

# CompSci 356: Computer Network Architectures

## Lecture 25: Domain Name System (DNS) and Content distribution networks

### Chapter 9.3.1

Xiaowei Yang

[xwy@cs.duke.edu](mailto:xwy@cs.duke.edu)

# Overview

- Domain Name System
- Content Distribution Networks
- DNS attacks

# Domain Name System (DNS)

## Outline

- Functions of DNS
- Design goals of DNS
- History of DNS
- DNS architecture: hierarchy is the key
  - Name space and resource records
  - Name servers
  - Name resolvers

<http://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/ref-guide/s1-bind-zone.html>

## Functions of DNS

- Map an easy-to-remember name to an IP address
  - Without DNS, to send an IP packet, we'd have to remember
    - 66.102.7.99
    - 64.236.24.28
  - With DNS
    - **www.google.com** → 66.102.7.99
    - **www.cnn.com** → 64.236.24.28
- DNS also provides inverse look up that maps an IP address to an easy-to-remember name

## Design goals of DNS

- The primary goal is a consistent name space which will be used for referring to resources.
  - Consistent: same names should refer to same resources
  - Resources: IP addresses, mail servers
- Enable Distributed management
  - The size of the name database will be large
  - The updates will be frequent
- Design goals determine its structure
  - A hierarchical name space
  - A distributed directory service

## Before there was DNS ....

.... there was the HOSTS.TXT file maintained on a host at SRI Network Information Center (NIC)

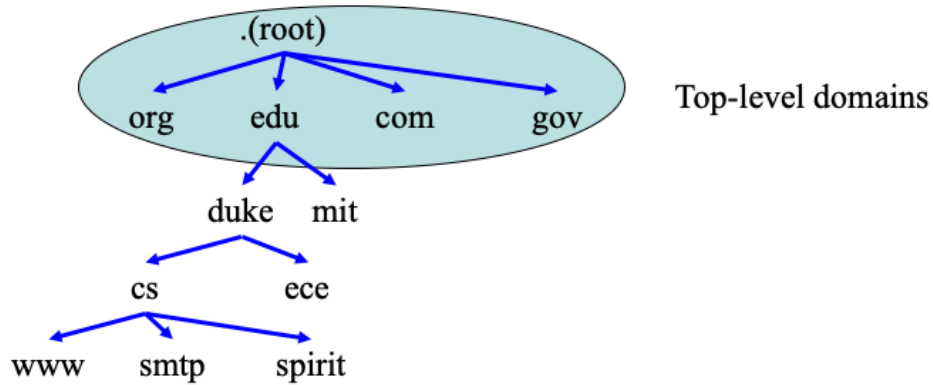
- Before DNS (until 1985), the name-to-IP address was done by downloading a single file (hosts.txt) from a central server with FTP
  - Names in hosts.txt are not structured
  - The hosts.txt file still works on most operating systems. It can be used to define local names

## Key components in DNS Architecture

- Domain name space and resource records (RRs)
- Name servers
- Name resolution

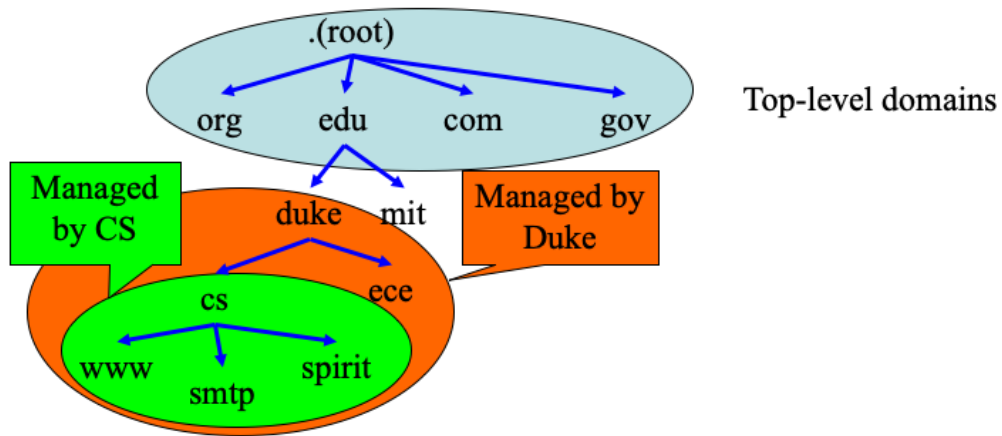


# Domain Namespace



- Domain namespace is a hierarchical and logical tree structure
- The label from a node to root in the DNS tree represents a **DNS name**
- Each subtree below a node is a **DNS domain**.
  - DNS domain can contain hosts or other domains (**subdomains**)
- Examples of DNS domains: .edu, duke.edu, cs.duke.edu

# Distributed Management

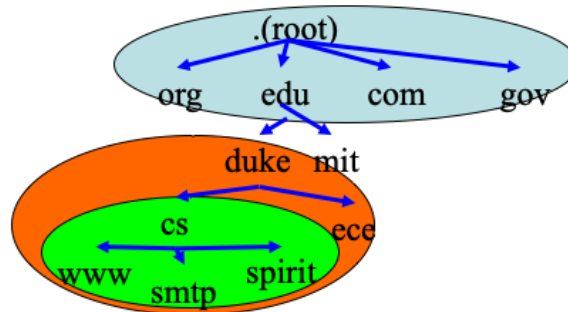


- Below top-level domain, administration of name space is delegated to organizations
- Each organization can delegate further

