DNS 101

Anurag Bhatia, Hurricane Electric

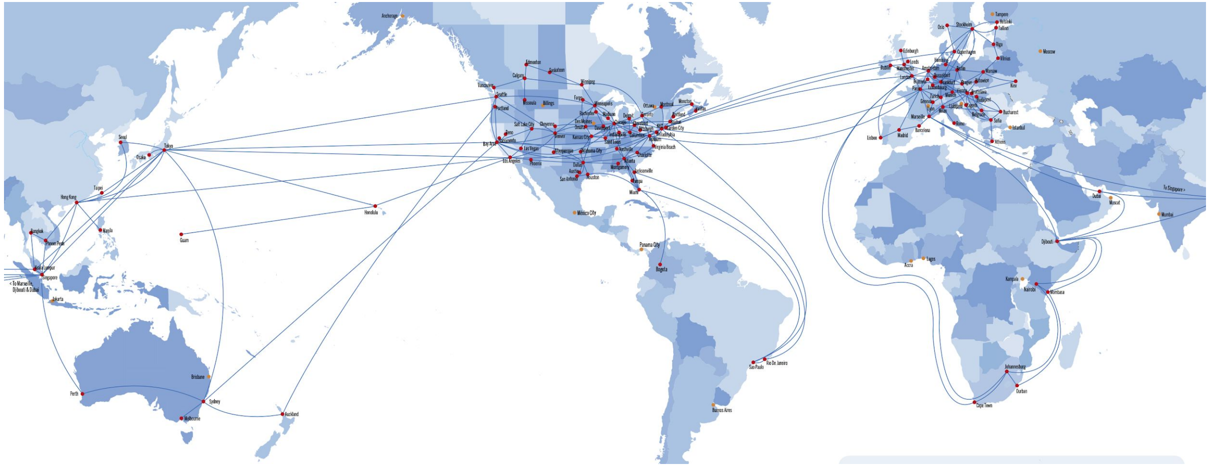
About me...

Working at Global backbone operator - Hurricane Electric and spend lot of time in looking at traceroutes, global routing, interesting patterns, issues etc

Besides routing have lot of interest in DNS, root DNS, IXPs, Network automation, tooling and virtualisation.

About Hurricane Electric

- Operating a Global IP backbone spanning across 45 countries, 240+ cities and multiple 100G rings across Atlantic & Pacifc.
- Connected with 8,588 networks in 29000+ BGP sessions across 257 locations.
- Operating at multi-terabit scale with 100s of Terabit of edge capacity.
- Doing filtering based on IRR and RPKI :-)



To be covered today...

1. Introduction to DNS
2. Root DNS Servers
3. Reverse DNS
4. Commonly used DNS records
5. Anycast - where it works, where it fails
6. How to setup anycast DNS with high availability
7. EDNS client subnet
8. DNS amplification attacks
9. Introduction to DNSSEC
10. Introduction to DoT and DoH

Housekeeping rules

1. Slide can be viewed on url given in the footer

2. Feel free to ask question in middle as we move from one topic to other. We will also try for discussion in the end.

3. Questions can be asked by typing in chat as well as by unmuting yourself.

DNS - Domain Name System

What is DNS?

1. A system designed to convert human readable hostnames into IP addresses. E.g he.net -> 216.218.236.2

2. System consists of set of servers called DNS servers to facilitate that.

3. System is based on hierarchical model

4. Besides BGP, DNS is what makes the internet as we know it!

Type of DNS servers

1. Authoritative DNS server

2. Recursive DNS server

3. DNS forwarder

Authoritative DNS servers

- Has the “authority” to the data of that given zone.
- Holds NS and SOA records for the given zone.
- Can give answer or point to another authoritative servers which has the delegation

Authoritative DNS servers

```
anurag@devops01 ~> dig he.net. ns +short
ns1.he.net.
ns4.he.net.
ns3.he.net.
ns5.he.net.
ns2.he.net.
anurag@devops01 ~>
```

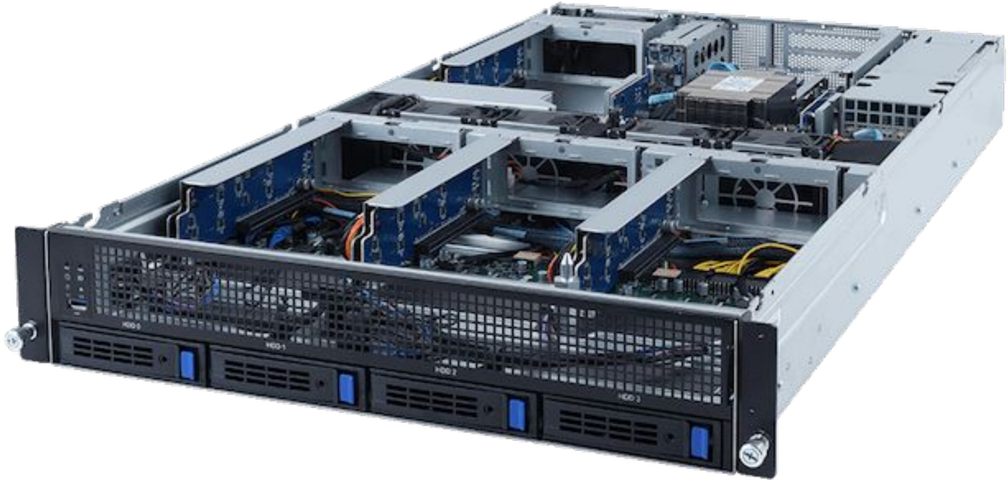
5 authoritative DNS servers which hold the zone data for “he.net”
They can either tell about all the records under “he.net” zone or can point further to a sub-zone NS which knows.

Characteristics of Authoritative DNS servers

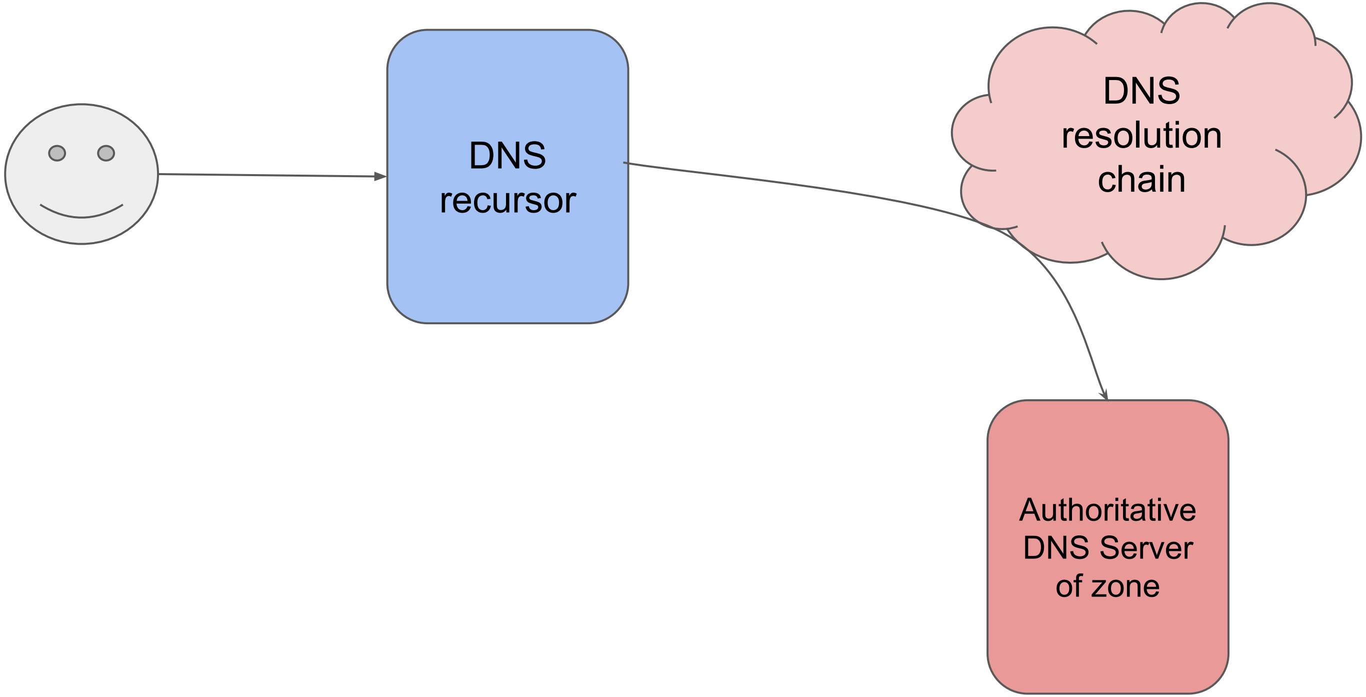
- Very often more than one is used. In fact many registrar’s require setup of atleast 2 on different IP address.
- All authoritative DNS servers are “supposed to” have identical records and anyone can be queried based on DNS resolver preference.
- One is master server & rest are slaves which sync up with the master
- Authoritative DNS zone hosting is low overhead operation in terms of compute.
- Holds DNS records like A, AAAA, MX, NS etc with TTL values
- CPU load depends on DNS queries per second

Recursive DNS servers

- Responsible for resolving a given hostname.
- Takes end to end responsibility of contacting entire DNS resolution chain to get the answer
- Keeps the reply in cache based on TTL of the responses.



