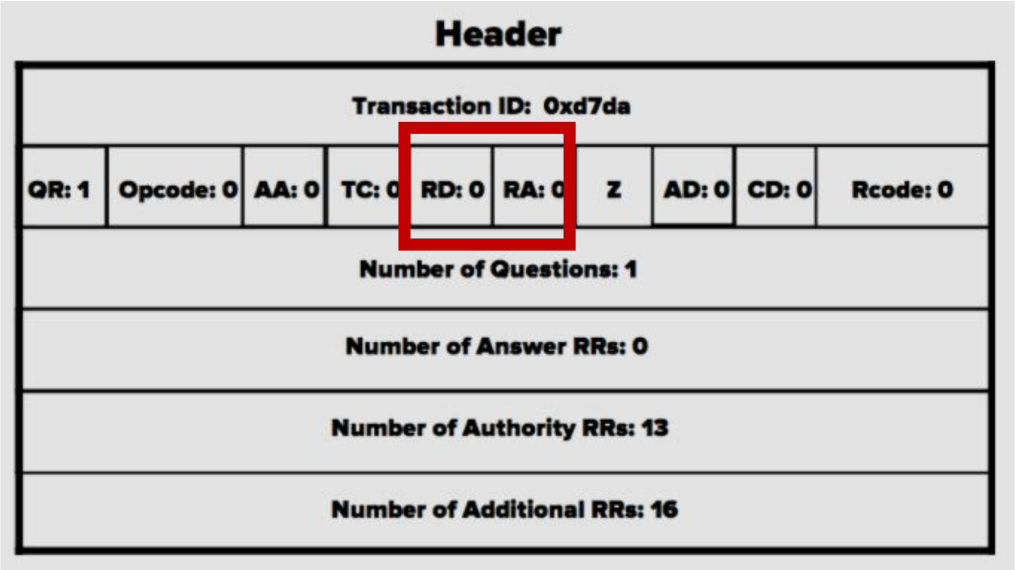
DNS Protocol

Bit 6: AA, authoritative answer. Set only when the responding machine is the authoritative name server of the queried domain.



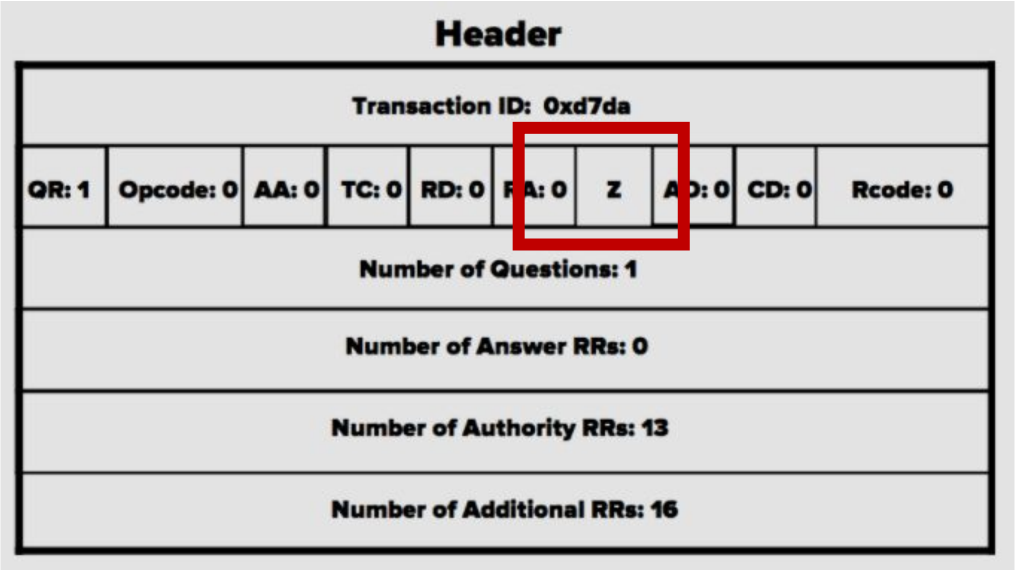
DNS Protocol

Bit 7: TC, truncated. Set if packet is larger than the UDP maximum size of 512 bytes.



