What is DNS?

Clara Gong

(D)omain (N)ame (S)ystem

Acts as a translator for the internet

Translates human oriented domain names into IP addresses and vice versa

Each device has its own IP that other machines use to connect

Also useful for abstraction of resources

Why a DNS server?

There are many many many many many IPs and hosts so it’s impossible to just know

Also it would be very slow if every computer needed to do its own manual search

Sometimes IP addresses change!

Domain Names, URLs, IP addresses

URL: https://suddenlysixam.club/projects/dns.html

https:// - protocol

suddenlysixam.club - domain

.club - top level domain

/projects/dns.html - path

DNS Queries

Ask a resolver for some information

Forward Lookup:

- Name -> IP (e.g. umiacs.umd.edu -> 128.8.120.33)

Reverse Lookup:

- IP -> Name (e.g. 128.8.120.33 -> umiacs.umd.edu)

DNS Queries (Examples)

```
bash-3.2$ nslookup umiacs.umd.edu
Server:         128.8.120.19
Address:     128.8.120.19#53


Name:    umiacs.UMD.EDU
Address: 128.8.120.33
```

```
bash-3.2$ nslookup 128.8.120.33
Server:         128.8.120.19
Address:     128.8.120.19#53


33.120.8.128.IN-ADDR.ARPA    name =
umiacswww-vip.umiacs.umd.edu.
```

Types of DNS servers

Recursive resolvers → server that accepts user queries and makes additional requests

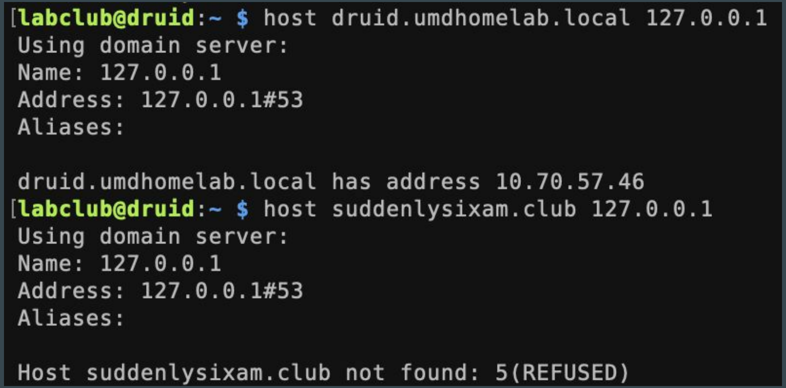
Root name server → top of hierarchy, determines where to search

Top Level Domain server → determines where to search at the domain level (e.x. .com)

Authoritative name server → gives IP address for requested address

Recursive Resolvers (Example)

Recursion not allowed:

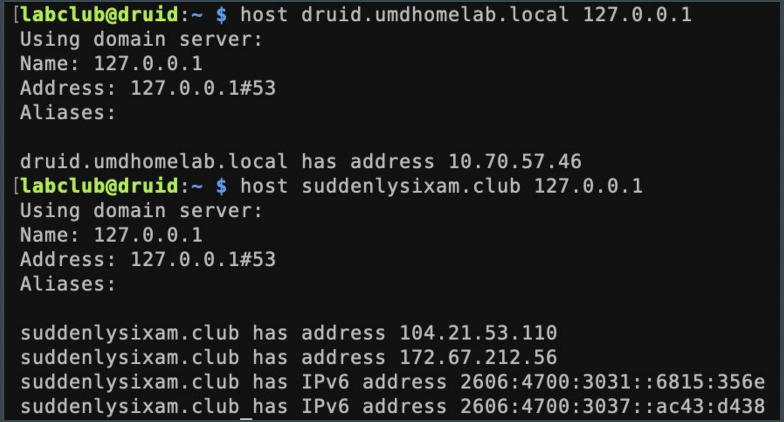


```
[labclub@druid:~ $ host druid.umdhomelab.local 127.0.0.1
Using domain server:
Name: 127.0.0.1
Address: 127.0.0.1#53
Aliases:

druid.umdhomelab.local has address 10.70.57.46
[labclub@druid:~ $ host suddenlysixam.club 127.0.0.1
Using domain server:
Name: 127.0.0.1
Address: 127.0.0.1#53
Aliases:

Host suddenlysixam.club not found: 5(REFUSED)
```