Typical DNS resolution chain



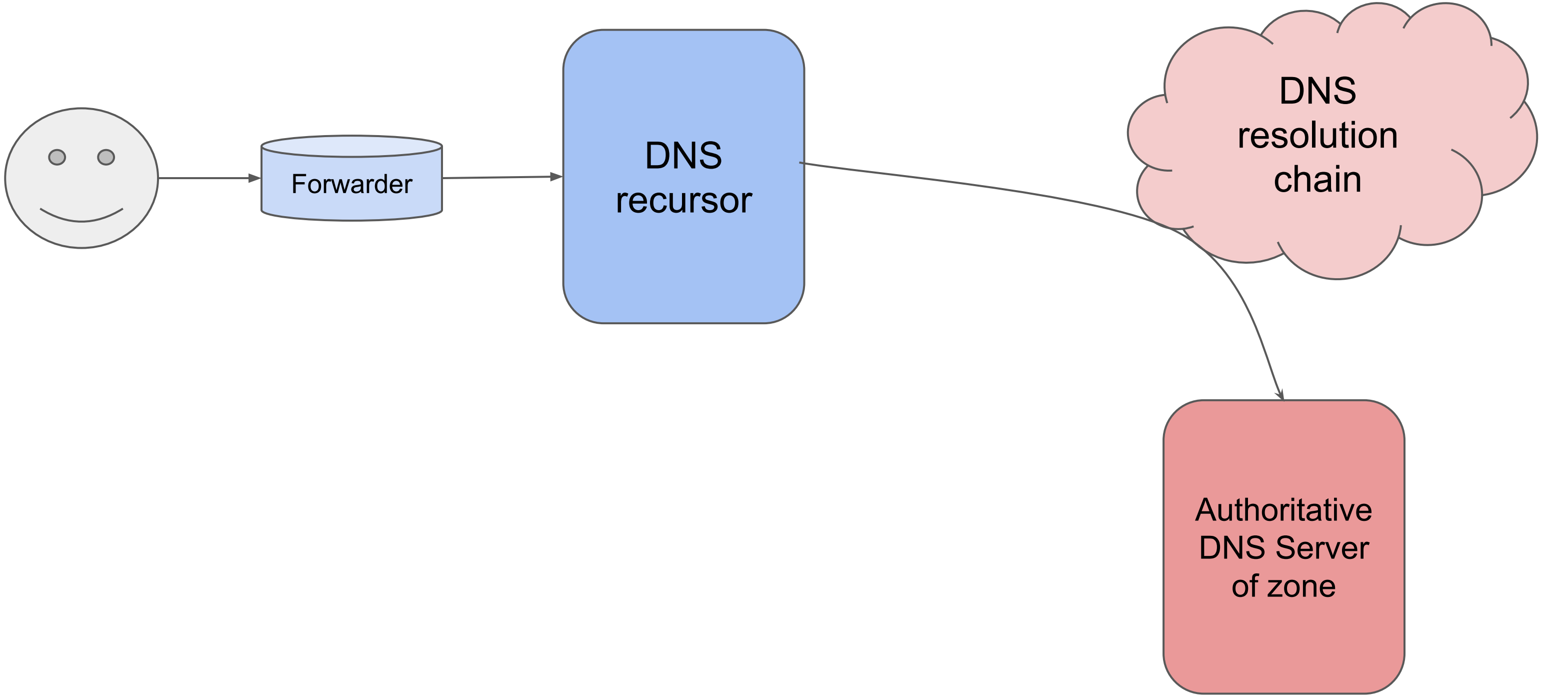
Characteristics of Recursive DNS servers

- Very often more than one server is used. Common to use anycasting here.
- Relatively a CPU intensive operation
- CPU load depends on DNS queries per second from the client & cached Vs non-cached replies
- Are offered by ISPs primarily but also available from Google (8.8.8.8), Cloudflare (1.1.1.1), PCH (9.9.9.9) etc.
- The ones with low latency & high cache typically give good performance

DNS forwarder

- Simply forward query to a DNS resolver & keep responses in cache
- Appear like “DNS resolver” to end user but in real simply forward queries
- Often run on routers/CPE and also security appliance. Can do DNS filtering

Typical DNS resolution chain



DNS resolution chain

Hierarchical structure of DNS

Why hierarchical structure?

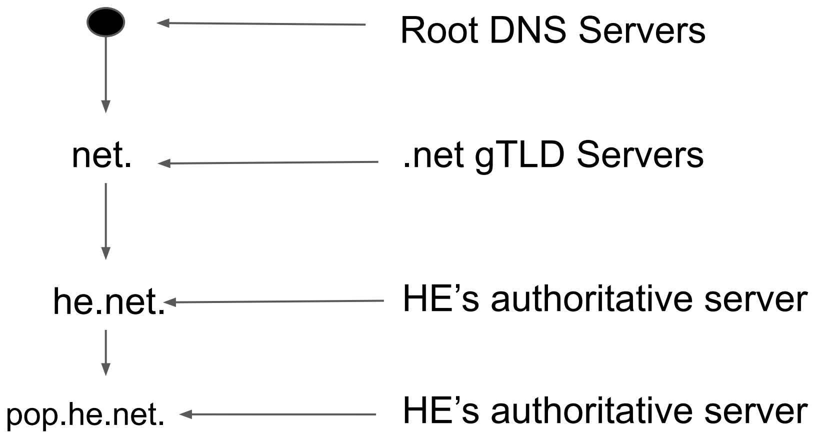
1. No single authority can hold data for all the domain names in the world. Imagine the scale - 1589 TLD (.com, .net, .in, .us) resulting in 370 million (37 crores) domain names across all TLDs. (in 2017 as per Wikipedia)

2. Ensures administrative boundary of the domain names making management easy.

3. High distribution = high uptime. Low risk of mass outage due to DDoS, Natural disasters, config mistakes etc.

Hierarchical structure example

Domain name: pop.he.net.



Root DNS Servers

net.

.net gTLD Servers

he.net.

HE’s authoritative server

pop.he.net.

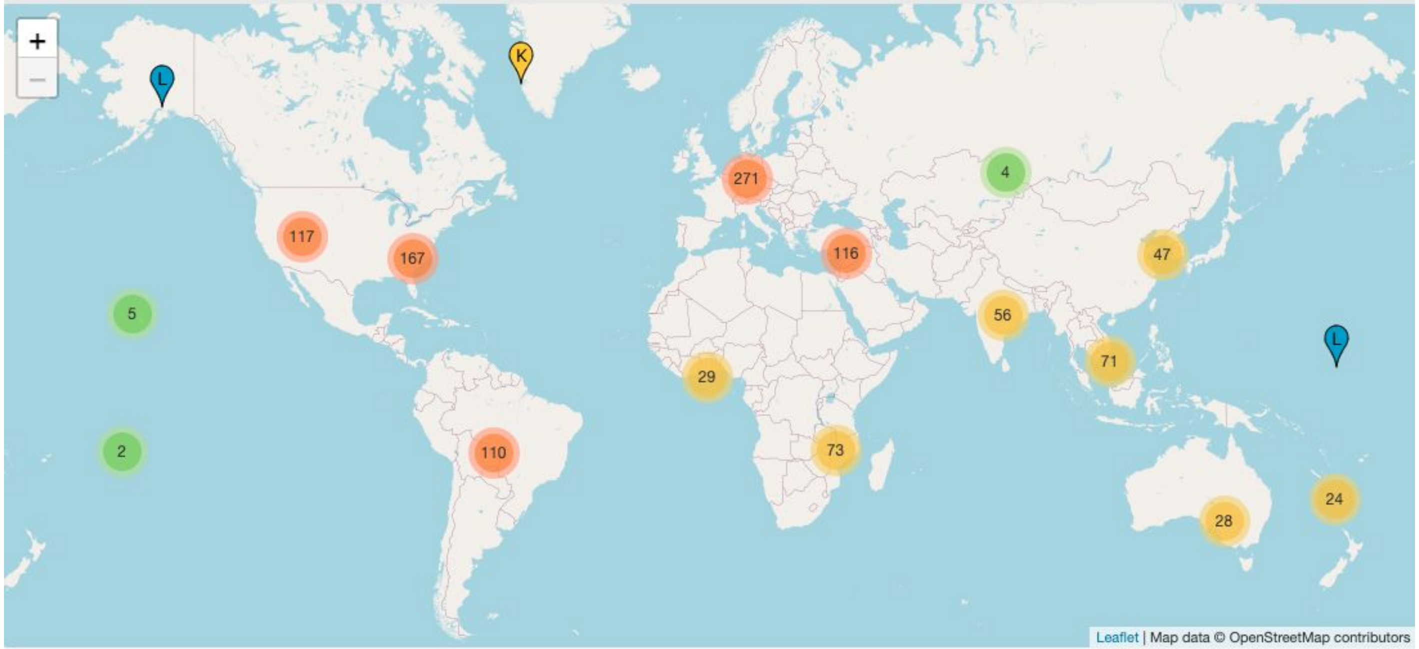
HE’s authoritative server

Root DNS servers

Root DNS servers

- Comes on top in the hierarchy of DNS resolution & defines the existence of domain names as we know them
- 13 logical systems run by 12 unique organisations
- Has definition of authoritative DNS servers of all TLDs like .com, .net, .in, .us etc
- All DNS resolvers in the world are hardcoded with the IP addresses of 13 root DNS servers
- 13 is simply number of unique IPs (max one can fit in 512 bytes), in reality they are anycasted with over 1300 instances across the world

Root DNS servers across the world

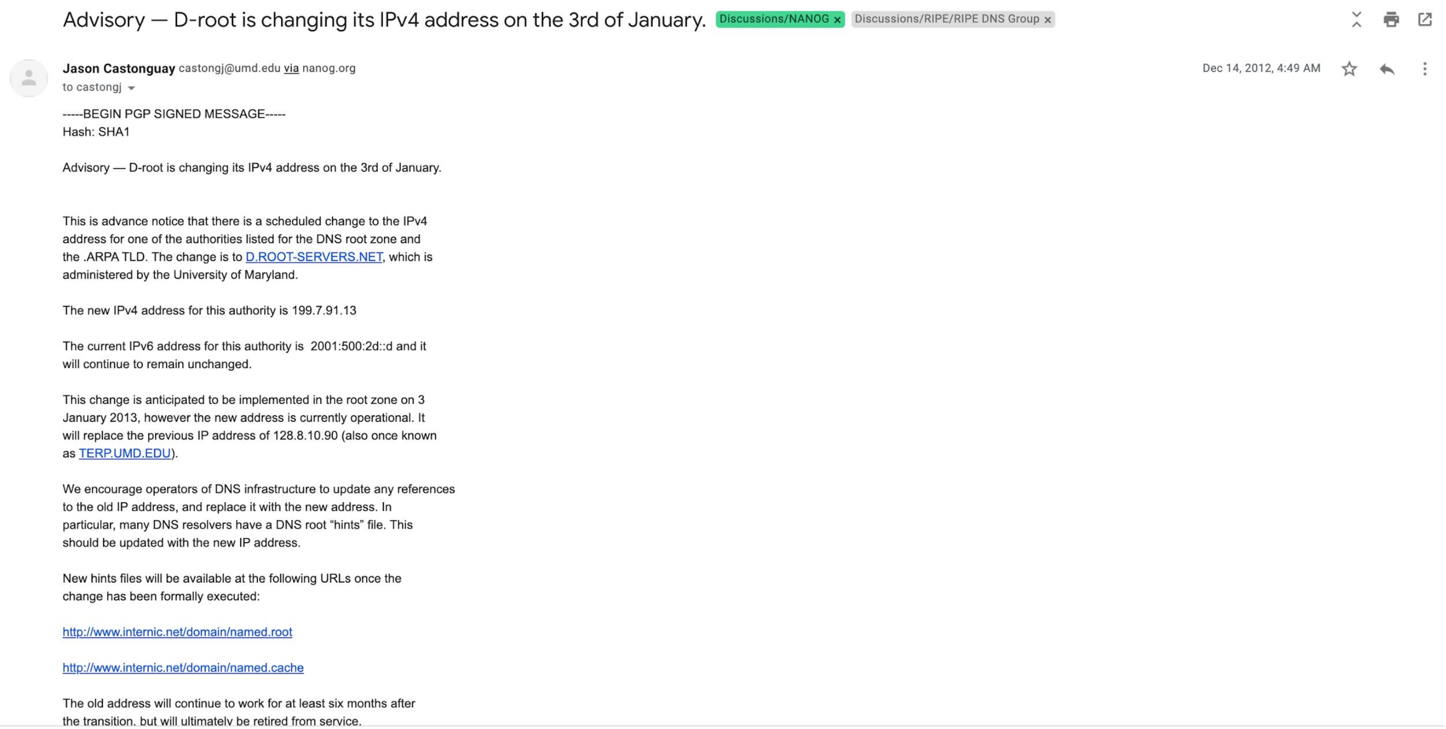


Source: https://root-servers.org/

Who operates root DNS servers?