DNS Protocol

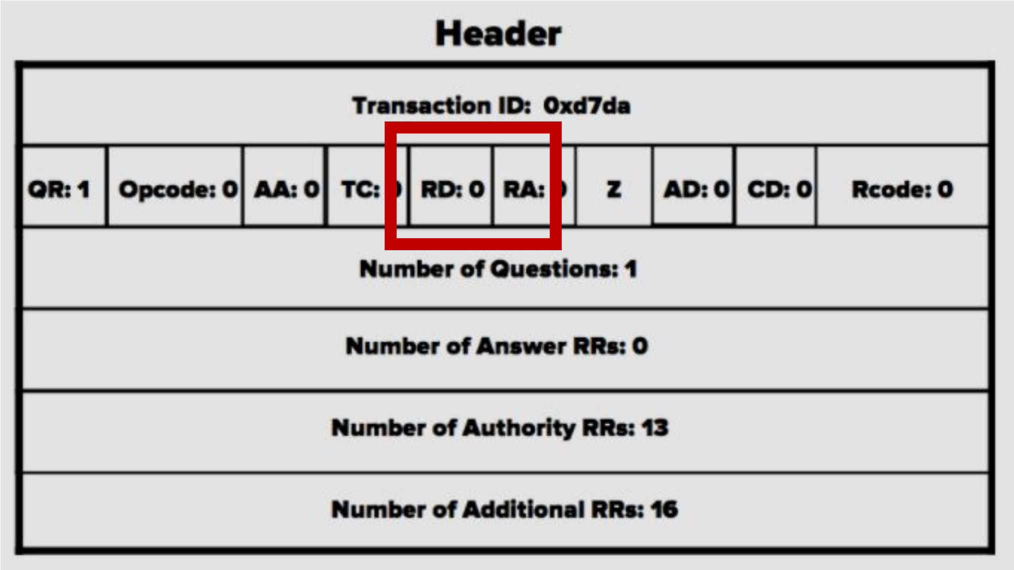
Bit 8: RD, recursion desired. If 0, the query is an iterative query. If 1, the query is recursive.



DNS Protocol

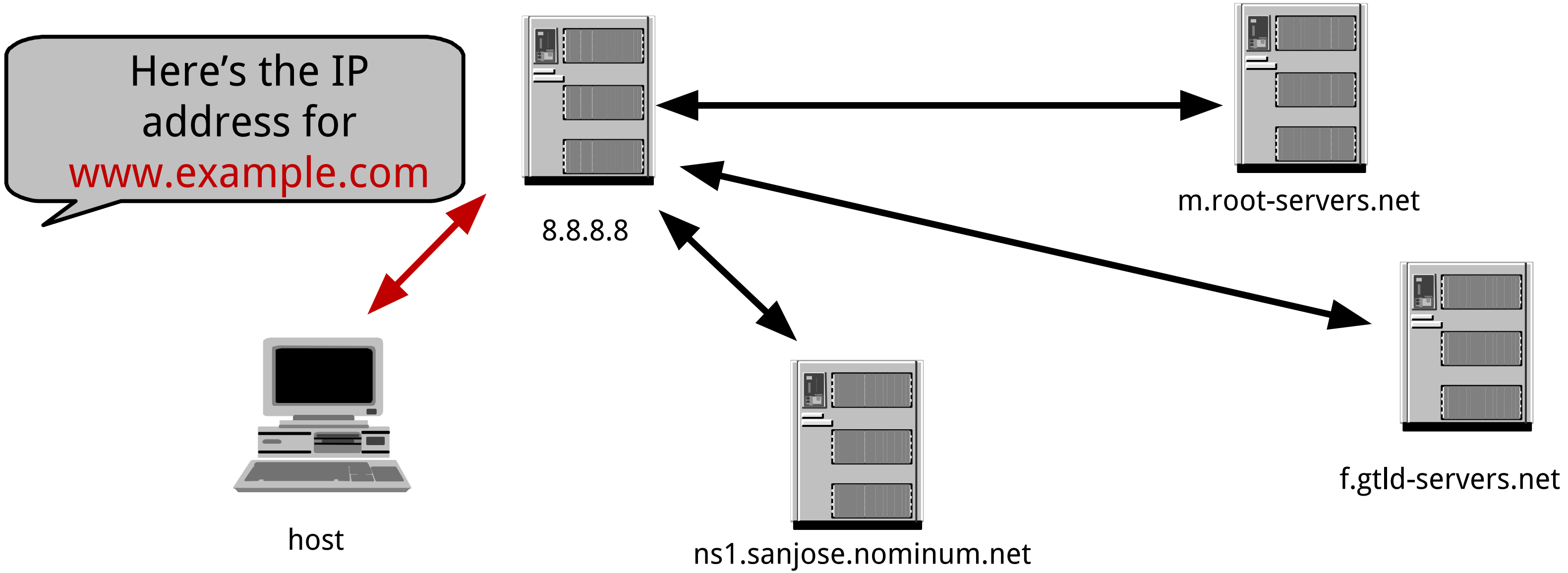
Recursive query: The resolver must complete the recursion and the response must be either an IP address or an error.