## Domain names

- Names of hosts can be assigned independent of host locations on a link layer network, IP network or autonomous system
  - My computer's DNS name *xiaowei.net* need not change even if my computer's IP address has changed

# Fully Qualified Domain Names



- Every node in the DNS domain tree can be identified by a unique Fully Qualified Domain Name (FQDN)
- A FQDN (from right to left) consists of labels ("cs", "duke", "edu") separated by a period (".") from root to the node
- Each label can be up to 63 characters long. The total number of characters of a DNS name is limited to 255.
- FQDN contains characters, numerals, and dash character ("-")
- FQDNs are not case-sensitive

# Top-level domains

- Three types of top-level domains:
  - **Generic Top Level Domains (gTLD)**: 3-character code indicates the function of the organization
    - Used primarily within the US
    - Examples: gov, mil, edu, org, com, net
  - **Country Code Top Level Domain (ccTLD)**: 2-character country or region code
    - Examples: us, va, jp, de
  - **Infrastructure top level domains**: A special domain (in-addr.arpa) used for IP address-to-name mapping

There are more than 1000+ top-level domains.

## Who “owns” DNS?

- The Internet needs governance
  - IP addresses, AS numbers, DNS, and other Internet names/numbers
  - Internet Assigned Numbers Authority (IANA) has the authority to manage the numbers
- Who implements IANA?
  - Originally by Jon Postel till 1998
  - By Internet Corporation of Assigned Names and Numbers (ICANN) formed in 1998
    - Used to be under the oversight of US government
    - By Oct 1, 2016, free of it

## Generic Top Level Domains (gTLD)

- Sponsored top level domains
  - Has a sponsor representing the community
  - Sponsor in charge of policies
  - .aero sponsored by the company SITA
- Unsponsored top level domains
  - ICANN
  - .com, .net, .info

The sponsor of the TLD is responsible to develop of policies, ensure transparency and accountability in its operations, and maintain the best interest of the sponsored internet community.

# Sponsored top level domains


<a href="#">.aero</a>	Members of the air-transport industry	<a href="#">SITA</a>
<a href="#">.asia</a>	Companies, organisations and individuals in the Asia-Pacific region	DotAsia Organisation
<a href="#">.cat</a>	Catalan linguistic and cultural community	<a href="#">Fundació puntCat</a>
<a href="#">.coop</a>	<a href="#">Cooperative associations</a>	<a href="#">DotCooperation LLC</a>
<a href="#">.edu</a>	Post-secondary institutions accredited by an agency recognized by the U.S. Department of Education	<a href="#">EDUCAUSE</a>
<a href="#">.gov</a>	United States Government	<a href="#">General Services Administration</a>
<a href="#">.int</a>	Organizations established by international treaties between governments	<a href="#">IANA</a>
<a href="#">.jobs</a>	Human resource managers	<a href="#">Society for Human Resource Management</a>
<a href="#">.mil</a>	United States Military	DoD Network Information Center
<a href="#">.mobi</a>	Providers and consumers of mobile products and services	<a href="#">dotMobi</a>
<a href="#">.museum</a>	Museums	<a href="#">Museum Domain Management Association</a>
<a href="#">.post</a>	Postal services	<a href="#">Universal Postal Union</a>
<a href="#">.tel</a>	For businesses and individuals to publish contact data	Telnic Ltd.
<a href="#">.travel</a>	Travel agents, airlines, hoteliers, tourism bureaus, etc.	Tralliance Corporation
<a href="#">.xxx</a>	Pornographic sites	<a href="#">ICM Registry</a>



## Un-sponsored top-level domains

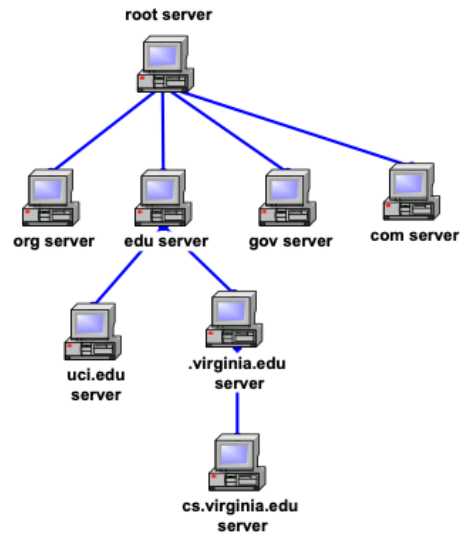
- .com
- .org
- .net
- .biz
- .info
- .name

# DNS (technical) architecture

- Domain name space
  - A hierarchical tree structure
  - A domain can be delegated to an organization
- Resource records
  - Records domain name related information
- Name servers 
  - Domain name hierarchy exists only in the abstract
  - Name servers implement the hierarchy
  - Maintains RRs
  - A host's name servers are specified in `/etc/resolv.conf`
- Name resolution

# Hierarchy of name servers

- The resolution of the hierarchical name space is done by a hierarchy of name servers
- Namespace is partitioned into zones. A zone is a contiguous portion of the DNS name space
- Each server is responsible (authoritative) for a zone.
- DNS server answers queries about host names in its zone