Recursion allowed:



```
[labclub@druid:~ $ host druid.umdhomelab.local 127.0.0.1
Using domain server:
Name: 127.0.0.1
Address: 127.0.0.1#53
Aliases:

druid.umdhomelab.local has address 10.70.57.46
[labclub@druid:~ $ host suddenlysixam.club 127.0.0.1
Using domain server:
Name: 127.0.0.1
Address: 127.0.0.1#53
Aliases:

suddenlysixam.club has address 104.21.53.110
suddenlysixam.club has address 172.67.212.56
suddenlysixam.club has IPv6 address 2606:4700:3031::6815:356e
suddenlysixam.club has IPv6 address 2606:4700:3037::ac43:d438
```

DNS Hierarchy

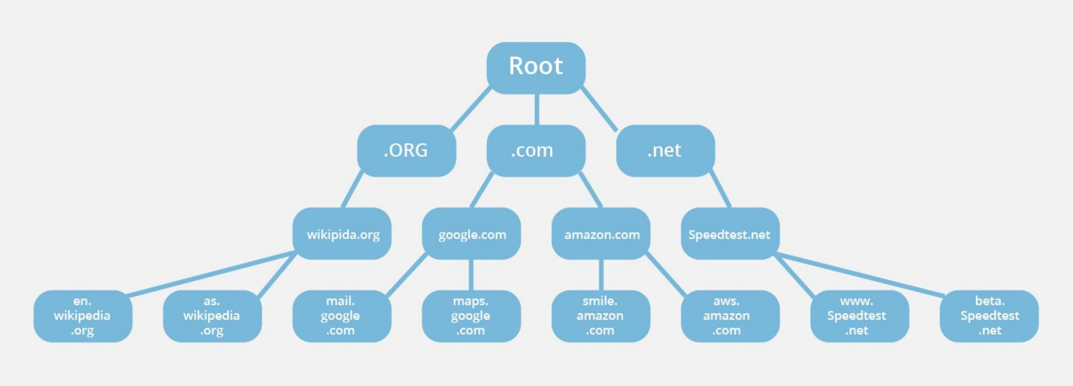
DNS is hierarchical, not a single giant database

Root servers (.)

Top Level Domain (TLD) servers (e.g. .com, .org, .edu, etc.)