Host OS usually sends a recursive query to DNS resolver (8.8.8.8)



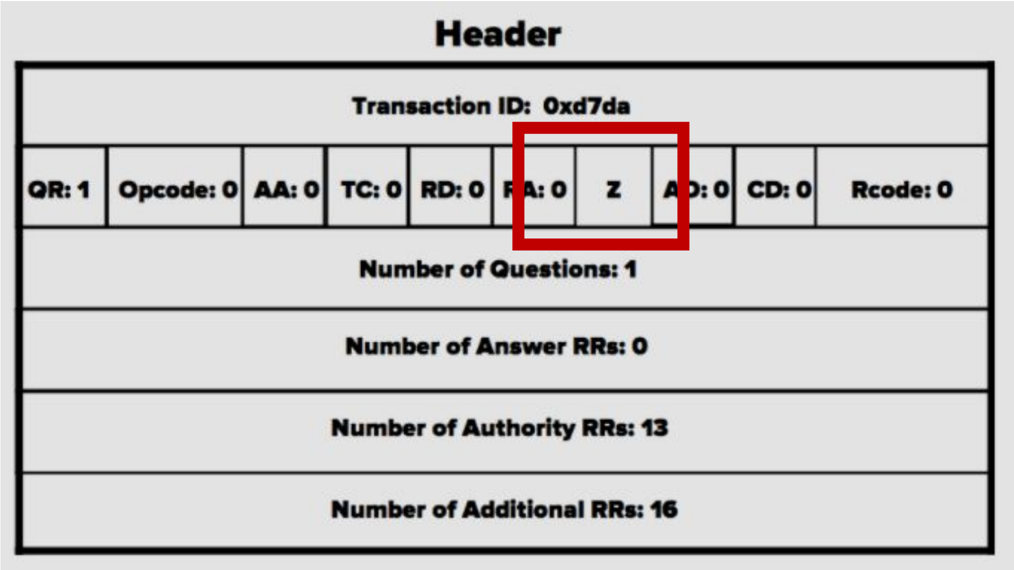
Resolution Process

The name server *8.8.8.8* responds to *host* with *www.example.com's* address



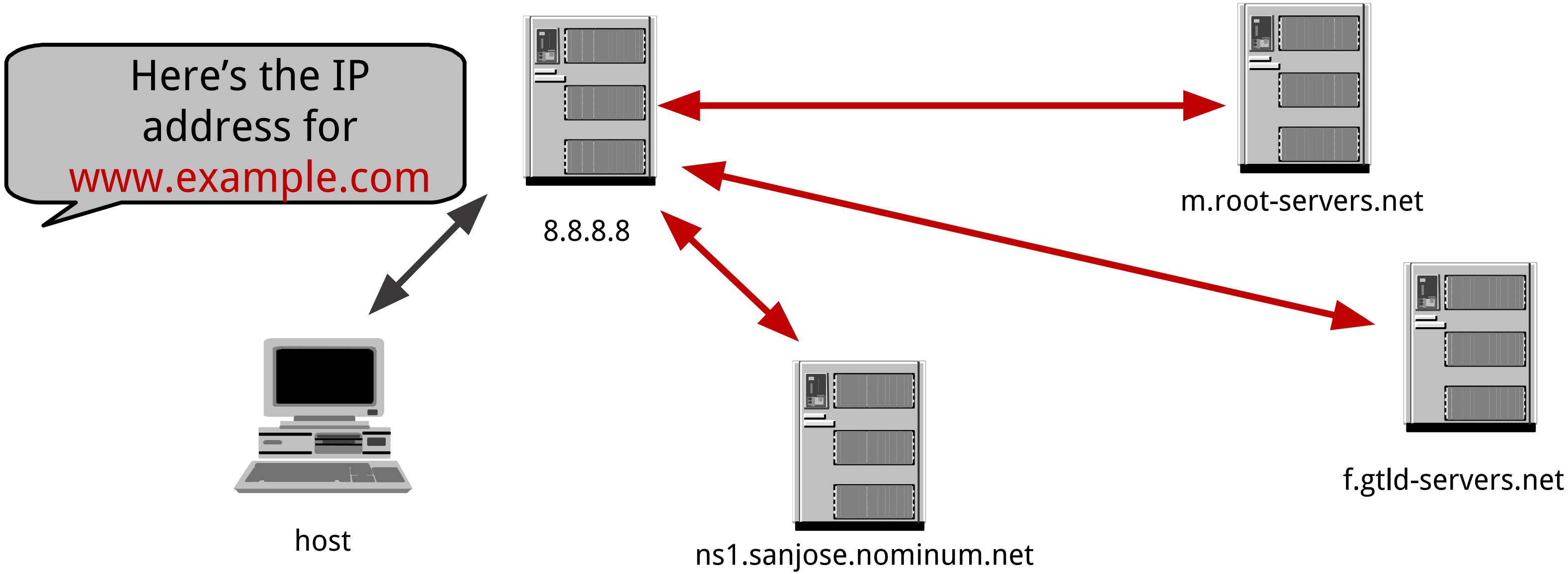
DNS Protocol

Iterative query: meaning the response must be an IP address, the location of an authoritative name server, or an error