Root Server	IPv4	IPv6	Organisation
a.root-servers.net.	198.41.0.4	2001:503:ba3e::2:30	Verisign Inc
b.root-servers.net.	199.9.14.201	2001:500:200::b	Information Sciences Institute
c.root-servers.net.	192.33.4.12	2001:500:2::c	Cogent
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
f.root-servers.net.	192.5.5.241	2001:500:2f::f	Internet Systems Consortium
g.root-servers.net.	192.112.36.4	2001:500:12::d0d	Defense Information Systems Agency
h.root-servers.net.	198.97.190.53	2001:500:1::53	U.S. 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

D root IPv4 renumbering 3rd Jan 2013



Advisory — D-root is changing its IPv4 address on the 3rd of January.

Jason Castonguay castongj@umd.edu via nanog.org
to castongj

Dec 14, 2012, 4:49 AM

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Advisory — D-root is changing its IPv4 address on the 3rd of January.

This is advance notice that there is a scheduled change to the IPv4 address for one of the authorities listed for the DNS root zone and the .ARPA TLD. The change is to D.ROOT-SERVERS.NET, which is administered by the University of Maryland.

The new IPv4 address for this authority is 199.7.91.13

The current IPv6 address for this authority is 2001:500:2d::d and it will continue to remain unchanged.

This change is anticipated to be implemented in the root zone on 3 January 2013, however the new address is currently operational. It will replace the previous IP address of 128.8.10.90 (also once known as TERP.UMD.EDU).

We encourage operators of DNS infrastructure to update any references to the old IP address, and replace it with the new address. In particular, many DNS resolvers have a DNS root “hints” file. This should be updated with the new IP address.

New hints files will be available at the following URLs once the change has been formally executed:

http://www.internic.net/domain/named.root

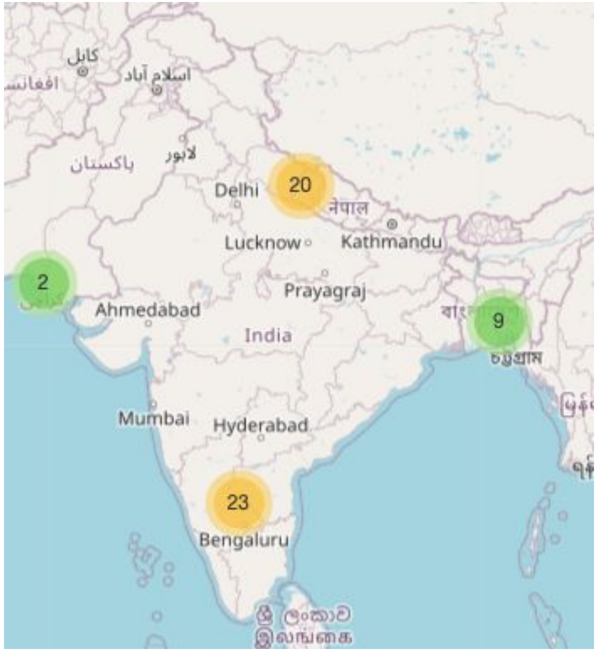
http://www.internic.net/domain/named.cache

The old address will continue to work for at least six months after the transition, but will ultimately be retired from service.

Source: https://seclists.org/nanog/2012/Dec/330

Root DNS servers across India

- 5 root DNS instances - 2 of J, E, F and K in Delhi NCR
- 2 root instances - E & F in Nagpur
- 1 instance of F root in Kolkata
- 7 instances in Mumbai - D, E, F, I, J, K, L
- 2 instances in Hyderabad - E & F
- 2 instances in Bangalore - F & J root
- 3 instances in Chennai - E and 2 F



Source: https://root-servers.org/

So many nodes, so things must be good?

Well, not really. Check are you even hitting local node?

Hint: Routing/peering relations are not perfect!

Live lookup at RIPE Atlas data on Root Instances: https://atlas.ripe.net/results/maps/root-instances

Which instance of root are you hitting?

Query in chaos class for id.server

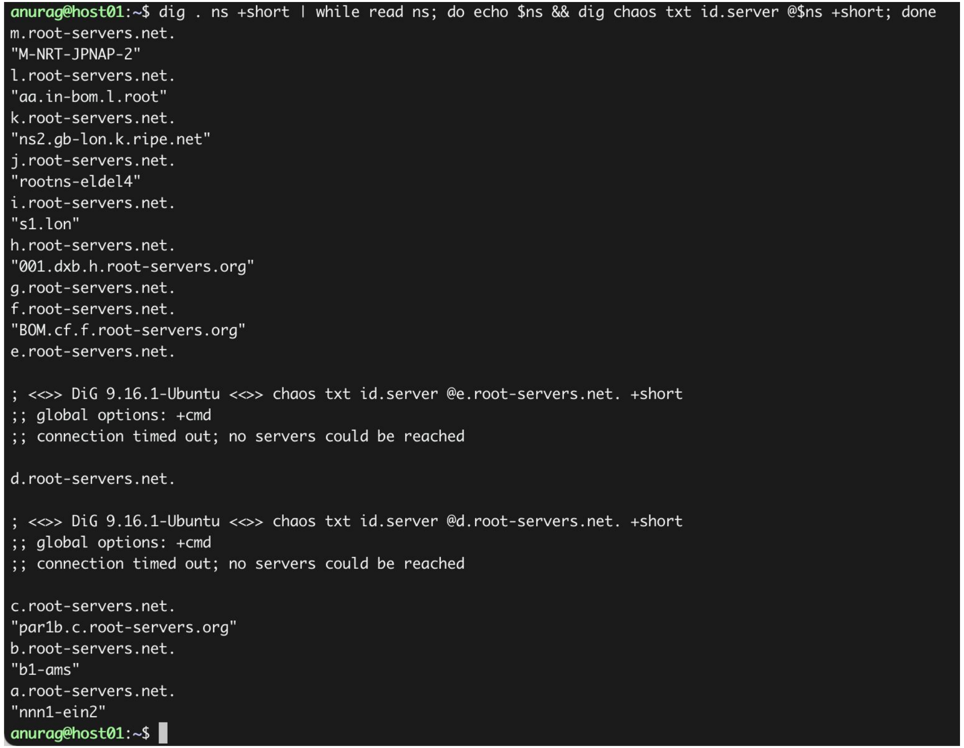
Command

Check for single server:

dig chaos txt id.server @a.root-servers.net. +short

Check for all root servers:

dig . ns +short | while read ns; do echo $ns && dig chaos txt id.server @$ns +short; done



```
anurag@host01:~$ dig . ns +short | while read ns; do echo $ns && dig chaos txt id.server @$ns +short; done
m.root-servers.net.
"M-NRT-JPNAP-2"
l.root-servers.net.
"aa.in-bom.l.root"
k.root-servers.net.
"ns2.gb-lon.k.ripe.net"
j.root-servers.net.
"rootns-eldel4"
i.root-servers.net.
"s1.lon"
h.root-servers.net.
"001.dxb.h.root-servers.org"
g.root-servers.net.
f.root-servers.net.
"BOM.cf.f.root-servers.org"
e.root-servers.net.

; <<>> DiG 9.16.1-Ubuntu <<>> chaos txt id.server @e.root-servers.net. +short
;; global options: +cmd
;; connection timed out; no servers could be reached

d.root-servers.net.

; <<>> DiG 9.16.1-Ubuntu <<>> chaos txt id.server @d.root-servers.net. +short
;; global options: +cmd
;; connection timed out; no servers could be reached

c.root-servers.net.
"par1b.c.root-servers.org"
b.root-servers.net.
"b1-ams"
a.root-servers.net.
"nnn1-ein2"
anurag@host01:~$
```