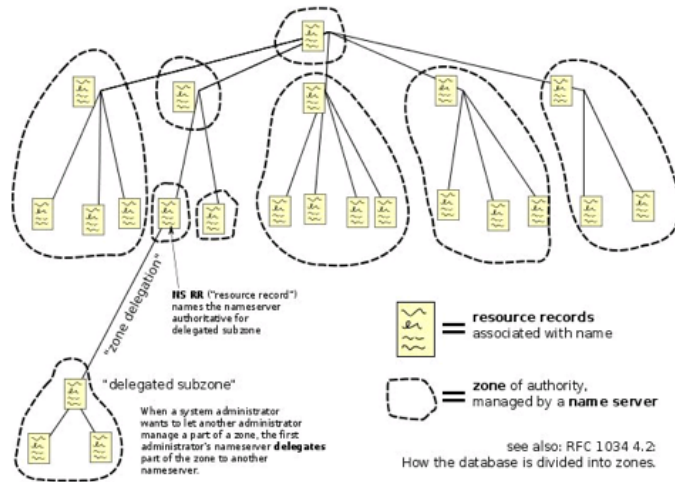


## DNS domain and zones

- Each zone is anchored at a specific domain node, but zones are not domains.
- *A DNS domain* is a subtree of the namespace
- A zone is a portion of the DNS namespace generally stored in a file (It could consists of multiple nodes)
- A server can divide part of its zone and **delegate** it to other servers
- A name server implements the zone information as a collection of resource records

# Zone and sub-domain

## Domain Name Space



# Primary and secondary name servers

- For each zone, there must be a primary name server and a secondary name server for reliability reason
  - The **primary server** (**master server**) maintains a **zone file** which has information about the zone. Updates are made to the primary server
  - The **secondary server** copies data stored at the primary server

## **Adding a host:**

- When a new host is added (“spirit.cs.duke.edu”) to a zone, the administrator adds the IP information on the host (IP address and name) to a configuration file on the primary server

# Root name servers

## Map of the Root Servers



- The root name servers know how to find the authoritative name servers for all top-level zones.
- There are 13 (virtual) root name servers
- Root servers are critical for the proper functioning of name resolution

## Addresses of root servers

A.ROOT-SERVERS.NET.	(VeriSign, Dulles, VA)	198.41.0.4
B.ROOT-SERVERS.NET.	(ISI, Marina Del Rey CA)	192.228.79.201
C.ROOT-SERVERS.NET.	(Cogent Communications)	192.33.4.12
D.ROOT-SERVERS.NET.	(University of Maryland)	128.8.10.90
E.ROOT-SERVERS.NET.	(Nasa Ames Research Center)	192.203.230.10
F.ROOT-SERVERS.NET.	(Internet Systems Consortium)	192.5.5.241
G.ROOT-SERVERS.NET.	(US Department of Defense)	192.112.36.4
H.ROOT-SERVERS.NET.	(US Army Research Lab)	128.63.2.53
I.ROOT-SERVERS.NET.	(Stockholm, Sweden)	192.36.148.17
J.ROOT-SERVERS.NET.	(Herndon, VA)	192.58.128.30
K.ROOT-SERVERS.NET.	(London, United Kingdom)	193.0.14.129
L.ROOT-SERVERS.NET.	(IANA, Los Angeles)	198.32.64.12
M.ROOT-SERVERS.NET.	(WIDE, Tokyo)	202.12.27.33

- Hard coded into every DNS resolver



## Resource Records

- A zone file includes a collection of resource records (RRs)
- (Name, Value, Type, Class, TTL)
  - Name and value are exactly what you expect
  - Type specifies how the Value should be interpreted
    - A, NS, CNAME, MX, AAAA
  - Class: allows other entities to define record types; IN is the widely used one to date
  - TTL: how long the record should be cached

# Resource Records

- The database records of the DNS distributed database are called **resource records (RR)**
- Resource records are stored in configuration files (zone files) at name servers.

Resource records for a zone→

db.mylab.com

```
$TTL 86400
mylab.com. IN SOA PC4.mylab.com.
                    hostmaster.mylab.com. (
                    1 ; serial
                    28800 ; refresh
                    7200 ; retry
                    604800 ; expire
                    86400 ; minimum ttl
                    )
;
mylab.com. IN NS PC4.mylab.com.
;
localhost      A      127.0.0.1
PC4.mylab.com. A      10.0.1.41
PC3.mylab.com. A      10.0.1.31
PC2.mylab.com. A      10.0.1.21
PC1.mylab.com. A      10.0.1.11
```

# Resource Records

```

$TTL 86400
mylab.com. IN SOA PC4.mylab.com.
Hostmaster.mylab.com. (
    1 ; serial
    28800 ; refresh
    7200 ; retry
    604800 ; expire
    86400 ; minimum ttl
)
;
mylab.com.      IN      NS      PC4.mylab.com.
;
localhost       A       127.0.0.1
PC4.mylab.com.  A       10.0.1.41
PC3.mylab.com.  A       10.0.1.31
PC2.mylab.com.  A       10.0.1.21
PC1.mylab.com.  A       10.0.1.11

```

← Max. age of cached data in seconds

← •Start of authority (SOA) record.  
Means: "This name server is authoritative for the zone Mylab.com"  
•PC4.mylab.com is the name server  
•hostmaster@mylab.com is the email address of the person in charge

← Name server (NS) record.  
One entry for each authoritative name server

← Address (A) records.  
One entry for each host address

[https://en.wikipedia.org/wiki/SOA\\_record](https://en.wikipedia.org/wiki/SOA_record)

namename of the zoneINzone class (usually IN for internet)

SOAabbreviation for *Start of Authority*

MNAMEPrimary master name server for this zone\* [UPDATE requests](#) should be forwarded toward the primary master<sup>[2]</sup>\* [NOTIFY requests](#) propagate outward from the primary master<sup>[3]</sup>

RNAMEEmail address of the administrator responsible for this zone. (As usual, the email address is encoded as a name. The part of the email address before the @becomes the first label of the name; the domain name after the @ becomes the rest of the name. In zone-file format, dots in labels are escaped with backslashes; thus the email address john.doe@example.com would be represented in a zone file as john\.doe.example.com.)