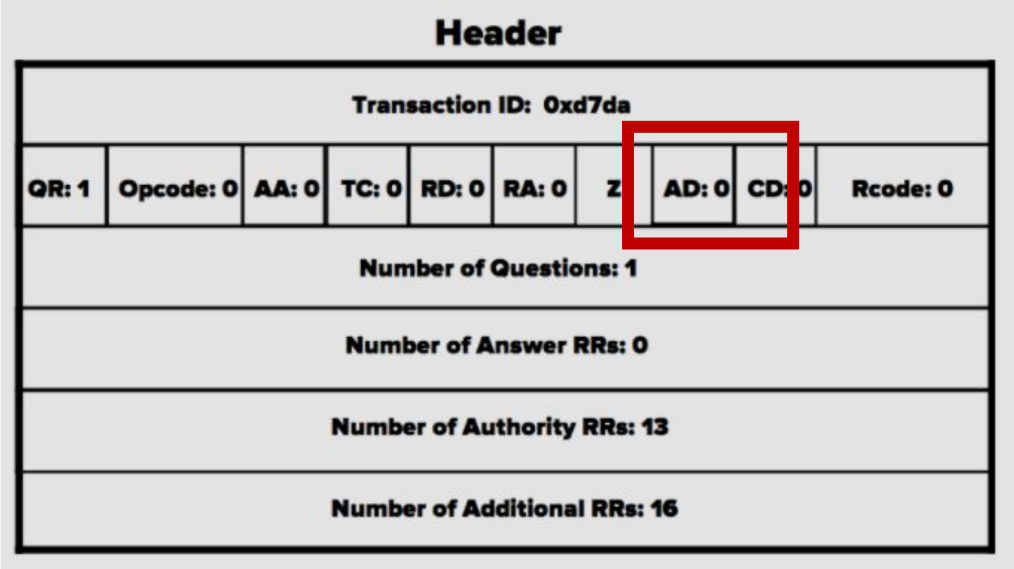
Resolution Process

The name server *8.8.8.8* responds to *host* with
www.example.com's address



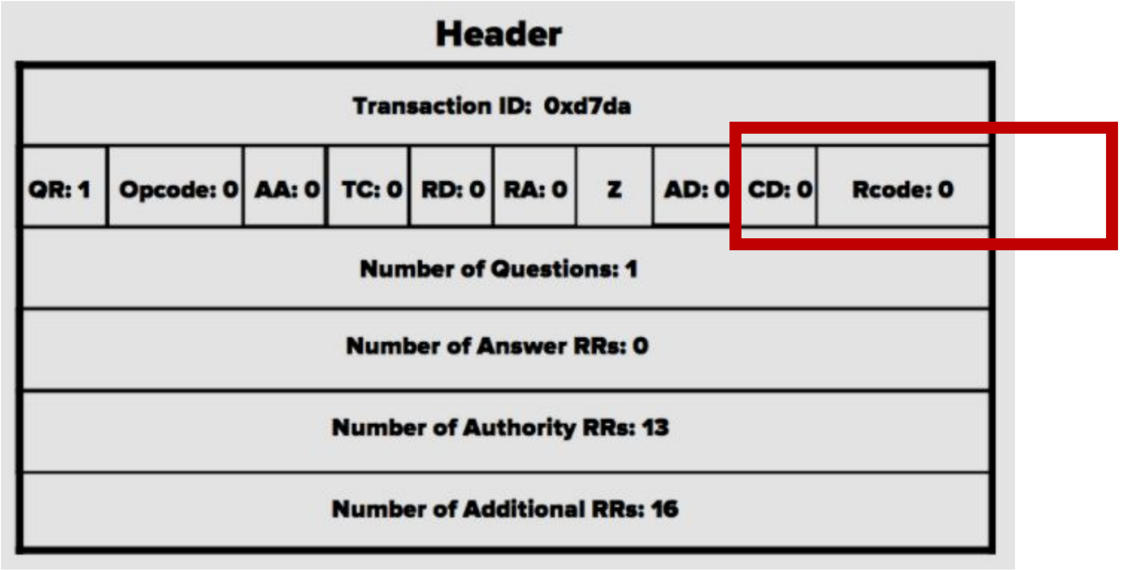
DNS Protocol

Bit 9: RA, recursion available. Set on response if the server supports recursion.