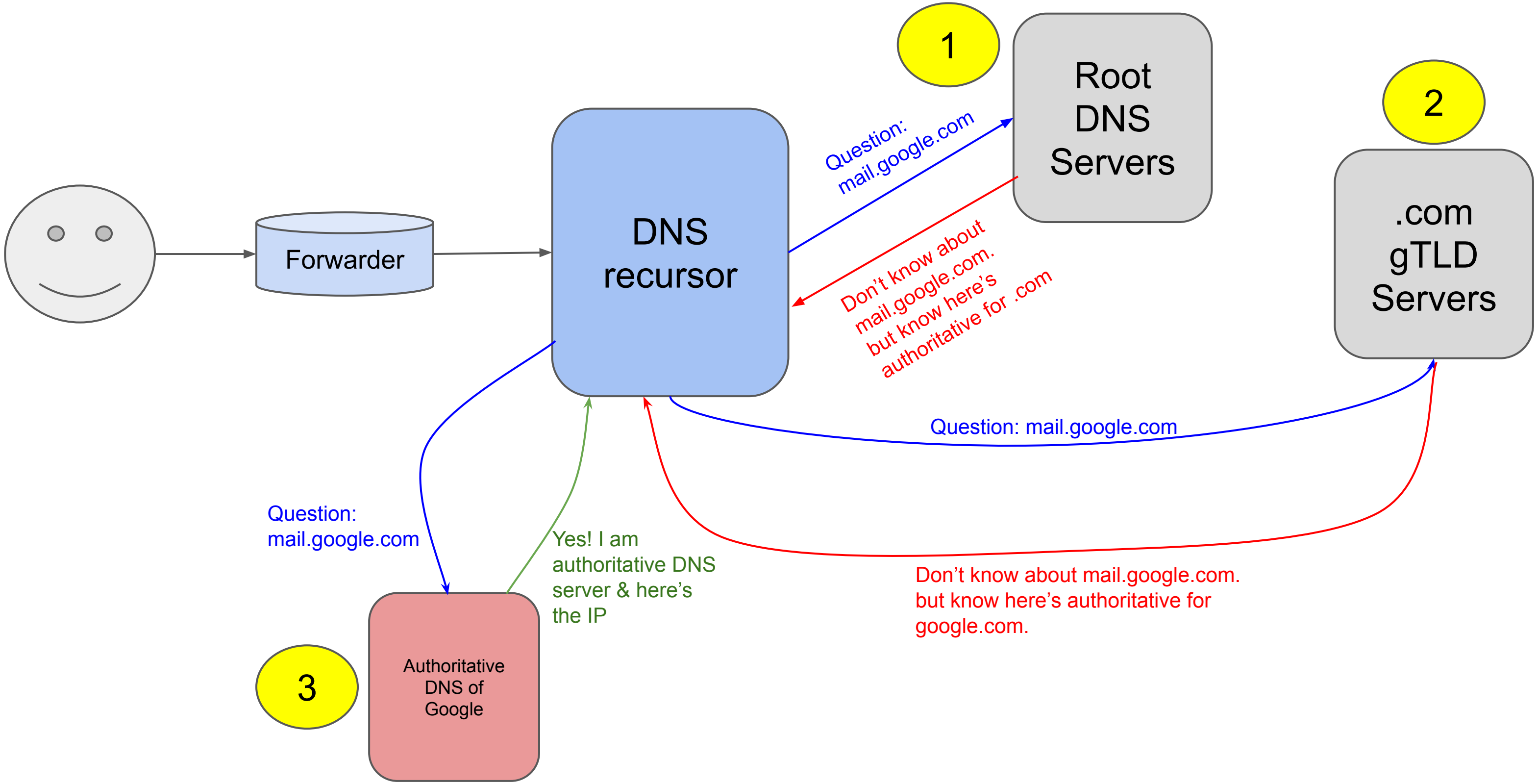
TLD servers

- Host zones for given Top Level Domains e.g GenricTLDs (gTLDs) like .com, .net and Country Code Top Level Domains (ccTLDs) like .in, .bd, .np, .us

- ccTLDs are administered often by respective countries with some exceptions.

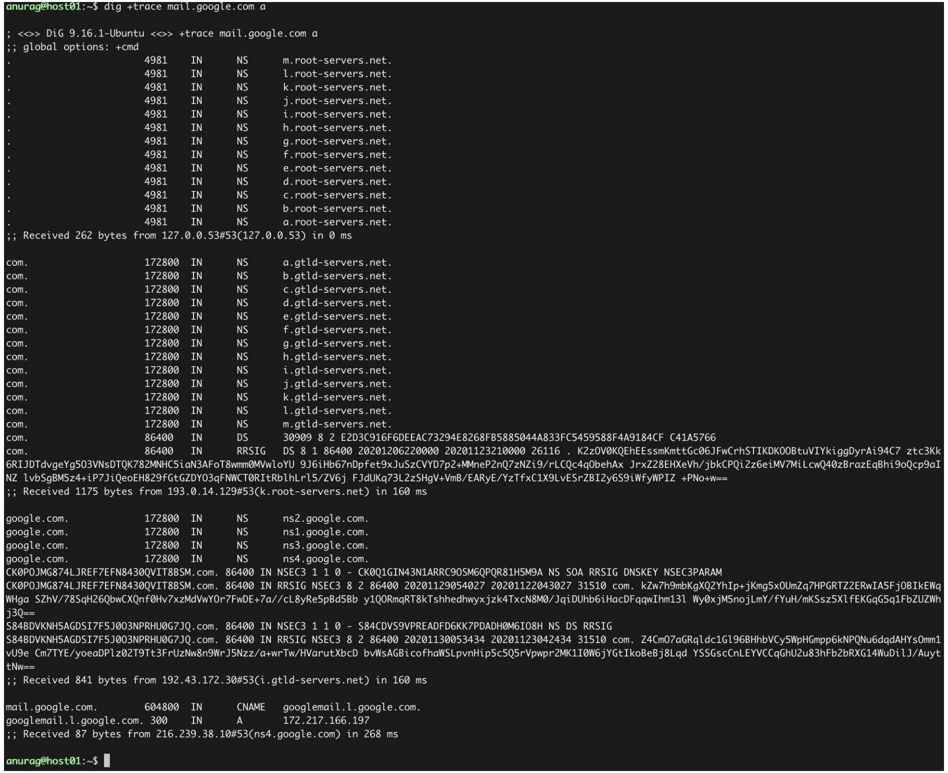
- When you register a domain name it goes as an entry into TLD servers of that respective TLDs.

Typical DNS resolution chain -> mail.google.com



1
Root DNS Servers
2
.com gTLD Servers
Forwarder
DNS recursor
Question: mail.google.com
Don't know about mail.google.com. but know here's authoritative for .com
Question: mail.google.com
Don't know about mail.google.com. but know here's authoritative for google.com.
Question: mail.google.com
Yes! I am authoritative DNS server & here's the IP
3
Authoritative DNS of Google

dig +trace mail.google.com a



```
anurag@host01:~$ dig +trace mail.google.com a

; <<>> DiG 9.16.1-Ubuntu <<>> +trace mail.google.com a
;; global options: +cmd
.                       4981    IN      NS      m.root-servers.net.
.                       4981    IN      NS      l.root-servers.net.
.                       4981    IN      NS      k.root-servers.net.
.                       4981    IN      NS      j.root-servers.net.
.                       4981    IN      NS      i.root-servers.net.
.                       4981    IN      NS      h.root-servers.net.
.                       4981    IN      NS      g.root-servers.net.
.                       4981    IN      NS      f.root-servers.net.
.                       4981    IN      NS      e.root-servers.net.
.                       4981    IN      NS      d.root-servers.net.
.                       4981    IN      NS      c.root-servers.net.
.                       4981    IN      NS      b.root-servers.net.
.                       4981    IN      NS      a.root-servers.net.
;; Received 262 bytes from 127.0.0.53#53(127.0.0.53) in 0 ms

com.                    172800  IN      NS      a.gtld-servers.net.
com.                    172800  IN      NS      b.gtld-servers.net.
com.                    172800  IN      NS      c.gtld-servers.net.
com.                    172800  IN      NS      d.gtld-servers.net.
com.                    172800  IN      NS      e.gtld-servers.net.
com.                    172800  IN      NS      f.gtld-servers.net.
com.                    172800  IN      NS      g.gtld-servers.net.
com.                    172800  IN      NS      h.gtld-servers.net.
com.                    172800  IN      NS      i.gtld-servers.net.
com.                    172800  IN      NS      j.gtld-servers.net.
com.                    172800  IN      NS      k.gtld-servers.net.
com.                    172800  IN      NS      l.gtld-servers.net.
com.                    172800  IN      NS      m.gtld-servers.net.
com.                    86400   IN      DS      30909 8 2 E2D3C916F6DEEAC73294E8268FB5885044A833FC5459588F4A9184CF C41A5766
com.                    86400   IN      RRSIG   DS 8 1 86400 20201206220000 20201123210000 26116 . K2zOV0KQEhEEssmKmttGc06JFwCrhSTIKDKOOBtuVIYkiggDyrAi94C7 ztc3Kk
6RIJDTdvgeYg5O3VNsDTQK782MNHC5iaN3AFoT8wmm0MVwloYU 9J6iHb67nDpfet9xJuSzCVYD7p2+MMneP2nQ7zNZi9/rLCQc4qObehAx JrxZ28EHXeVh/jbkCPQi2z6eiMV7MiLcwQ40zBrazEqBhi9oQcp9aI
NZ lvbSgBM5z4+iP7JiQeoEH829fGtGZDYO3qFNWCT0RItRblhLrl5/ZV6j FJdUKq73L2zSHgV+VmB/EARyE/YzTfxC1X9LvESrZBI2y6S9iWfyWPIZ +PNo+w==
;; Received 1175 bytes from 193.0.14.129#53(k.root-servers.net) in 160 ms

google.com.             172800  IN      NS      ns2.google.com.
google.com.             172800  IN      NS      ns1.google.com.
google.com.             172800  IN      NS      ns3.google.com.
google.com.             172800  IN      NS      ns4.google.com.
CK0POJMG874LJREF7EFN8430QVIT8BSM.com. 86400 IN NSEC3 1 1 0 - CK0Q1GIN43N1ARRC9OSM6QPQR81H5M9A NS SOA RRSIG DNSKEY NSEC3PARAM
CK0POJMG874LJREF7EFN8430QVIT8BSM.com. 86400 IN RRSIG NSEC3 8 2 86400 20201129054027 20201122043027 31510 com. kZw7h9mbKgXQ2YhIp+jKmg5xOUmZq7HPGRTZ2ERwIA5FjOBIkEWq
WHga SZhV/78SqH26QbwCXQnf0Hv7xzMdVwYOr7FwDE+7a//cL8yRe5pBd5Bb y1QORmqRT8kTshhedhwyxjzk4TxcN8M0/JqiDUhb6iHacDFqqwIhm13l Wy0xjM5nojLmY/fYuH/mKSsz5XlfEKGqG5q1FbZUZWh
j3Q==
S84BDVKNH5AGDSI7F5J0O3NPRHU0G7JQ.com. 86400 IN NSEC3 1 1 0 - S84CDVS9VPREADFD6KK7PDADH0M6IO8H NS DS RRSIG
S84BDVKNH5AGDSI7F5J0O3NPRHU0G7JQ.com. 86400 IN RRSIG NSEC3 8 2 86400 20201130053434 20201123042434 31510 com. Z4CmO7aGRqldc1Gl96BHhbVCy5WpHGmpp6kNPQNu6dqdAHYsOmm1
vU9e Cm7TYE/yoeaDPlz02T9Tt3FrUzNw8n9WrJ5Nzz/a+wrTw/HVarutXbcD bvWsAGBicofhaWSLpvnHip5c5Q5rVpwpr2MK1I0W6jYGtIkoBeBj8Lqd YSSGscCnLEYVCCqGhU2u83hFb2bRXG14WuDilJ/Auyt
tNw==
;; Received 841 bytes from 192.43.172.30#53(i.gtld-servers.net) in 160 ms

mail.google.com.        604800  IN      CNAME   googlemail.l.google.com.
googlemail.l.google.com. 300    IN      A       172.217.166.197
;; Received 87 bytes from 216.239.38.10#53(ns4.google.com) in 268 ms

anurag@host01:~$
```