SERIALSerial number for this zone. If a secondary name server slaved to this one observes an increase in this number, the slave will assume that the zone has been updated and initiate a [zone transfer](#).

REFRESHnumber of seconds after which secondary name servers should query the master for the SOA record, to detect zone changes.

Recommendation for small and stable zones:[\[4\]](#) 86400 seconds (24 hours).

RETRYnumber of seconds after which secondary name servers should retry to request the serial number from the master if the master does not respond. It must be less than *Refresh*. Recommendation for small and stable zones:[\[4\]](#) 7200 seconds (2 hours).

EXPIREnumber of seconds after which secondary name servers should stop answering request for this zone if the master does not respond. This value must be bigger than the sum of *Refresh* and *Retry*. Recommendation for small and stable zones:[\[4\]](#) 3600000 seconds (1000 hours).

TTL, a.k.a. MINIMUM[Time To Live](#) for purposes of negative caching. Recommendation for small and stable zones:[\[4\]](#) 172800 seconds (2 days). Originally this field had the meaning of a *minimum* TTL value for resource records in this zone; it was changed to its current meaning by [RFC 2308](#).[\[5\]](#)

The MINIMUM field of the SOA controls the length of time that the negative result may be cached.

when a name server loads a zone, it forces the TTL of all authoritative RRs to be at least the MINIMUM field of the SOA, here 86400 seconds, or one day.

The *<time-to-refresh>* directive is the numerical value slave servers use to determine how long to wait before asking the master nameserver if any changes have been made to the zone.

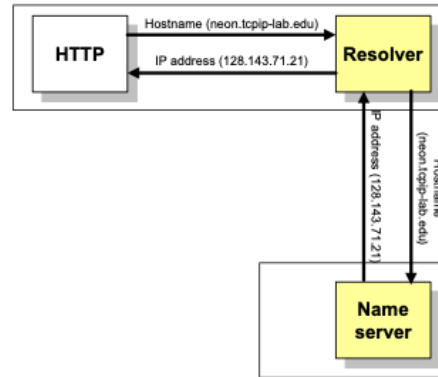
The *<time-to-refresh>* directive is the numerical value slave servers use to determine how long to wait before asking the master nameserver if any changes have been made to the zone.

The *<time-to-refresh>* directive is the numerical value slave servers use to determine how long to wait before asking the master nameserver if any changes have been made to the zone.

<http://rscott.org/dns/soa.html>: yes. Hostmaster.mylab.com is an email address. @ sign is not allowed in a rr.

# Domain name resolution

1. User program issues a request for the IP address of a hostname  
gethostbyname()
2. Local resolver formulates a **DNS query** to the name server of the host
3. Name server checks if it is authorized to answer the query.
  - a) If yes, it responds.
  - b) Otherwise, it will query other name servers, starting at the root tree
4. When the name server has the answer it sends it to the resolver.

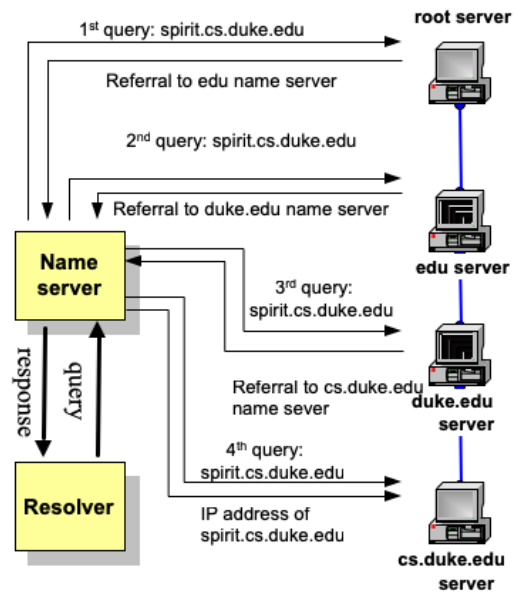


# Recursive and Iterative Queries

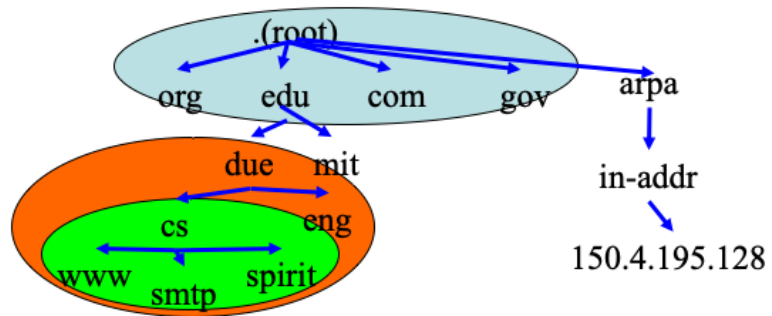
- There are two types of queries:
  - Recursive queries
  - Iterative (non-recursive) queries
- The type of query is determined by a bit in the DNS query
- **Recursive query:** When the name server of a host cannot resolve a query, the server issues a query to resolve the query
- **Iterative queries:** When the name server of a host cannot resolve a query, it sends a referral to another server to the resolver