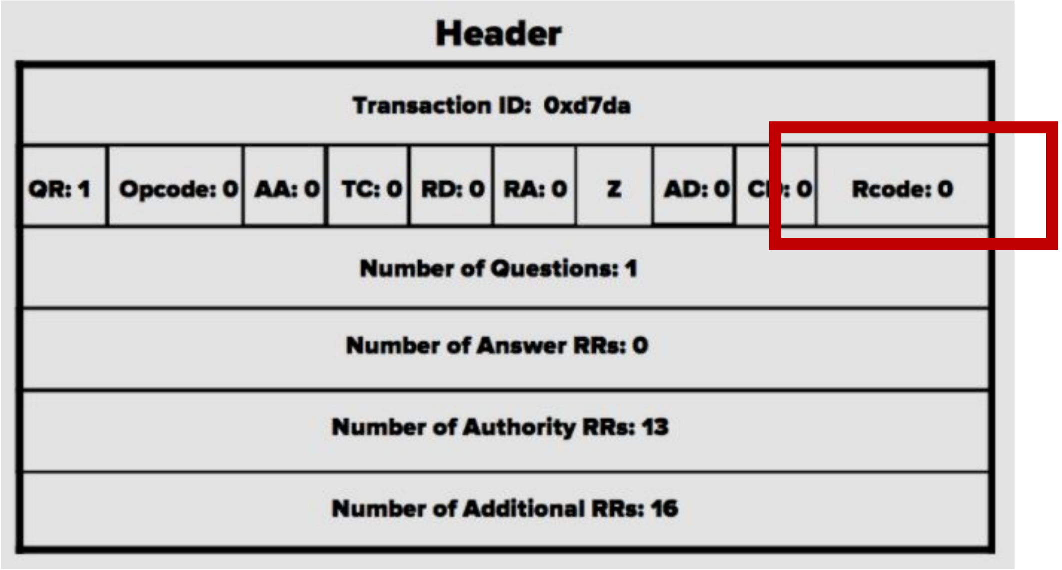
DNS Protocol

Bits 10-12: Reserved in old DNS



DNS Protocol

Bits 13-16: Rcode, return code. 0 for no error, or 3 if the name does not exist.

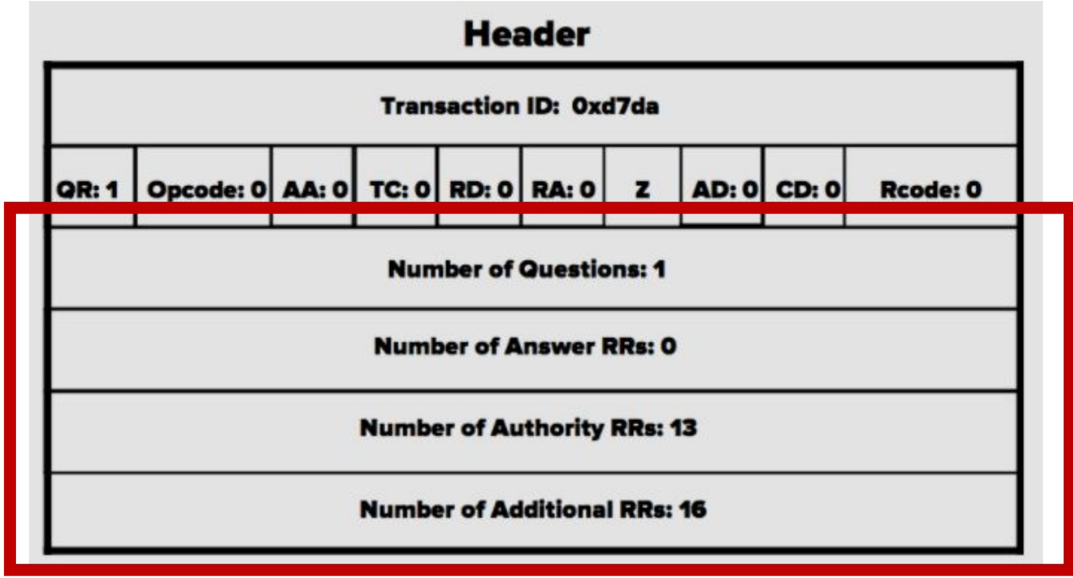


DNS Protocol

The remaining four header fields are

- number of questions,
- answer resource records
- authority resource records
- additional resource records

These numbers vary depending on whether it is a query or response, and what kind of response. In general, however, there will always be at least one question.



DNS Protocol

Question

```
⊟ Queries
  ⊟ www.google.com: type A, class IN
      Name: www.google.com
      Type: A (Host address)
      Class: IN (0x0001)
```

DNS Protocol

Answer Resource Records

```
⊟ Answers
  ⊟ www.google.com: type A, class IN, addr 74.125.131.147
       Name: www.google.com
       Type: A (Host address)
       Class: IN (0x0001)
       Time to live: 5 minutes
       Data length: 4
       Addr: 74.125.131.147 (74.125.131.147)
  ⊞ www.google.com: type A, class IN, addr 74.125.131.103
  ⊞ www.google.com: type A, class IN, addr 74.125.131.104
  ⊞ www.google.com: type A, class IN, addr 74.125.131.106
  ⊞ www.google.com: type A, class IN, addr 74.125.131.99
  ⊞ www.google.com: type A, class IN, addr 74.125.131.105
```