Reverse DNS

Reverse DNS

- Often when DNS is referred, it’s forward DNS i.e hostname -> IP
- Reverse DNS maps IP to a hostname
- Useful for things like showing hostnames in the traceroute
- Needed by mail servers where forward record matches the reverse record besides few other things
- Reverse DNS is defined using PTR records
- Uses in-addr.arpa zone for IPv4 and ip6.arpa. Zone for IPv6

216.218.186.2 -> mail.he.net

Step1: Reverse the IP

2.186.218.216

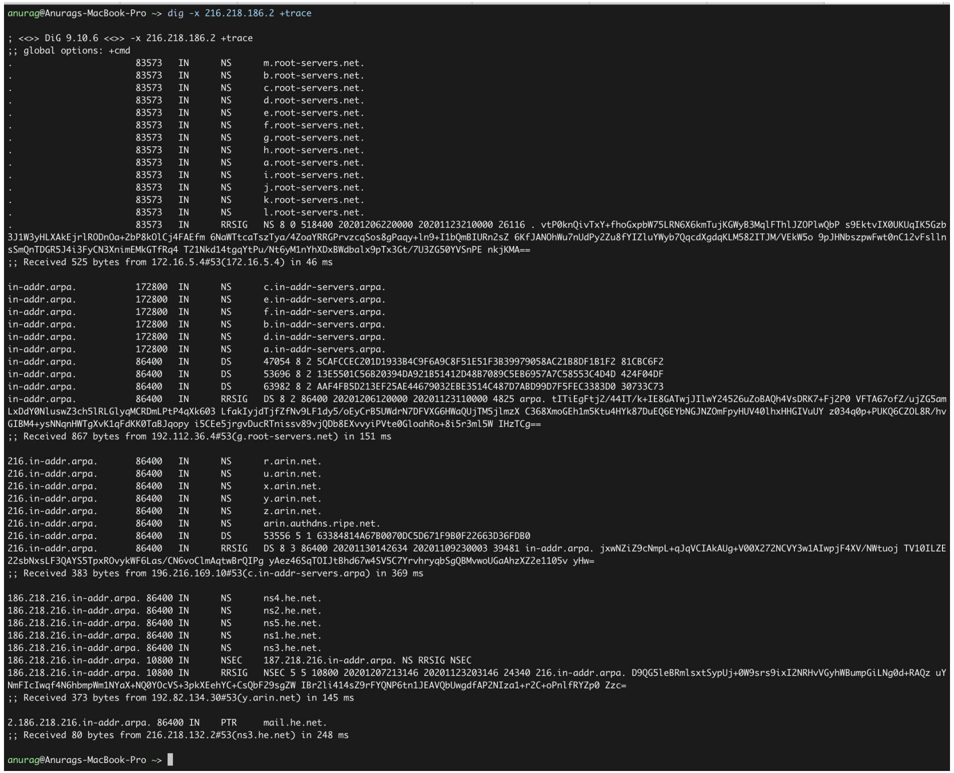
Step2: Add in-addr.arpa. in front of it

2.186.218.216.in-addr.arpa.

Step3: Point it to the hostname using record type PTR

2.186.218.216.in-addr.arpa. PTR mail.he.net.

Reverse DNS lookup: 216.218.186.2



```
anurag@Anurags-MacBook-Pro ~> dig -x 216.218.186.2 +trace

; <<>> DiG 9.10.6 <<>> -x 216.218.186.2 +trace
;; global options: +cmd
.                       83573   IN      NS      m.root-servers.net.
.                       83573   IN      NS      b.root-servers.net.
.                       83573   IN      NS      c.root-servers.net.
.                       83573   IN      NS      d.root-servers.net.
.                       83573   IN      NS      e.root-servers.net.
.                       83573   IN      NS      f.root-servers.net.
.                       83573   IN      NS      g.root-servers.net.
.                       83573   IN      NS      h.root-servers.net.
.                       83573   IN      NS      a.root-servers.net.
.                       83573   IN      NS      i.root-servers.net.
.                       83573   IN      NS      j.root-servers.net.
.                       83573   IN      NS      k.root-servers.net.
.                       83573   IN      NS      l.root-servers.net.
.                       83573   IN      RRSIG   NS 8 0 518400 20201206220000 20201123210000 26116 . vtP0knQivTxY+fhoGxpbW75LRN6X6kmTujKGWyB3MqlFThlJZOPlwQbP s9EktvIX0UKUqIK5Gzb
3J1W3yHLXAkEjrlRODnOa+2bP8kOlCj4FAEfm 6NaWTtcaTszTya/4ZoaYRRGPrvzcqSos8gPaqy+ln9+I1bQmBIURn2sZ 6KfJANOhWu7nUdPy2Zu8fYIZluYWyb7QqcdXgdqKLM582ITJM/VEkW5o 9pJHNbszpwFwt0nC12vFslln
sSmQnTDGR5J4i3FyCN3XnimEMkGTfRq4 T21Nkd14tgqYtPu/Nt6yM1nYhXDxBWdbalx9pTx3Gt/7U3ZG50YVSnPE nkjKMA==
;; Received 525 bytes from 172.16.5.4#53(172.16.5.4) in 46 ms

in-addr.arpa.           172800  IN      NS      c.in-addr-servers.arpa.
in-addr.arpa.           172800  IN      NS      e.in-addr-servers.arpa.
in-addr.arpa.           172800  IN      NS      f.in-addr-servers.arpa.
in-addr.arpa.           172800  IN      NS      b.in-addr-servers.arpa.
in-addr.arpa.           172800  IN      NS      d.in-addr-servers.arpa.
in-addr.arpa.           172800  IN      NS      a.in-addr-servers.arpa.
in-addr.arpa.           86400   IN      DS      47054 8 2 5CAFCCEC201D1933B4C9F6A9C8F51E51F3B39979058AC21B8DF1B1F2 81CBC6F2
in-addr.arpa.           86400   IN      DS      53696 8 2 13E5501C56B20394DA921B51412D48B7089C5EB6957A7C58553C4D4D 424F04DF
in-addr.arpa.           86400   IN      DS      63982 8 2 AAF4FB5D213EF25AE44679032EBE3514C487D7ABD99D7F5FEC3383D0 30733C73
in-addr.arpa.           86400   IN      RRSIG   DS 8 2 86400 20201206120000 20201123110000 4825 arpa. tITiEgFtj2/44IT/k+IE8GATwjJIlwY24526uZoBAQh4VsDRK7+Fj2P0 VFTA67ofZ/ujZG5am
LxDdY0NluswZ3ch5lRLGlyqMCRDmLPtP4qXk603 LfakIyjdTjfZfNv9LF1dy5/oEyCrB5UWdrN7DFVXG6HWaQUjTM5jlmzX C368XmoGEh1m5Ktu4HYk87DuEQ6EYbNGJNZOmFpyHUV40lhxHHGIVuUY z034q0p+PUKQ6CZOL8R/hv
GIBM4+ysNNqnHWTgXvK1qFdKK0TaBJqopy i5CEe5jrgvDucRTnissv89vjQDb8EXvvyiPVte0GloahRo+8i5r3ml5W IHzTCg==
;; Received 867 bytes from 192.112.36.4#53(g.root-servers.net) in 151 ms

216.in-addr.arpa.       86400   IN      NS      r.arin.net.
216.in-addr.arpa.       86400   IN      NS      u.arin.net.
216.in-addr.arpa.       86400   IN      NS      x.arin.net.
216.in-addr.arpa.       86400   IN      NS      y.arin.net.
216.in-addr.arpa.       86400   IN      NS      z.arin.net.
216.in-addr.arpa.       86400   IN      NS      arin.authdns.ripe.net.
216.in-addr.arpa.       86400   IN      DS      53556 5 1 63384814A67B0070DC5D671F9B0F22663D36FDB0
216.in-addr.arpa.       86400   IN      RRSIG   DS 8 3 86400 20201130142634 20201109230003 39481 in-addr.arpa. jxwNZiZ9cNmpL+qJqVCIAkAUg+V00X272NCVY3w1AIwpjF4XV/NWtuoj TV10ILZE
22sbNxsLF3QAYS5TpxROvykWF6Las/CN6voClmAqtwBrQIPg yAez46SqTOIJtBhd67w4SV5C7YrvhryqbSgQBMvwoUGaAhzXZ2e1105v yHw=
;; Received 383 bytes from 196.216.169.10#53(c.in-addr-servers.arpa) in 369 ms

186.218.216.in-addr.arpa. 86400 IN      NS      ns4.he.net.
186.218.216.in-addr.arpa. 86400 IN      NS      ns2.he.net.
186.218.216.in-addr.arpa. 86400 IN      NS      ns5.he.net.
186.218.216.in-addr.arpa. 86400 IN      NS      ns1.he.net.
186.218.216.in-addr.arpa. 86400 IN      NS      ns3.he.net.
186.218.216.in-addr.arpa. 10800 IN      NSEC    187.218.216.in-addr.arpa. NS RRSIG NSEC
186.218.216.in-addr.arpa. 10800 IN      RRSIG   NSEC 5 5 10800 20201207213146 20201123203146 24340 216.in-addr.arpa. D9QG5leBRmlsxtSypUj+0W9srs9ixI2NRHvVGyhWBumpGiLNg0d+RAQz uY
NmFIcIwqf4N6hbmpWm1NYaX+NQ0YOcVS+3pkXEehYC+CsQbF29sgZW IBr2li414sZ9rFYQNP6tn1JEAVQbUwgdfAP2NIza1+r2C+oPnlfRYZp0 Zzc=
;; Received 373 bytes from 192.82.134.30#53(y.arin.net) in 145 ms

2.186.218.216.in-addr.arpa. 86400 IN    PTR     mail.he.net.
;; Received 80 bytes from 216.218.132.2#53(ns3.he.net) in 248 ms

anurag@Anurags-MacBook-Pro ~> 
```

Subnet boundary consideration

216.218.186.1 - rDNS zone - 1.186.218.216.in-addr.arpa.
216.218.186.2 - rDNS zone - 2.186.218.216.in-addr.arpa.
216.218.186.3 - rDNS zone - 3.186.218.216.in-addr.arpa.
216.218.186.4 - rDNS zone - 4.186.218.216.in-addr.arpa.
216.218.186.255 - rDNS zone - 255.186.218.216.in-addr.arpa.