Authoritative servers

Resolvers

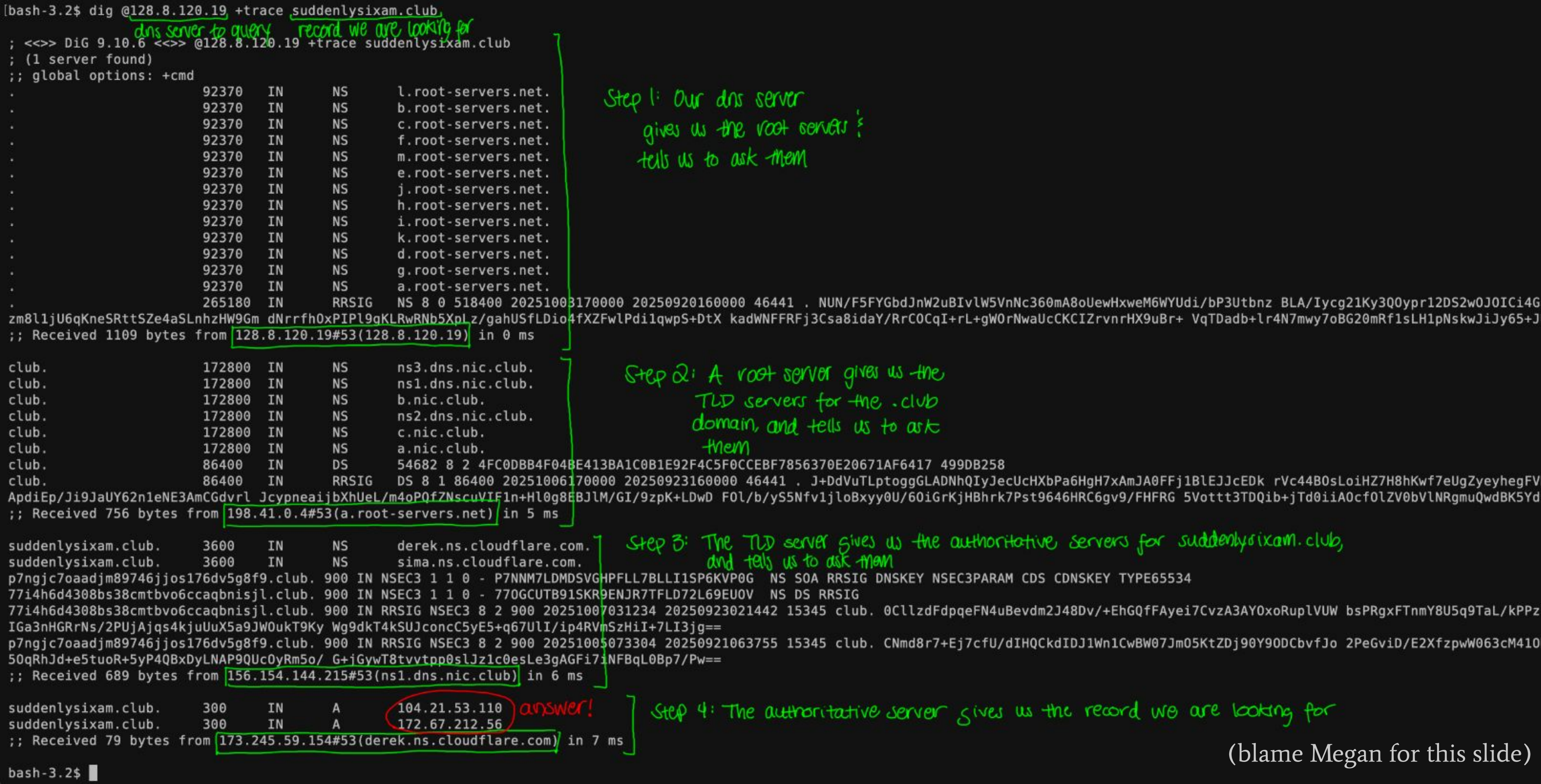


from: https://www.cloudflare.com/learning/dns/glossary/dns-root-server/

What happens when you make a query?

1. First check DNS cache for IP info
2. Send out a request that is received by a resolver
3. The resolver queries a root name server
4. The root name server returns which TLD server to search
5. The resolver sends another request to the TLD server specified
6. TLD server responds with which authoritative name server to search
7. Resolver sends request to name server
8. Name server gives the IP info
9. DNS resolver sends IP info to the original request location (e.x. Web browser)
10. Then it can do whatever it wants like make a web page request

