DNS protocol

- DNS uses UDP or TCP
- Special protocol - not simply an application, it’s a fundamental network protocol for making the Internet operate
- www.hawaii.edu ->
  - web3x-vip-www00.its.hawaii.edu ->
    - 128.171.133.5

DNS protocol

- The Internet has one global system for:
  - Addressing hosts IP
    (by design)

  - Naming hosts DNS
    By accident, an afterthought

DNS protocol

- The Internet has one global system for:
  - Addressing hosts IP
    (by design)
    - Numerical addresses appreciated by routers
    - Provide little (if any) information about location

  - Naming hosts DNS
    By accident, an afterthought
    - Naming appreciated by humans
    - Hierarchical, related to host location

Using Internet services can be divided into four logical steps

1. A person has name of entity she wants to access	www.hawaii.edu
2. She invokes an application to perform the task	Chrome
3. The application invokes DNS to resolve the name into an IP address	128.171.133.5
4. The application invokes transport protocol to establish an app-to-app connection	

The DNS system is a distributed database which enables to resolve a name into an IP address

www.hawaii.edu

128.171.133.5

In practice, names can be mapped to more than one IP

www.hawaii.edu	128.171.133.5
www.netflix.com	52.18.41.127
	34.248.59.175
	52.209.236.203
www.pschmitt.net	185.199.111.153
	185.199.110.153
	185.199.109.153
	185.199.108.153

In practice, names can be mapped to more than one IP

Why is this a good thing?

na

www.hawaii.edu	128.171.133.5
www.netflix.com	52.18.41.127
	34.248.59.175
	52.209.236.203
www.pschmitt.net	185.199.111.153
	185.199.110.153
	185.199.109.153
	185.199.108.153

In practice, names can be mapped to more than one IP

Why is this a good thing?
Load balancing
Reduce latency by picking nearby servers
Tailored content based on requester's location/identity

www.hawaii.edu	128.171.133.5
www.netflix.com	52.18.41.127
	34.248.59.175
	52.209.236.203
www.pschmitt.net	185.199.111.153
	185.199.110.153
	185.199.109.153
	185.199.108.153

In practice, IPs can be mapped by more than one name

www.hawaii.edu	128.171.133.5
ee449.pschmitt.net	128.171.216.7
salmo-trutta.pschmitt.net	128.171.216.7

Inserting Resource Records into DNS

- Example: just created startup “FooBar”
- Get a block of address space from ISP
  - Say 212.44.9.128/25
- Register foobar.com at Network Solutions (say)
  - Provide registrar with names and IP addresses of your authoritative name server (primary and secondary)
  - Registrar inserts RR pairs into the com TLD server:
    - (foobar.com, dns1.foobar.com, NS)
    - (dns1.foobar.com, 212.44.9.129, A)
- Put in your (authoritative) server dns1.foobar.com:
  - Type A record for www.foobar.com
  - Type MX record for foobar.com

Inserting Resource Records into DNS

- In addition, need to provide reverse PTR bindings
  - E.g., 212.44.9.129 " dns1.foobar.com
- Normally, these would go in 129.9.44.212.in-addr.arpa
- Problem: you can’t run the name server for that domain. Why not?
  - Because your block is 212.44.9.128/25, not 212.44.9.0/24
  - And whoever has 212.44.9.0/25 won’t be happy with you owning their PTR records
- Solution: ISP runs it for you
  - Now it’s more of a headache to keep it up-to-date :-(

Inserting Resource Records into DNS

- In addition, need to provide reverse PTR bindings
  - E.g., 212.44.9.1
- Normally, these r.arpa
- Problem: you c domain. Why not?
  - Because your 4
  - And whoever u owning their PTR records
- Solution: ISP runs it for you
  - Now it's more of a headache to keep it up-to-date :-(

Why do you think we use PTR records?

Inserting Resource Records into DNS

- In addition, need to provide reverse PTR bindings
  - E.g., 212.44.9.1
- Normally, these r.arpa
- Problem: you c domain. Why not?
  - Because your 4
  - And whoever u owning their PTR records
- Solution: ISP runs it for you
  - Now it's more of a headache to keep it up-to-date :-(

Why do you think we use PTR records?
Anti-spam
Logging

DNS Measurements (old study from 2000)

- What is being looked up?
  - ~60% requests for A records
  - ~25% for PTR records
  - ~5% for MX records
  - ~6% for ANY records
- How long does it take?
  - Median ~100msec (but 90th percentile ~500msec)
  - 80% have no referrals (I don’t know the answer but you should check this other server next); 99.9% have fewer than four
- Query packets per lookup: ~2.4

DNS Measurements (old study from 2000)

- Top 10% of names accounted for ~70% of lookups
  - Caching seems like an excellent idea
- 9% of lookups are unique
  - Cache hit rate can never exceed 91%
- Cache hit rates ~ 75%
  - But caching for more than 10 hosts doesn’t add much

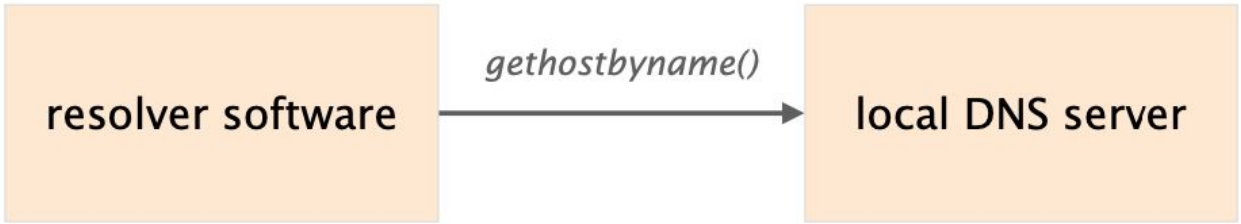
DNS Measurements (old study from 2000)