So x.186.218.216.in-addr.arpa. can cover 216.218.186.x/24

Subnet boundary consideration

216.218.186.0 - rDNS zone - 0.186.218.216.in-addr.arpa.
216.218.187.0 - rDNS zone - 0.187.218.216.in-addr.arpa.
216.218.188.0 - rDNS zone - 0.188.218.216.in-addr.arpa.
216.218.189.0 - rDNS zone - 0.189.218.216.in-addr.arpa.
216.218.255.0 - rDNS zone - 0.255.218.216.in-addr.arpa.

So 0.x.218.216.in-addr.arpa. can cover 216.218.x.0/24

Subnet boundary consideration

216.218.0.0 - rDNS zone - 0.0.218.216.in-addr.arpa.
216.219.0.0 - rDNS zone - 0.0.219.216.in-addr.arpa.
216.220.0.0 - rDNS zone - 0.0.220.216.in-addr.arpa.
216.221.0.0 - rDNS zone - 0.0.221.216.in-addr.arpa.
216.255.0.0 - rDNS zone - 0.0.255.216.in-addr.arpa.

So 0.0.218.216.in-addr.arpa. can cover 216.218.0.0/16

Subnet boundary consideration

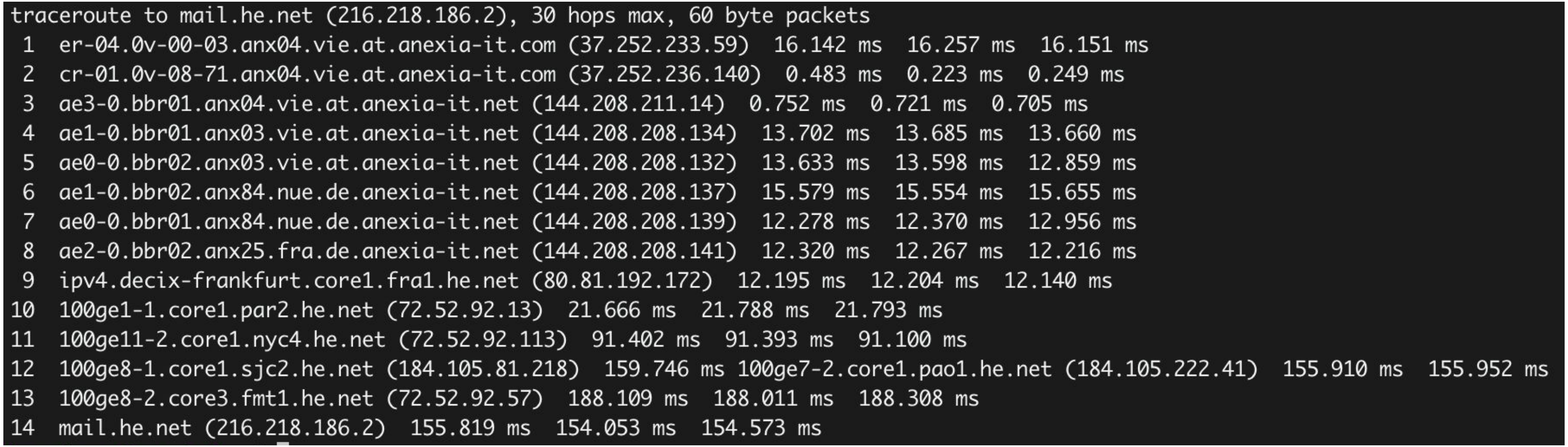
216.0.0.0 - rDNS zone - 0.0.0.216.in-addr.arpa.
217.0.0.0 - rDNS zone - 0.0.0.217.in-addr.arpa.
218.0.0.0 - rDNS zone - 0.0.0.218.in-addr.arpa.
219.0.0.0 - rDNS zone - 0.0.0.219.in-addr.arpa.
255.0.0.0 - rDNS zone - 0.0.0.255.in-addr.arpa.

So 0.0.0.216.in-addr.arpa. can cover 216.0.0.0/8

Misc about reverse DNS records

- Allocation depend on the 8 bit boundary in IPv4 i.e for /8 or /16 or /24.
- If allocation is outside of 8 bit boundary like e.g /22, it has to be broken into 4 /24 zones.
- For IPv6 the expansion can be very long. One can use dig to query before creating record to get expanded zone if not using automation
- Wildcard record often make sense to cover for your entire IP address space where customer has not required for address record.
- Often a good idea to add airport codes in rDNS of IPs of routers where they are located. Though in India railway station codes can be used. :)
- Also a good idea to add interface speed in the reverse DNS
- IXP operators can add AS-number of the members

Reverse DNS in action with trace



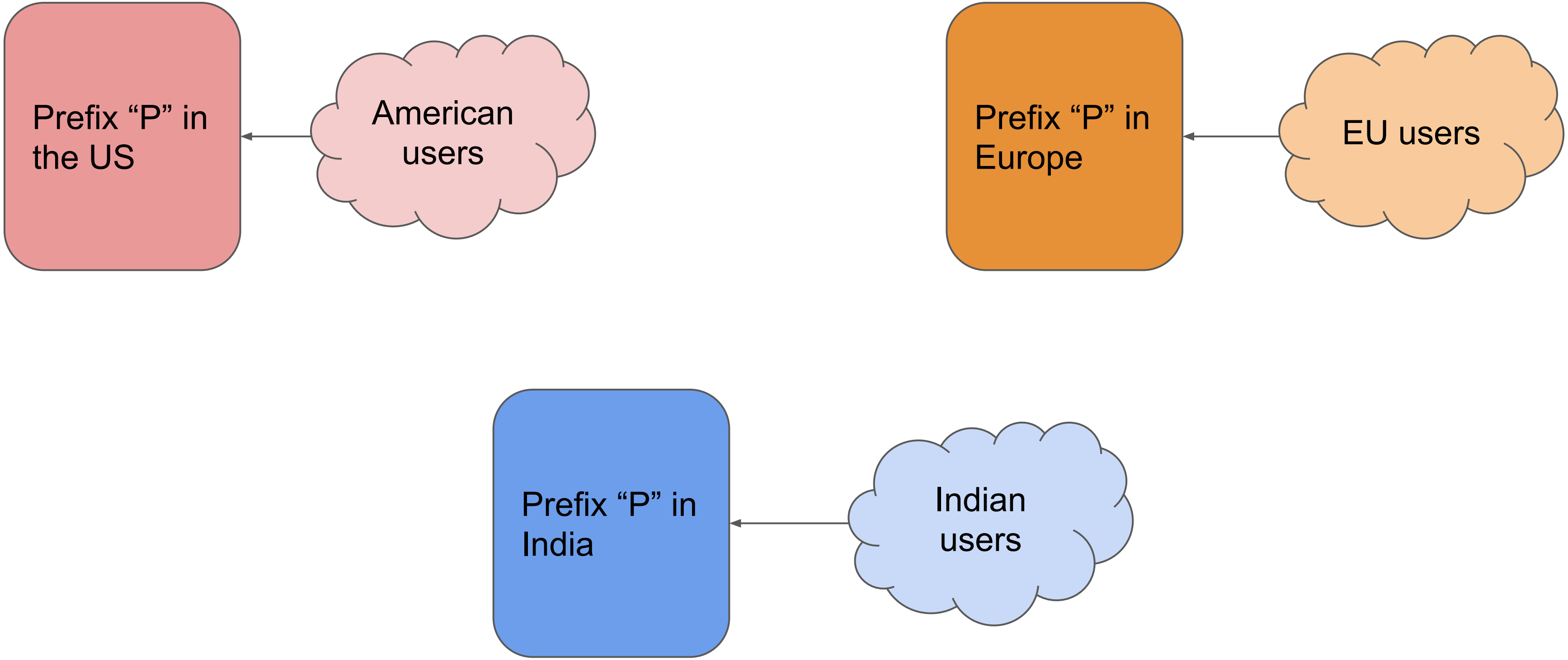
```
traceroute to mail.he.net (216.218.186.2), 30 hops max, 60 byte packets
 1  er-04.0v-00-03.anx04.vie.at.anexia-it.com (37.252.233.59)  16.142 ms  16.257 ms  16.151 ms
 2  cr-01.0v-08-71.anx04.vie.at.anexia-it.com (37.252.236.140)  0.483 ms  0.223 ms  0.249 ms
 3  ae3-0.bbr01.anx04.vie.at.anexia-it.net (144.208.211.14)  0.752 ms  0.721 ms  0.705 ms
 4  ae1-0.bbr01.anx03.vie.at.anexia-it.net (144.208.208.134)  13.702 ms  13.685 ms  13.660 ms
 5  ae0-0.bbr02.anx03.vie.at.anexia-it.net (144.208.208.132)  13.633 ms  13.598 ms  12.859 ms
 6  ae1-0.bbr02.anx84.nue.de.anexia-it.net (144.208.208.137)  15.579 ms  15.554 ms  15.655 ms
 7  ae0-0.bbr01.anx84.nue.de.anexia-it.net (144.208.208.139)  12.278 ms  12.370 ms  12.956 ms
 8  ae2-0.bbr02.anx25.fra.de.anexia-it.net (144.208.208.141)  12.320 ms  12.267 ms  12.216 ms
 9  ipv4.decix-frankfurt.core1.fra1.he.net (80.81.192.172)  12.195 ms  12.204 ms  12.140 ms
10  100ge1-1.core1.par2.he.net (72.52.92.13)  21.666 ms  21.788 ms  21.793 ms
11  100ge11-2.core1.nyc4.he.net (72.52.92.113)  91.402 ms  91.393 ms  91.100 ms
12  100ge8-1.core1.sjc2.he.net (184.105.81.218)  159.746 ms 100ge7-2.core1.pao1.he.net (184.105.222.41)  155.910 ms  155.952 ms
13  100ge8-2.core3.fmt1.he.net (72.52.92.57)  188.109 ms  188.011 ms  188.308 ms
14  mail.he.net (216.218.186.2)  155.819 ms  154.053 ms  154.573 ms
```

Commonly used DNS records

- A records: hostname -> IPv4 address mapping
- AAAA records: hostname -> IPv6 address mapping
- MX records: Tells where to send emails for the given domain
- PTR: IP -> hostname mapping
- NS records: Telling about the authoritative server for the delegation
- SOA: Lists master auth server, contact details of admin, serial number, zone refresh values etc
- CNAME: hostname -> alias to another hostname
- TXT: To publish text values used for various purposes (SPF, domain ownership verification, DKIM & more
- Glue records -> Similar to NS records but created at a level up and break cyclic dependency when resolving DNS. Created at domain registrar.