What happens when you make a query? (Example)



```
[bash-3.2$ dig @128.8.120.19 +trace suddenlysixam.club

; <<>> DiG 9.10.6 <<>> @128.8.120.19 +trace suddenlysixam.club
; (1 server found)
;; global options: +cmd
.                       92370   IN      NS      l.root-servers.net.
.                       92370   IN      NS      b.root-servers.net.
.                       92370   IN      NS      c.root-servers.net.
.                       92370   IN      NS      f.root-servers.net.
.                       92370   IN      NS      m.root-servers.net.
.                       92370   IN      NS      e.root-servers.net.
.                       92370   IN      NS      j.root-servers.net.
.                       92370   IN      NS      h.root-servers.net.
.                       92370   IN      NS      i.root-servers.net.
.                       92370   IN      NS      k.root-servers.net.
.                       92370   IN      NS      d.root-servers.net.
.                       92370   IN      NS      g.root-servers.net.
.                       92370   IN      NS      a.root-servers.net.
.                       265180  IN      RRSIG   NS 8 0 518400 20251003170000 20250920160000 46441 . NUN/F5FYGbdJnW2uBIvlW5VnNc360mA8oUewHxweM6WYUdi/bP3Utbnz BLA/Iycg21Ky3QOypr12DS2wOJOICi4G
zm8l1jU6qKneSRttSZe4aSLnhzHW9Gm dNrrfhOxPIPl9gKLRwRNb5XpLz/gahUSfLDio4fXZFwlPdi1qwpS+DtX kadWNFFRFj3Csa8idaY/RrCOCqI+rL+gWOrNwaUcCKCIZrvnrHX9uBr+ VqTDadb+lr4N7mwy7oBG20mRf1sLH1pNskwJiJy65+J
;; Received 1109 bytes from 128.8.120.19#53(128.8.120.19) in 0 ms

club.                   172800  IN      NS      ns3.dns.nic.club.
club.                   172800  IN      NS      ns1.dns.nic.club.
club.                   172800  IN      NS      b.nic.club.
club.                   172800  IN      NS      ns2.dns.nic.club.
club.                   172800  IN      NS      c.nic.club.
club.                   172800  IN      NS      a.nic.club.
club.                   86400   IN      DS      54682 8 2 4FC0DBB4F04BE413BA1C0B1E92F4C5F0CCEBF7856370E20671AF6417 499DB258
club.                   86400   IN      RRSIG   DS 8 1 86400 20251006170000 20250923160000 46441 . J+DdVuTLptoggGLADNhQIyJecUcHXbPa6HgH7xAmJA0FFj1BlEJJcEDk rVc44BOsLoiHZ7H8hKwf7eUgZyeyhegFV
ApdiEp/Ji9JaUY62n1eNE3AmCGdvrl JcypneaijbXhUeL/m4oPQfZNscuVIF1n+Hl0g8EBJlM/GI/9zpK+LDwD FOl/b/yS5Nfv1jloBxyy0U/6OiGrKjHBhrk7Pst9646HRC6gv9/FHFRG 5Vottt3TDQib+jTd0iiAOcfOlZV0bVlNRgmuQwdBK5Yd
;; Received 756 bytes from 198.41.0.4#53(a.root-servers.net) in 5 ms

suddenlysixam.club.     3600    IN      NS      derek.ns.cloudflare.com.
suddenlysixam.club.     3600    IN      NS      sima.ns.cloudflare.com.
p7ngjc7oaadjm89746jjos176dv5g8f9.club. 900 IN NSEC3 1 1 0 - P7NNM7LDMDSVGHPFLL7BLLI1SP6KVP0G  NS SOA RRSIG DNSKEY NSEC3PARAM CDS CDNSKEY TYPE65534
77i4h6d4308bs38cmtbvo6ccaqbnisjl.club. 900 IN NSEC3 1 1 0 - 77OGCUTB91SKR9ENJR7TFLD72L69EUOV  NS DS RRSIG
77i4h6d4308bs38cmtbvo6ccaqbnisjl.club. 900 IN RRSIG NSEC3 8 2 900 20251007031234 20250923021442 15345 club. 0CllzdFdpqeFN4uBevdm2J48Dv/+EhGQfFAyei7CvzA3AYOxoRuplVUW bsPRgxFTnmY8U5q9TaL/kPPz
IGa3nHGRrNs/2PUjAjqs4kjuUuX5a9JWOukT9Ky Wg9dkT4kSUJconcC5yE5+q67UlI/ip4RVmSzHiI+7LI3jg==
p7ngjc7oaadjm89746jjos176dv5g8f9.club. 900 IN RRSIG NSEC3 8 2 900 20251005073304 20250921063755 15345 club. CNmd8r7+Ej7cfU/dIHQCkdIDJ1Wn1CwBW07JmO5KtZDj90Y9ODCbvfJo 2PeGviD/E2XfzpwW063cM41O
5OqRhJd+e5tuoR+5yP4QBxDyLNAP9QUcOyRm5o/ G+jGywT8tvvtpp0slJz1c0esLe3gAGFi7iNFBqL0Bp7/Pw==
;; Received 689 bytes from 156.154.144.215#53(ns1.dns.nic.club) in 6 ms

suddenlysixam.club.     300     IN      A       104.21.53.110
suddenlysixam.club.     300     IN      A       172.67.212.56
;; Received 79 bytes from 173.245.59.154#53(derek.ns.cloudflare.com) in 7 ms

bash-3.2$ ▌
```

dns server to query

record we are looking for

Step 1: Our dns server gives us the root servers & tells us to ask them

Step 2: A root server gives us the TLD servers for the .club domain, and tells us to ask them

Step 3: The TLD server gives us the authoritative servers for suddenlysixam.club, and tells us to ask them

answer!