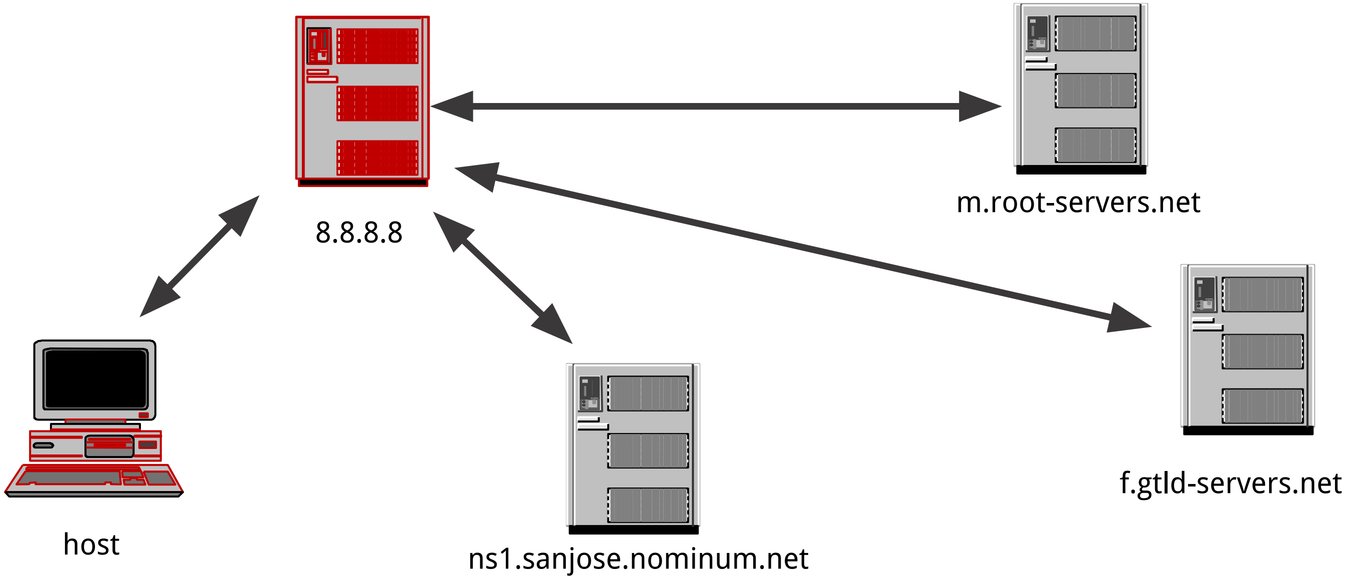
DNS Protocol

Authority Resource Records

```
⊟ Domain Name System (response)
    [Request In: 3]
    [Time: 0.014981000 seconds]
    Transaction ID: 0xccf9
  ⊞ Flags: 0x8000 Standard query response, No error
    Questions: 1
    Answer RRs: 0
    Authority RRs: 4
    Additional RRs: 4
  ⊞ Queries
  ⊟ Authoritative nameservers
    ⊟ google.com: type NS, class IN, ns ns2.google.com
        Name: google.com
        Type: NS (Authoritative name server)
        Class: IN (0x0001)
        Time to live: 2 days
        Data length: 6
        Name Server: ns2.google.com
    ⊞ google.com: type NS, class IN, ns ns1.google.com
    ⊞ google.com: type NS, class IN, ns ns3.google.com
    ⊞ google.com: type NS, class IN, ns ns4.google.com
  ⊞ Additional records
```

Cache

The name server *8.8.8.8* responds to *host* with *www.example.com's* address



DNS Protocol

Time to live dictates how long it will be in seconds until the receiver refreshes its DNS related information (cache)

```
⊟ Domain Name System (response)
    [Request In: 3]
    [Time: 0.014981000 seconds]
    Transaction ID: 0xccf9
  ⊞ Flags: 0x8000 Standard query response, No error
    Questions: 1
    Answer RRs: 0
    Authority RRs: 4
    Additional RRs: 4
  ⊞ Queries
  ⊟ Authoritative nameservers
    ⊟ google.com: type NS, class IN, ns ns2.google.com
        Name: google.com
        Type: NS (Authoritative name server)
        Class: IN (0x0001)
        Time to live: 2 days
        Data length: 6
        Name Server: ns2.google.com
    ⊞ google.com: type NS, class IN, ns ns1.google.com
    ⊞ google.com: type NS, class IN, ns ns3.google.com
    ⊞ google.com: type NS, class IN, ns ns4.google.com
  ⊞ Additional records
```