Anycast - where it works & where it fails?

Anycast - same pool across different locations



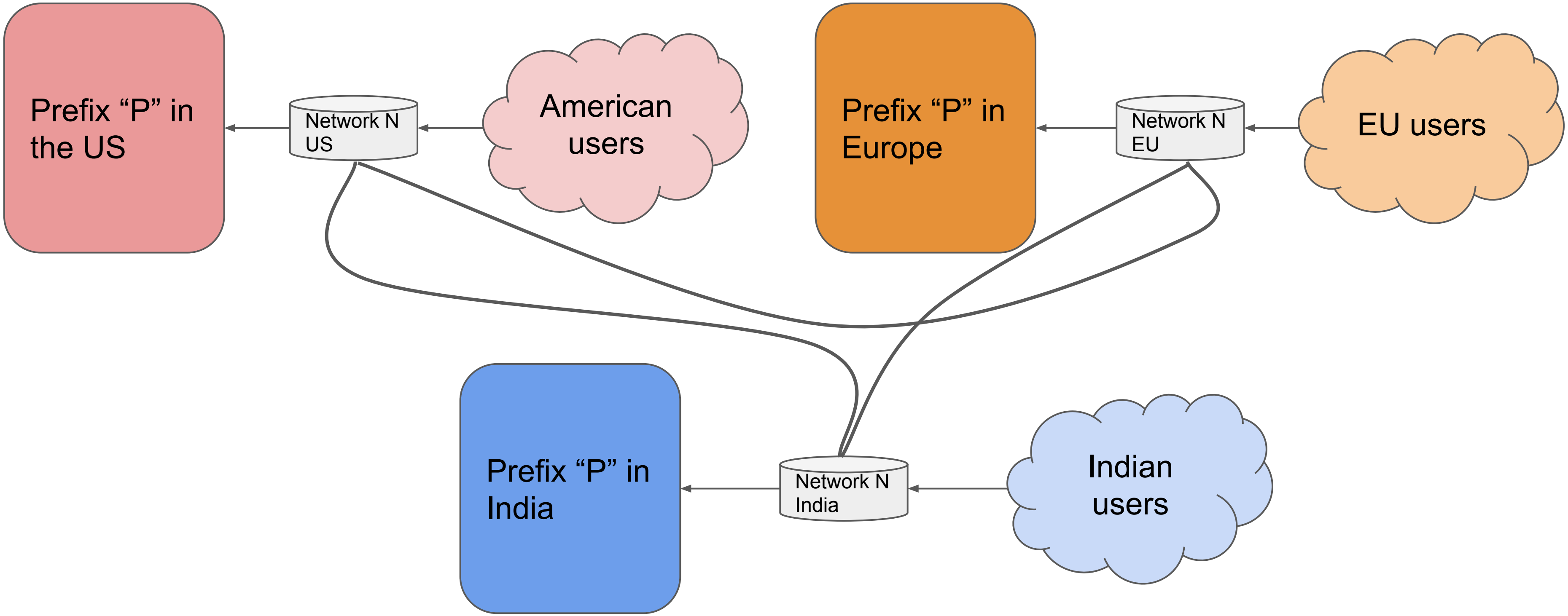
Reminder from Routing 101

1. Networks have highest localpref towards their customer routers

2. Second highest localpref towards their peers

3. Lowest localpref towards their upstream transit provider

Ideal anycast setup



Ideal anycast setup

1. Uses same (set) of transit providers at each location

2. High amounts of local peering

3. Transit provider has similar set of further transit & peering at each location

4. Impossible to achieve when building with nodes across India, US, Europe, South America, Africa etc due to availability and price considerations. E.g you can get Airtel in India but might not in the US, you can get HE in US, Europe but not India etc.

More realistic ideal anycast setup

1. Create two group of nodes - one to serve global traffic at key locations like Singapore, Palo Alto, Frankfurt, Amsterdam, etc. Other one to serve local locations only with peering.

2. Routes are announced to carefully picked transit providers at global locations but for local locations they are announced only to local peers at IX'es and often with no-export tag.

3. Avoid multilateral peering for anycast prefixes unless you know & trust all IX members at the route-server or have strong BGP community controls.

Broken anycast case study: nextdns.io

1. nextdns.io running DNS recursors offering firewall features & some advanced filtering

2. Using anycast with nodes across 53 locations including multiple locations in the US, Europe and Asia

3. Brings up their node in Pune, India & suddenly Google fiber user in the US complains of high latency & bad routing on reddit (post here)

4. This case study is based on visible data from traceroutes & BGP table

u/revelous -> Shares traceroute on Reddit

```
Host                                                          Loss%   Snt   Last   Avg  Best  Wrst StDev

 1. gateway                                                    0.0%    26    0.4   0.4   0.2   0.5   0.1
 2. <redacted>                                                 0.0%    26    2.5   2.3   1.2   2.8   0.3
 3. <redacted>.googlefiber.net                                 0.0%    26    2.7   2.8   1.9   3.6   0.3
 4. <redacted>.googlefiber.net                                 0.0%    26    7.8   7.9   6.8   8.9   0.4
 5. ix-ae-12-0.tcore1.dt8-dallas.as6453.net                    0.0%    26    7.6   7.6   6.7  11.2   0.8
 6. if-ae-37-3.tcore1.aeq-ashburn.as6453.net                   0.0%    26   37.8  39.4  37.7  43.7   1.5
 7. if-ae-2-2.tcore2.aeq-ashburn.as6453.net                   80.8%    26   37.7  38.3  37.7  38.8   0.6
 8. if-ae-12-2.tcore4.njy-newark.as6453.net                    3.8%    26   37.5  38.6  37.5  39.1   0.4
 9. if-ae-1-3.tcore3.njy-newark.as6453.net                     0.0%    26   38.8  38.4  37.3  42.4   0.9
10. 66.198.70.10                                               0.0%    26  238.0 238.1 237.0 238.8   0.3
11. ???
12. 14.142.23.68.static-vsnl.net.in                           76.0%    26  257.1 258.5 255.8 268.2   4.8
13. ???
14. dns1.nextdns.io                                            0.0%    25  288.9 289.2 288.2 293.7   1.1
```

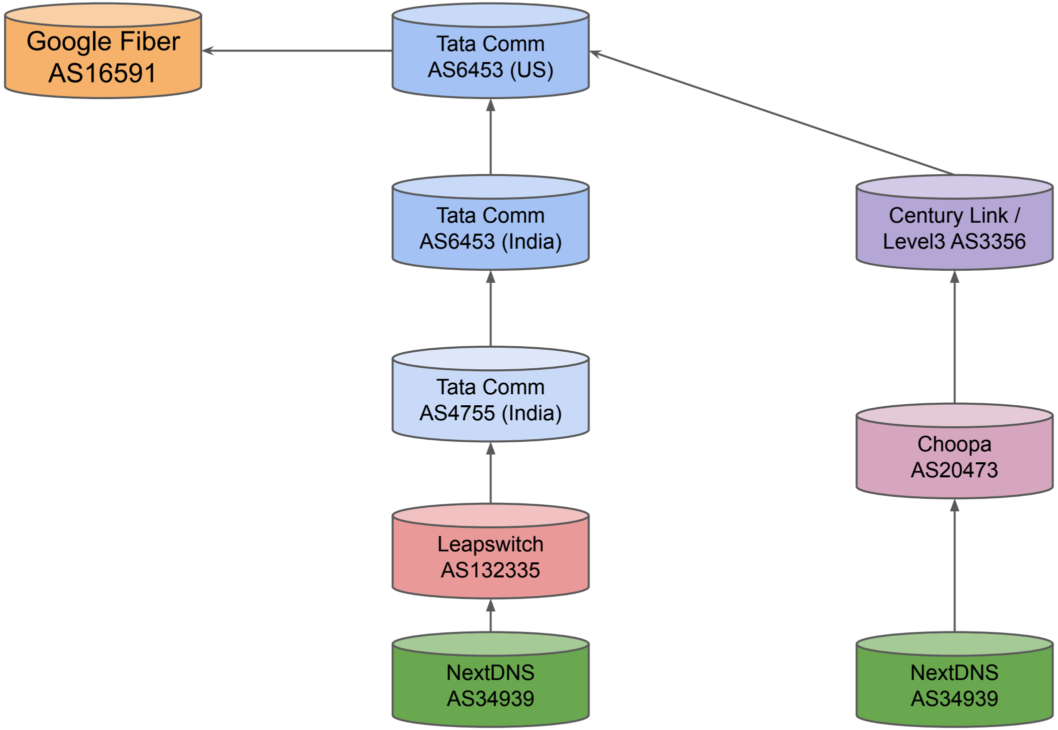
NextDNS setup in Pune

- Originated 45.90.28.0/24 from AS34939 to Leapswitch which further announced to Tata Comm AS4755 & that further announced to Tata Comm AS6453 - taking route to default free zone & visible to all other networks.