# Recursive/iterative queries

- In a recursive query, the resolver expects the response from the name server
- If the server cannot supply the answer, it will send the query to the “closest known” authoritative name server (here: In the worst case, the closest known server is the root server)
- The root server sends a referral to the “edu” server. Querying this server yields a referral to the server of “duke.edu”
  - A “referral” is IP address to an intermediate name server
- ... and so on
- First: recursive
- Subsequent: iterative



## Inverse query



- What's the host name for IP address 128.195.4.150
  - IP address is converted to domain name: 150.4.195.128.in-addr.arpa
  - Resolver sends query for this address



## Canonical names and aliases

;; ANSWER SECTION:

```
www.cs.duke.edu.      86400 IN CNAME prophet.cs.duke.edu.  
prophet.cs.duke.edu. 86400 IN   A      152.3.140.5
```

- Hosts can have several names.
- One is called canonical names and others are called aliases

## Caching

- To reduce DNS traffic, name servers caches information on domain name/IP address mappings
- When an entry for a query is in the cache, the server does not contact other servers
- Note: If an entry is sent from a cache, the reply from the server is marked as “unauthoritative”
- Caching-only servers

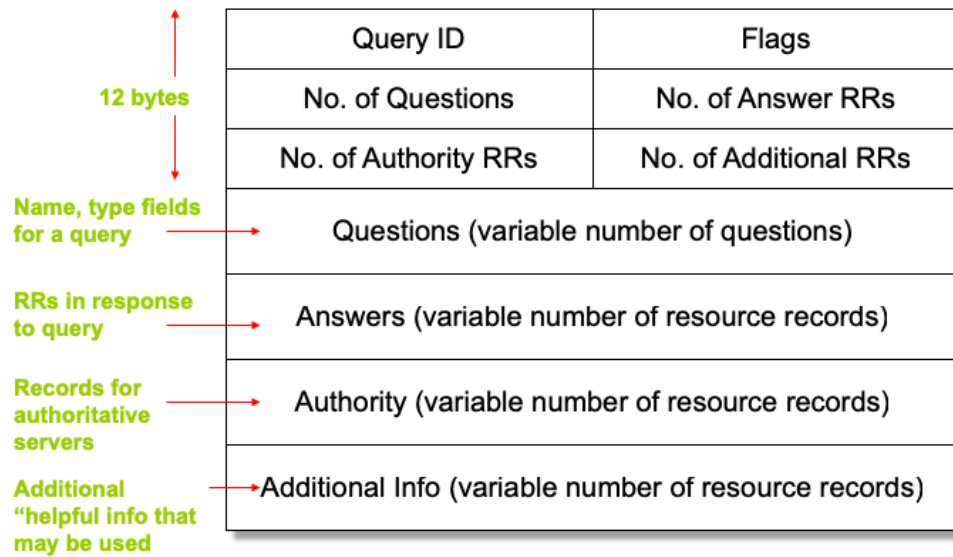
## Negative caching

- Two negative responses
  - Name in question does not exist
  - The name in record exists, but the requested data do not
- Negative responses will be cached too