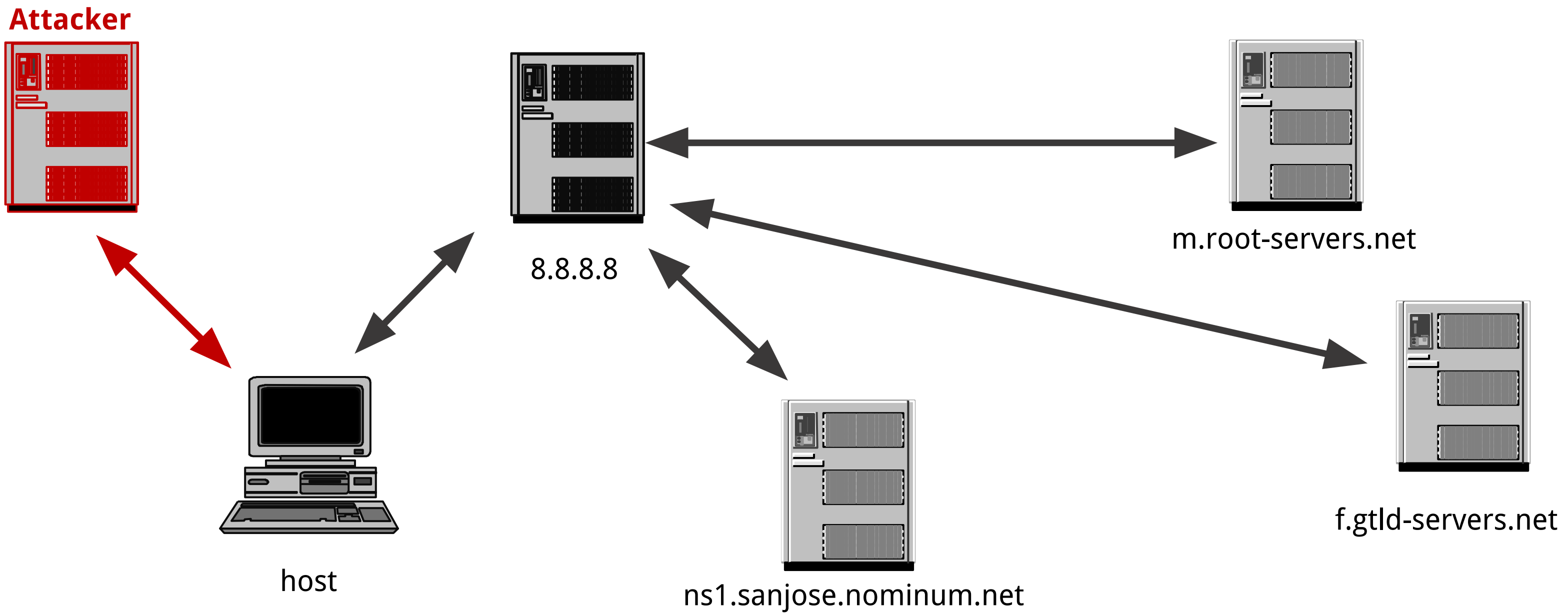
DNS Cache Poisoning

DNS Spoofing

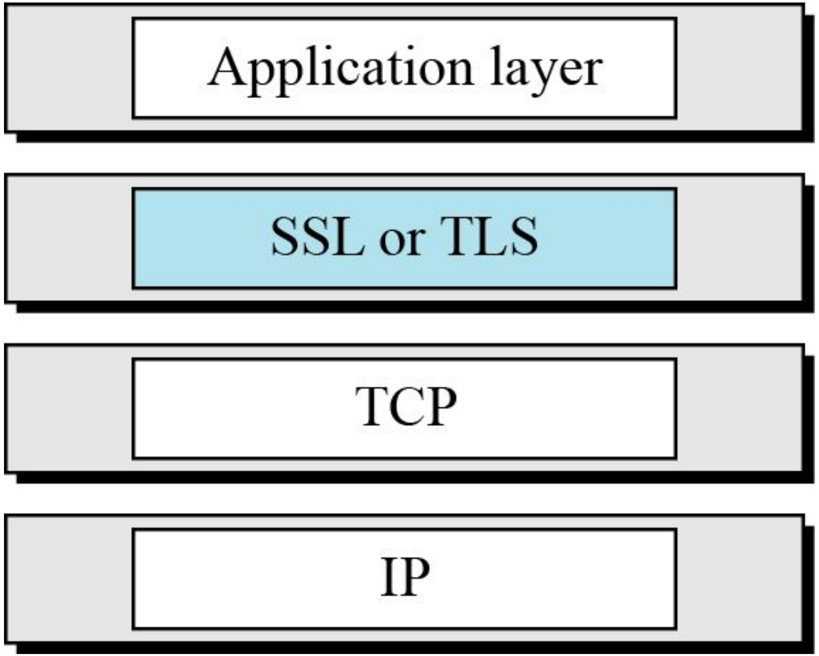
Corrupt Domain Name System data is introduced into a DNS resolver's cache, causing the name server to return an incorrect IP address, which results in diverting traffic to a wrong computer.

DNS Cache Poisoning

Attacker sends out spoofed DNS response



Defense?