Step 4: The authoritative server gives us the record we are looking for

(blame Megan for this slide)

Records

NAME	TTL	CLASS	TYPE	DATA
faculty	3600	IN	CNAME	umiacswww-vip.umiacs.umd.edu.

The name is the record you are looking up.

TTL (Time To Live) is how long this record may be cached for in seconds

Class is set to IN for internet queries

Type is the type of record

The data is type dependent but it’s going to be the data you were requesting

Record Types

- A - These records hold IP addresses like 128.8.120.19
- AAAA - These records hold IPv6 addresses
- CNAME - These are like aliases and have another hostname
- MX - this provides the name of the mail handlers for the record
- TXT - This a generic string. There can be many of these for one name
- NS - This is used to indicate the name server(s) to talk to for the name. These can be chained
- SOA - Tells you about the admins of the name, generally only domains will have these. ie umd.edu will have one but not www.umd.edu
- SRV - used for service discovery.
- PTR - Used for looking up IPS returns domains. IPv4 and IPv6 don't collide so only one type is needed for both.
- There are some less commonly used record types that we won't discuss.

IP Addresses

IPv4 (32 bit) → limited number of global IP addresses, uses NAT

IPv6 (128 bit) → designed to solve IP space limits and for more modern services

NAT → network address translation, can map multiple private IPs to 1 public IP

Private IP → used for communication within network

Public IP → for communication outside of network, global

Localhost / 127.0.0.1 → your own computer!

CIDR / Subnets / Netmasks

CIDR → classless inter-domain routing

IP addresses are made of a network prefix (MSB) and host identifiers (LSB)

Network prefix → identifies a whole network or subnet

Host identifier → marks specific host on network

CIDR works with variable length prefixes

Subnets → used to divide networks into parts using netmasks

e.x. 192.168.1.0/24 has 24 bit prefix and netmask 255.255.255.0

CIDR / Subnets / Netmasks (Examples)

Network: identify subnet
Usable: available for use
Broadcast: send traffic to all on subnet at once

CIDR Prefix	Binary (CIDR prefix = number of 1’s)	Subnet Mask	Total # of IP addresses	Example IP range:
/32	11111111 11111111 11111111 11111111	255.255.255.255	1	192.168.1.5/32 Single IP address: 192.168.1.5
/24	11111111 11111111 11111111 00000000	255.255.255.0	256	192.168.1.0/24 Network: 192.168.1.0 Usable: 192.168.1.1 - 192.168.1.254 Broadcast: 192.168.1.255
/22	11111111 11111111 11111100 00000000	255.255.252.0	1024	192.168.64.0/22 Network: 192.168.64.0 Usable: 192.168.64.1 - 192.168.67.254 Broadcast: 192.168.67.255
/16	11111111 11111111 00000000 00000000	255.255.0.0	65,536	192.168.0.0/16 Network: 192.168.0.0 Usable: 192.168.0.1 - 192.168.255.254 Broadcast: 192.168.255.255
/0	00000000 00000000 00000000 00000000	0.0.0.0	4,294,967,296	0.0.0.0/0 0.0.0.0 – 255.255.255.255

Static IP & DHCP

Static IP → permanently assigned, manual configuration

DHCP = Dynamic Host Configuration Protocol

DHCP → assigns temporary IPs and gives additional network info

host / nslookup / dig / ping

Host, nslookup, dig are all commands that attempt to provide more DNS info

Ping - sends packets to a thing and sees if it gets a response

```
>> host suddenlysixam.club
suddenlysixam.club has address 104.21.53.110
suddenlysixam.club has address 172.67.212.56
suddenlysixam.club has IPv6 address
2606:4700:3031::6815:356e
suddenlysixam.club has IPv6 address
2606:4700:3037::ac43:d438
```

```
>> ping suddenlysixam.club
PING suddenlysixam.club (104.21.53.110) 56(84) bytes of data.
64 bytes from 104.21.53.110 (104.21.53.110): icmp_seq=1 ttl=49
time=7.61 ms
--- suddenlysixam.club ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 7.612/7.612/7.612/0.000 ms
```

Question, comments, concerns?