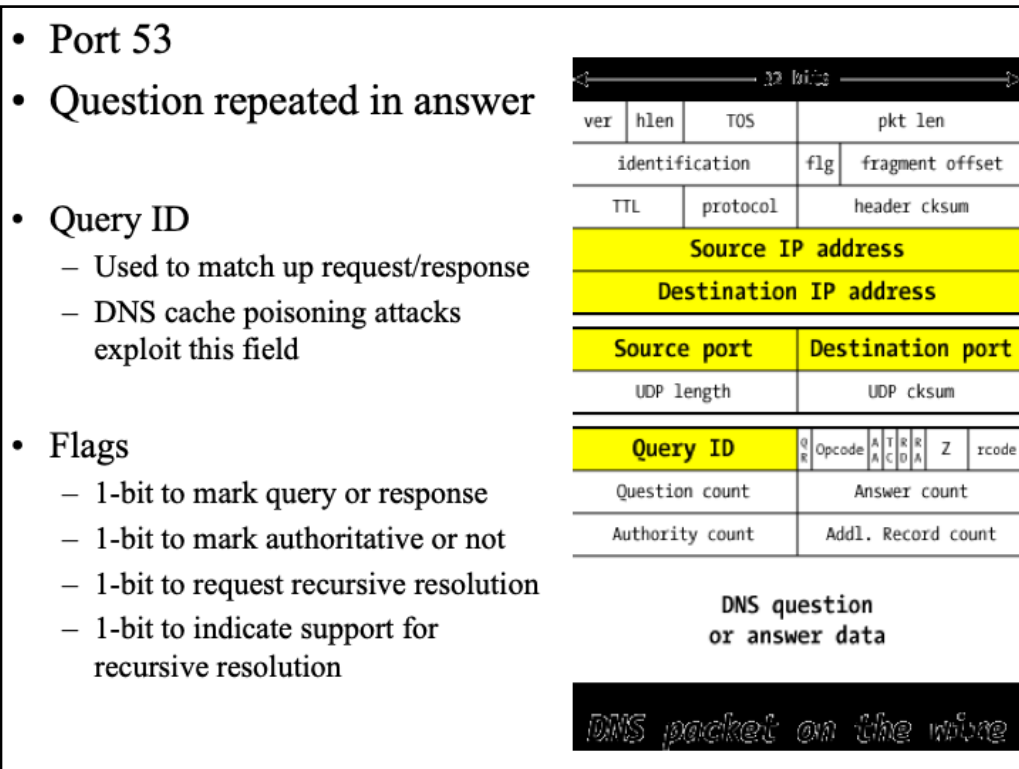
# Dig

- DNS lookup utility
- xwy@liberty:~\$ dig +norecurse @a.root-servers.net NS www.cs.duke.edu
- .....
- ;; QUESTION SECTION:
- ;www.cs.duke.edu. IN NS
- ;; AUTHORITY SECTION:
- edu. 172800 IN NS L.GTLD-SERVERS.NET.
- edu. 172800 IN NS G.GTLD-SERVERS.NET.
- edu. 172800 IN NS C.GTLD-SERVERS.NET.
- edu. 172800 IN NS D.GTLD-SERVERS.NET.
- edu. 172800 IN NS A.GTLD-SERVERS.NET.
- edu. 172800 IN NS F.GTLD-SERVERS.NET.
- edu. 172800 IN NS E.GTLD-SERVERS.NET.
- ;; ADDITIONAL SECTION:
- A.GTLD-SERVERS.NET. 172800 IN A 192.5.6.30
- A.GTLD-SERVERS.NET. 172800 IN AAAA 2001:503:a83e::2:30
- C.GTLD-SERVERS.NET. 172800 IN A 192.26.92.30
- D.GTLD-SERVERS.NET. 172800 IN A 192.31.80.30
- E.GTLD-SERVERS.NET. 172800 IN A 192.12.94.30
- F.GTLD-SERVERS.NET. 172800 IN A 192.35.51.30
- G.GTLD-SERVERS.NET. 172800 IN A 192.42.93.30
- L.GTLD-SERVERS.NET. 172800 IN A 192.41.162.30

# DNS Message Format



37



#### Source / Destination IP address

These reflect the IP addresses of the machines that sent and should receive the packet. It's possible to **forge** the source address, but pointless to forge the destination.

Analog in the real world: on an envelope sent in the US Mail, you can put anything you want as the return address — the source address — but if you lie about the recipient, it's not going to go where you want.

#### Source / Destination port numbers

DNS servers listen on port 53/udp for queries from the outside world, so the first packet of any exchange always includes 53 as the UDP destination port.

The source port varies considerably (though not enough, as we'll find shortly): sometimes it's also port 53/udp, sometimes it's a fixed port chosen at random by the operating system, and sometimes it's just a random port that changes every time.

As far as DNS functionality is concerned, the source port doesn't really matter as long as the replies get routed to it properly. But this turns out to be the crux of the problem at hand.

#### Query ID

This is a unique identifier created in the query packet that's left intact by the server sending the reply: it allows the server making the request to associate the answer with the question.

A nameserver might have many queries outstanding at one time — even multiple queries to the same server — so this Query ID helps match the answers with the awaiting questions.

This is also sometimes called the Transaction ID (TXID).

#### QR (Query / Response)

Set to **0** for a query by a client, **1** for a response from a server.

#### Opcode

Set by client to **0** for a standard query; the other types aren't used in our examples.

#### AA (Authoritative Answer)

Set to **1** in a server response if this answer is Authoritative, **0** if not.

#### TC (Truncated)

Set to **1** in a server response if the answer can't fit in the 512-byte limit of a UDP packet response; this means the client will need to try again with a TCP query, which doesn't have the same limits.

#### RD (Recursion Desired)

The client sets this to **1** if it wishes that the server will perform the entire lookup of the name recursively, or **0** if it just wants the best information the server has and the client will continue with the iterative query on its own. Not all nameservers will honor a recursive request (root servers, for instance, won't ever perform recursive queries).

RA (Recursion Available)

The server sets this to indicate that it will (**1**) or won't (**0**) support recursion.

Z — reserved

This is reserved and must be zero

rcode

Response code from the server: indicates success or failure

Question record count

The client fills in the next section with a single "question" record that specifies what it's looking for: it includes the name (**www.unixwiz.net**), the type (**A**, **NS**, **MX**, etc.), and the class (virtually always **IN**=Internet).

The server repeats the question in the response packet, so the question count is almost always **1**.

Answer/authority/additional record count

Set by the server, these provide various kinds of answers to the query from the client: we'll dig into these answers shortly.

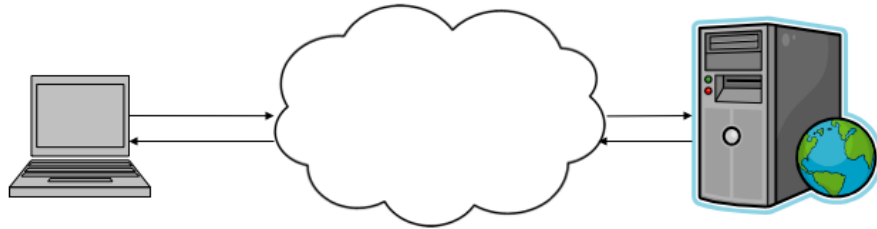
DNS Question/Answer data

This is the area that holds the question/answer data referenced by the count fields above. These will be discussed in great detail later.