DNSSEC

Domain Name System Security Extensions (DNSSEC) is a suite of extensions that add security to the Domain Name System (DNS) protocol by enabling DNS responses to be validated.

Specifically, DNSSEC provides origin authority, data integrity, and authenticated denial of existence. With DNSSEC, the DNS protocol is much less susceptible to certain types of attacks, particularly DNS spoofing attacks.

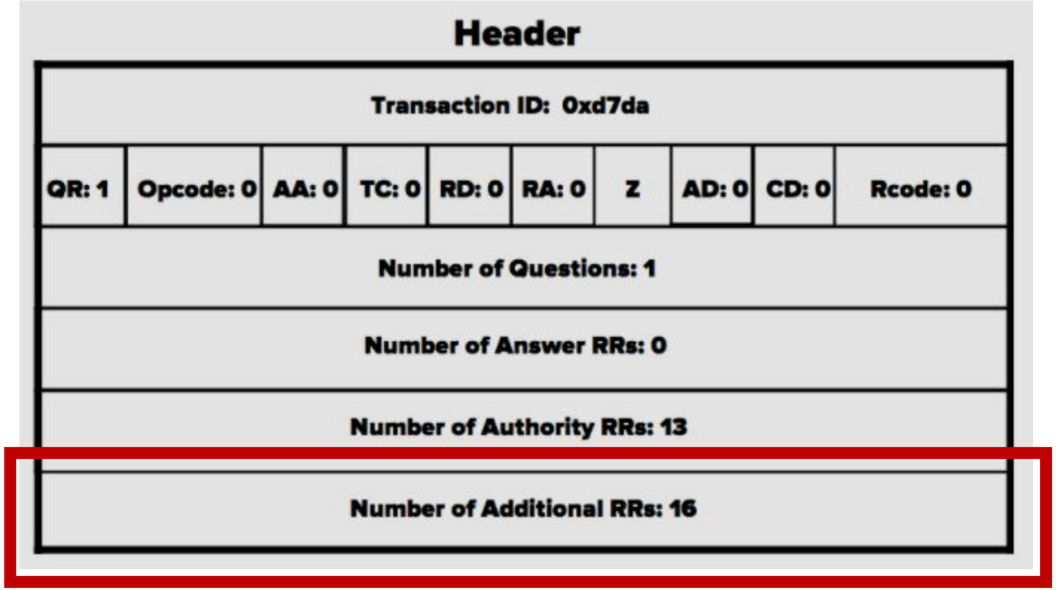
DNSSEC Mechanism

- Using public key cryptographic algorithms signatures are applied over the DNS data

- By comparing the signatures with public keys the integrity and authenticity of the data can be established.

DNSSEC Mechanism

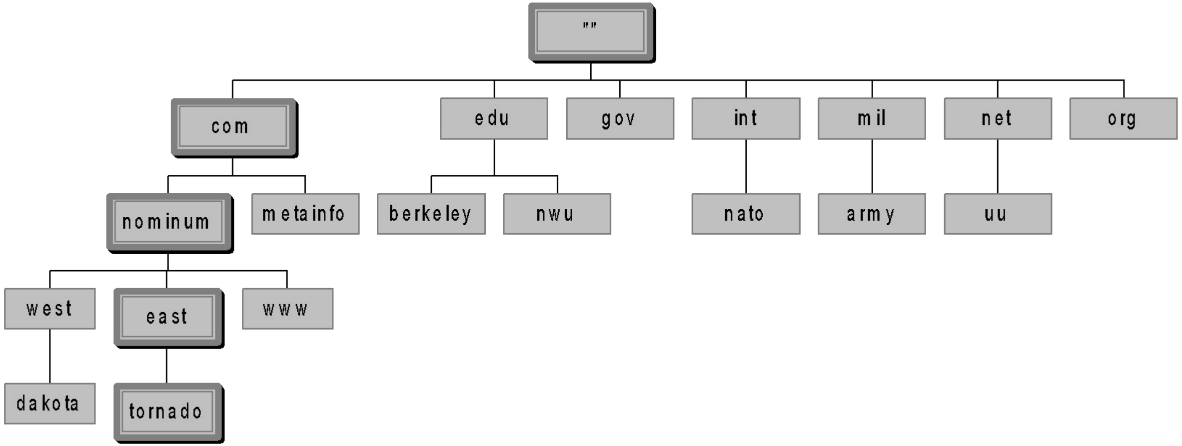
- Private Key: kept private and stored locally
- Public Keys: Published in the DNS as a DNSKEY Resource Record
- Signatures: Published in the DNS as a RRSIG Resource Record



Validate Public Keys

Distributing keys through DNS:

- Use one trusted key to establish authenticity of other keys
- Building chains of trust from the root down
- Parents need to sign the keys of their children



Validate Public Keys

Distributing keys through DNS:

- Use one trusted key to establish authenticity of other keys
- Building chains of trust from the root down
- Parents need to sign the keys of their children