- Does DNS give answers?
  - ~23% of lookups fail to elicit an answer!
  - ~13% of lookups result in NXDOMAIN (or similar)
    - Mostly reverse lookups
  - Only ~64% of queries are successful!
    - How come the web seems to work so well?
- ~63% of DNS packets in unanswered queries!
  - Failing queries are frequently retransmitted
  - 99.9% successful queries have <=2 retransmissions

Moral of the story

- If you design a highly resilient system, many things can be going wrong without you noticing it

User perspective: DNS relies on two components



trigger resolution process

send request to local DNS server

usually, near the endhosts

configured statically (resolv.conf)

or dynamically (DHCP)

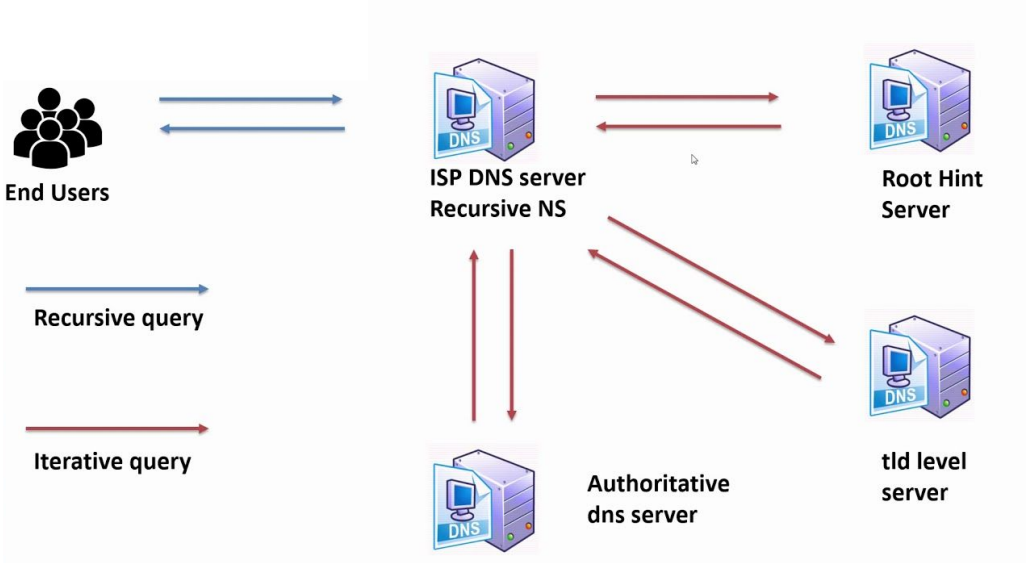
How does it work? Recursive vs Iterative Queries

Recursive query

- Ask each server to get answer for you

Iterative query

- Ask server who to ask next



Exercise

On Linux and Mac computers you can use the command line tool `dig` to perform DNS lookups. The corresponding tool for Windows is `nslookup`. First, perform a lookup for nyu.edu using your default DNS server by running the command `dig nyu.edu` or `nslookup nyu.edu`.

What is the IP address of the server behind nyu.edu?

Exercise

On Linux and Mac computers you can use the command line tool `dig` to perform DNS lookups. The corresponding tool for Windows is `nslookup`. First, perform a lookup for nyu.edu using your default DNS server by running the command `dig nyu.edu` or `nslookup nyu.edu`.

What is the IP address of the server behind nyu.edu?

Solution: Note that the actual IP address can depend on the local DNS server you use. We got the following answer with dig:

;; ANSWER SECTION:
nyu.edu. 60 IN A 216.165.47.10

Note that the format is slightly different for nslookup:

Non-authoritative answer:
Name: nyu.edu
Address: 216.165.47.10

Exercise

Now, perform the same lookup, but use one of the DNS root servers (e.g., a.root-servers.net) by running

```
dig @a.root-servers.net nyu.edu
```

```
nslookup nyu.edu a.root-servers.net
```

Exercise

Now, perform the same lookup, but use one of the DNS root servers (e.g., a.root-servers.net) by running

```
dig @a.root-servers.net nyu.edu
```

```
nslookup nyu.edu a.root-servers.net
```

Why does the answer differ compared to the one from your local DNS server?

Exercise

Now, perform the same lookup, but use one of the DNS root servers (e.g., a.root-servers.net) by running

```
dig @a.root-servers.net nyu.edu
```

```
nslookup nyu.edu a.root-servers.net
```

Why does the answer differ compared to the one from your local DNS server?

Solution: The request is not sent to an open DNS resolver, but to a DNS server that only provides answers about its own zone. Therefore, the root DNS server only points you to the name servers responsible for the next zone in the hierarchy, the edu zone.

Exercise

How would you proceed with this answer to find the IP address behind nyu.edu?

Exercise

How would you proceed with this answer to find the IP address behind nyu.edu?
Solution: Now that we know which servers are responsible for the edu zone, we can continue step-by-step just like your local DNS server would. Next, we would send a request to one of the edu name servers:

```
dig @a.edu-servers.net nyu.edu
```

The reply points us to the name servers in charge of the zone of NYU. By sending a request to them, we finally get the IP address behind the URL nyu.edu.