## Server Selections and CDNs

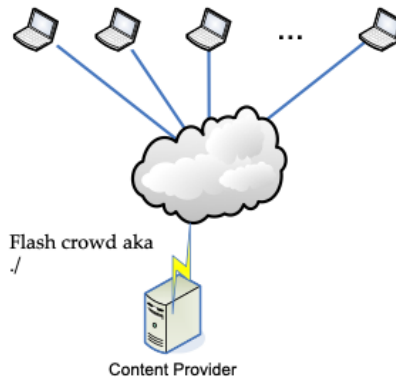


## A traditional web application



- HTTP request <http://www.cs.duke.edu>
- A DNS lookup on [www.cs.duke.edu](http://www.cs.duke.edu) returns the IP address of the web server
- Requests are sent to the web site

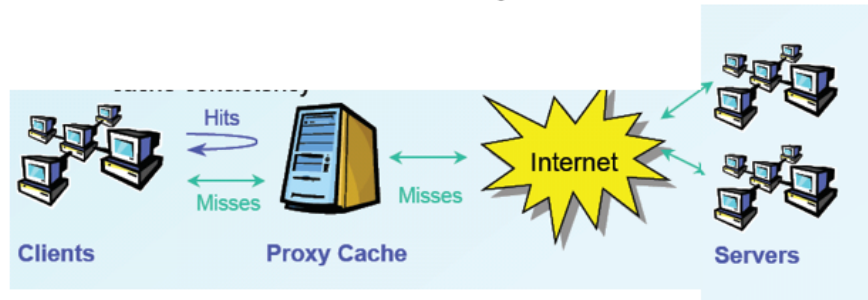
## What problem does CDN solve?



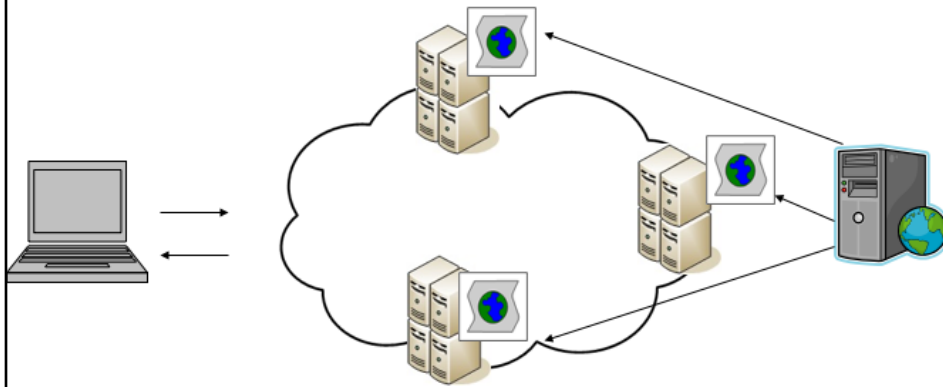
- Flash crowd may overwhelm a server and the access network
- Reduce latency, and network load

# Proxy caching

- Enhance web performance
  - Cache content
  - Reduce server load, latency, network utilization



## A content distribution network



- A single provider that manages multiple replicas
- A client obtains content from a close replica

## Pros and cons of CDN

- Pros
  - + Multiple content providers may use the same CDN
    - economy of scale
  - + All other advantages of proxy caching
  - + Fault tolerance
  - + Load balancing across multiple CDN nodes
- Cons
  - Expensive

## CDN challenges

- Balancing load among multiple caches
- Fault tolerant
- Low latency
- Cache consistency
- DDoS resistance

## How a CDN works

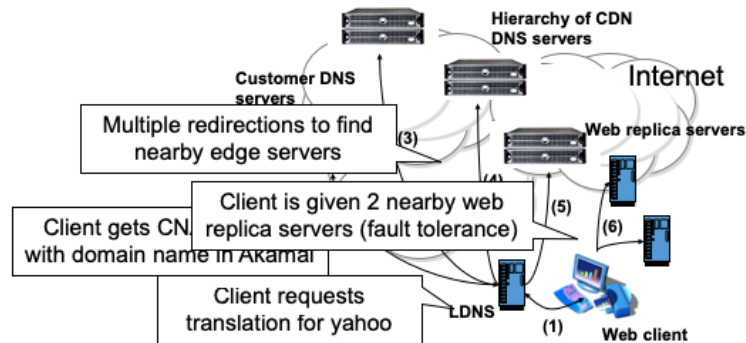
- One key technology:
  - DNS-based redirection: load balancing, latency
- Static content
  - Partial content

## DNS redirection

- Using a hierarchy of DNS servers that translate a client's web request to a nearby Akamai server
  - A client requests a DNS resolution ([www.yahoo.com](http://www.yahoo.com))
  - Akamai's customer's DNS name server uses a canonical name entry redirecting it to a DNS server in akamai's network
  - A hierarchy of DNS name servers responds to the DNS name-translation request
  - Name of the Akamai customer and the name of the requested content as a guide to determine the best two Akamai edge servers



## CDNs Basics



- Web client's request redirected to 'close' by server
  - Client gets web site's DNS CNAME entry with domain name in CDN network
  - Hierarchy of CDN's DNS servers direct client to 2 nearby servers

```

; <<>> DiG 9.4.2-P2 <<>> images.pcworld.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29098
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 9, ADDITIONAL: 2

;; QUESTION SECTION:
images.pcworld.com.                IN      A

;; ANSWER SECTION:
images.pcworld.com.                885     IN      CNAME   images.pcworld.com.edgesuite.net.
images.pcworld.com.edgesuite.net. 21585   IN      CNAME   a1694.g.akamai.net.
a1694.g.akamai.net.                5       IN      A       128.109.34.38
a1694.g.akamai.net.                5       IN      A       128.109.34.45

;; AUTHORITY SECTION:
g.akamai.net.                      973     IN      NS      n1g.akamai.net.
g.akamai.net.                      973     IN      NS      n2g.akamai.net.
g.akamai.net.                      973     IN      NS      n3g.akamai.net.
g.akamai.net.                      973     IN      NS      n4g.akamai.net.
g.akamai.net.                      973     IN      NS      n5g.akamai.net.
g.akamai.net.                      973     IN      NS      n6g.akamai.net.
g.akamai.net.                      973     IN      NS      n7g.akamai.net.
g.akamai.net.                      973     IN      NS      n8g.akamai.net.
g.akamai.net.                      973     IN      NS      n0g.akamai.net.

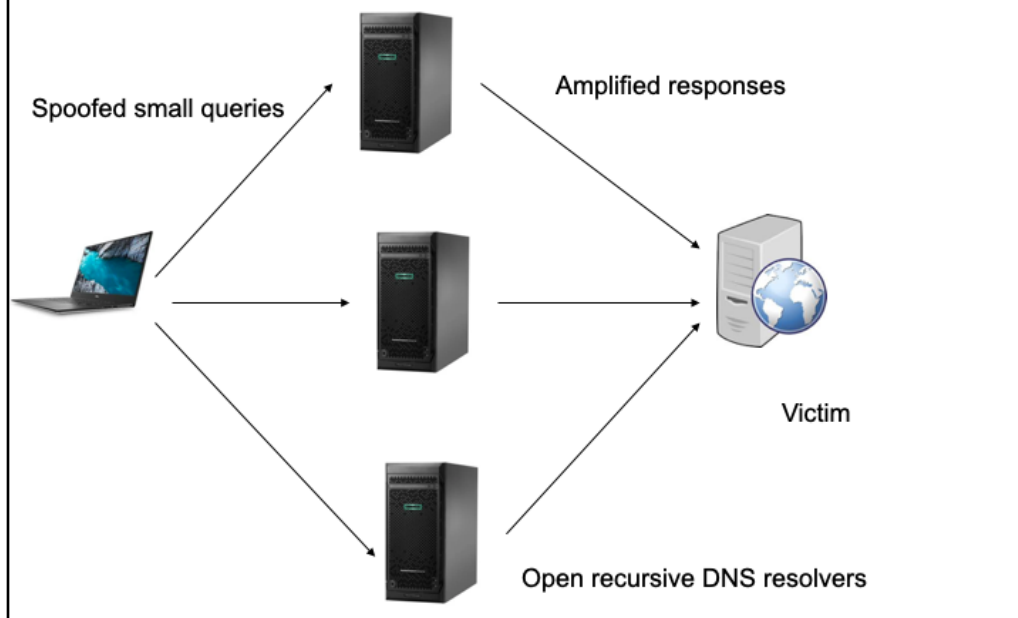
;; ADDITIONAL SECTION:
n1g.akamai.net.                    1663    IN      A       97.65.135.156
n5g.akamai.net.                    889     IN      A       128.109.247.10

;; Query time: 1 msec
;; SERVER: 152.3.140.1#53(152.3.140.1)
;; WHEN: Mon Feb 23 18:05:12 2009
;; MSG SIZE  revd: 337

```

## DNS attacks

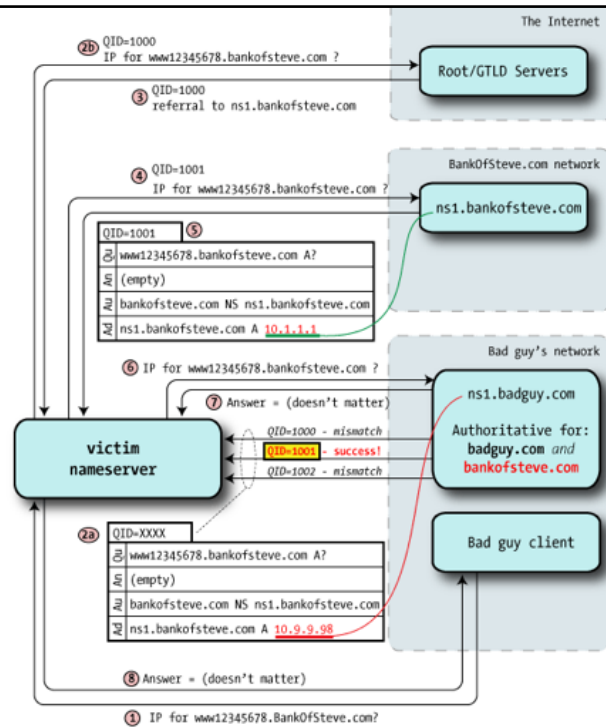
## DNS reflection attacks



Ways to amplify: To amplify a DNS attack, each DNS request can be sent using the EDNS0 DNS protocol extension, which allows for large DNS messages, or using the cryptographic feature of the DNS security extension (DNSSEC) to increase message size. Spoofed queries of the type "ANY," which returns all known information about a DNS zone in a single request, can also be used.

Upload a large TXT record, and query for that record

- DNS Cache Poisoning Attacks



## Conclusion

- DNS
- DNS and Content Distribution Networks
- DNS attacks