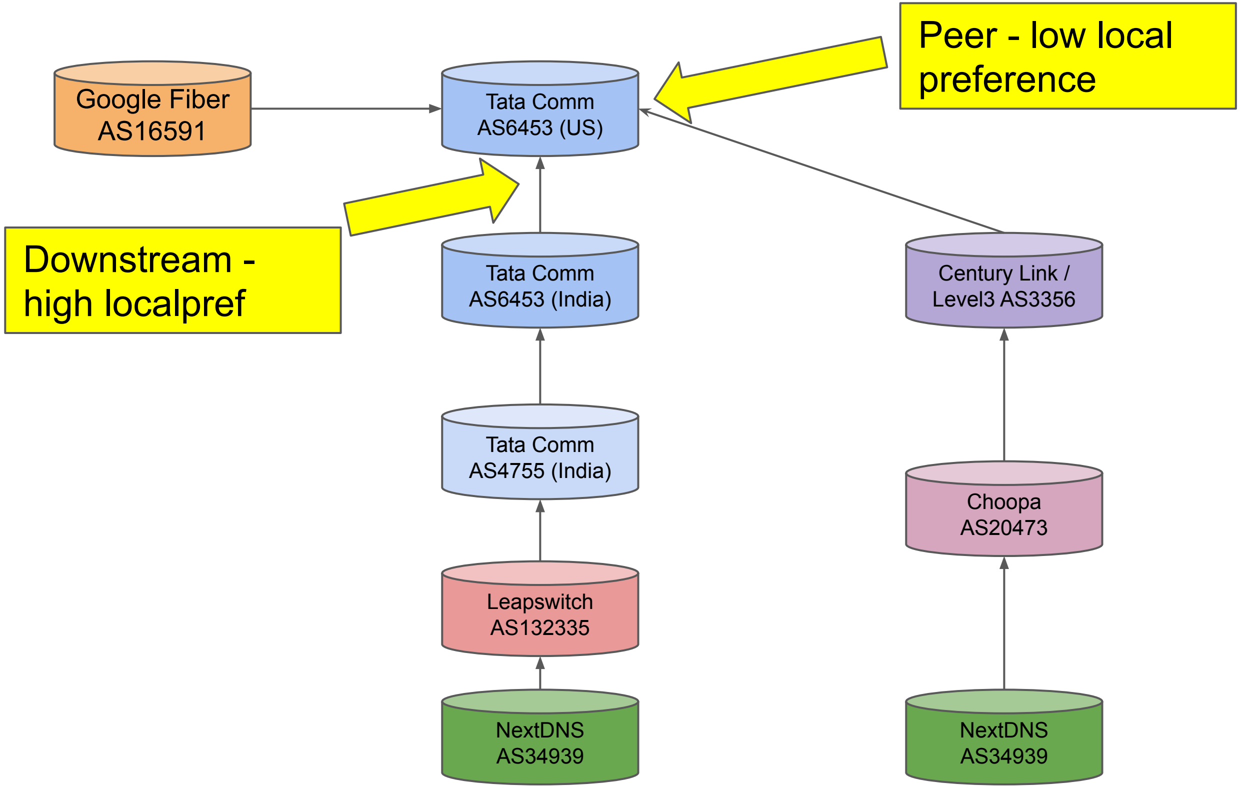
- By design NextDNS wasn’t peered to Google Fiber in the US and Google Fiber was relying on its transit provider (Tata Communications) to reach 45.90.28.0/24.

- Before Pune node came up, Tata Comm AS6453 routes in the US learnt route via its peer Century Link / Level3 AS3356 in New York which learnt it from its customer Choopa AS20473 which further learnt from its customer NextDNS AS34939.

NextDNS route announcement after Pune went up...



NextDNS route announcement after Pune went up...



Peer - low local preference

Downstream - high localpref

After NextDNS stop announcement in Pune

```
Traceroute to 45.90.28.0 (45.90.28.0), 48 byte packets

1  192.168.1.1  9.179ms  2.829ms  2.767ms
2  10.26.1.66  4.6ms  3.823ms  3.775ms
3  23.255.225.10  23-255-225-10.googlefiber.net AS16591  3.652ms  3.699ms  4.282ms
4  23.255.224.195  23-255-224-195.mci.googlefiber.net AS16591  4.767ms  4.454ms  4.87ms
5  *  *  *
6  23.255.225.114  23-255-225-114.googlefiber.net  AS16591  14.999ms  13.249ms  13.698ms
7  206.82.141.198  ix-ae-60-0.tcore1.ct8-chicago.as6453.net  AS6453  13.058ms  13.091ms  13.129ms
8  4.68.110.193  lag-20.ear1.Chicago2.Level3.net  AS3356  14.4ms  13.984ms  14.297ms
9  *  4.69.142.105  ae-1-3507.ear4.Chicago2.Level3.net AS3356  20ms  19.957ms
10  4.14.14.158  CHOOPA-LLC.ear4.Chicago2.Level3.net AS3356  19.768ms  19.94ms  40.116ms
11  *  *  *
12  *  *  *
13  *  *  *
14  45.90.28.0  dns1.nextdns.io  AS34939  14.5ms  14.154ms  14.16ms
```

Note:

1. It’s my guess that announcement has stop in Pune based on traceroutes. NextDNS never confirmed that.
2. Latest trace taken from RIPE Atlas: https://atlas.ripe.net/measurements/28266797/#probes

Possible workarounds to make such setup work

1. NextDNS could have put node peered to IXPs in India and restricted announcement to Indian peers only.

2. No-export could further be used to limit BGP announcement at the IXPs.

3. They might have taken BGP community via Leapswitch via Tata Comm to limit BGP announcement with AS4755 & its Indian peers (like Airtel AS9498, Jio AS55836) only.

Setup Anycast DNS resolver

DNS resolver for own customers

1. For resolver facing own customers - global BGP routing optimisation issue doesn't comes into picture. Your own backbone knows best paths to nearest servers.

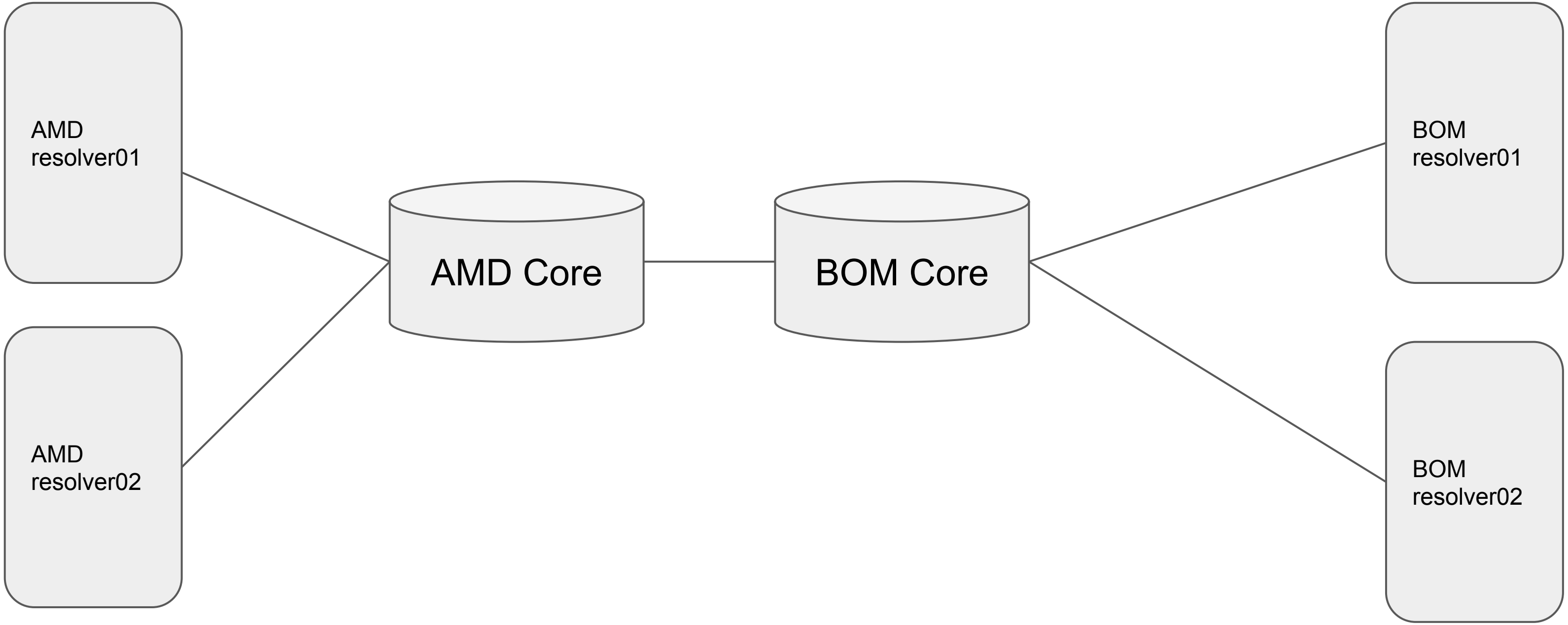
2. Helps in simplifying IP address used in the network. So e.g instead of 2 IPs in Ahmedabad, 2 in Surat, 2 in Rajkot, 2 in Mumbai one can use 2 IPs across all the nodes.

3. Anycast IPs (/32 in IPv4 and /128s in IPv6) can be originated via BGP daemon running on the server itself. One can use FRR to inject route into iBGP.

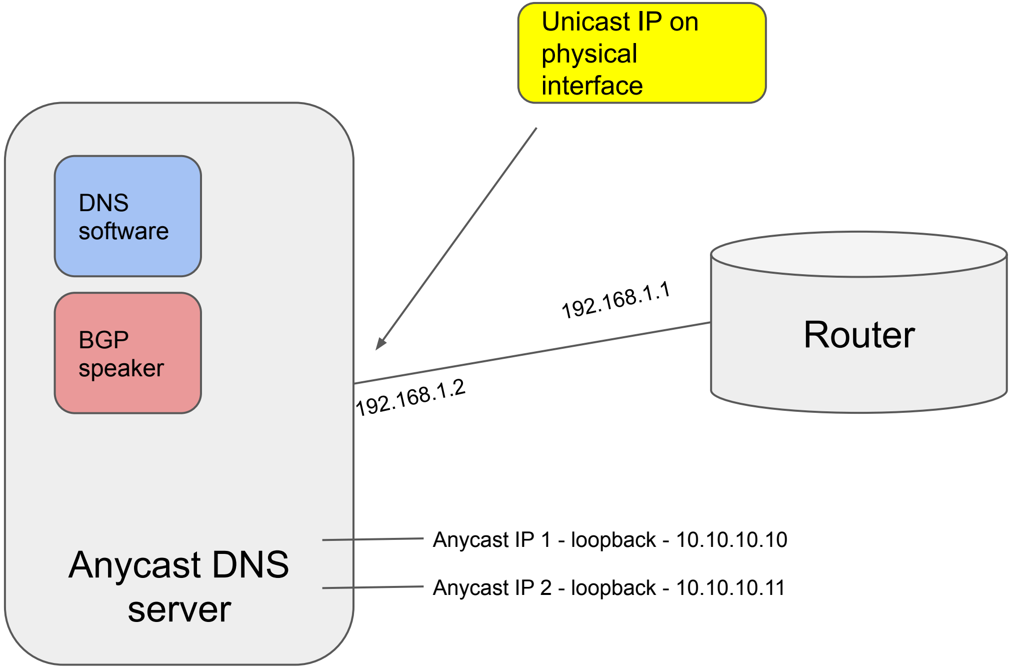
4. For maintenance, BGP announcement can be pulled off and traffic goes to the other nodes.

5. If server hardware fails, BGP session breaks & anycast takes care of moving traffic away

Anycast DNS recursor



Anycast DNS server


Unicast IP on physical interface
DNS software
BGP speaker
192.168.1.1
Router
192.168.1.2
Anycast IP 1 - loopback - 10.10.10.10
Anycast IP 2 - loopback - 10.10.10.11
Anycast DNS server

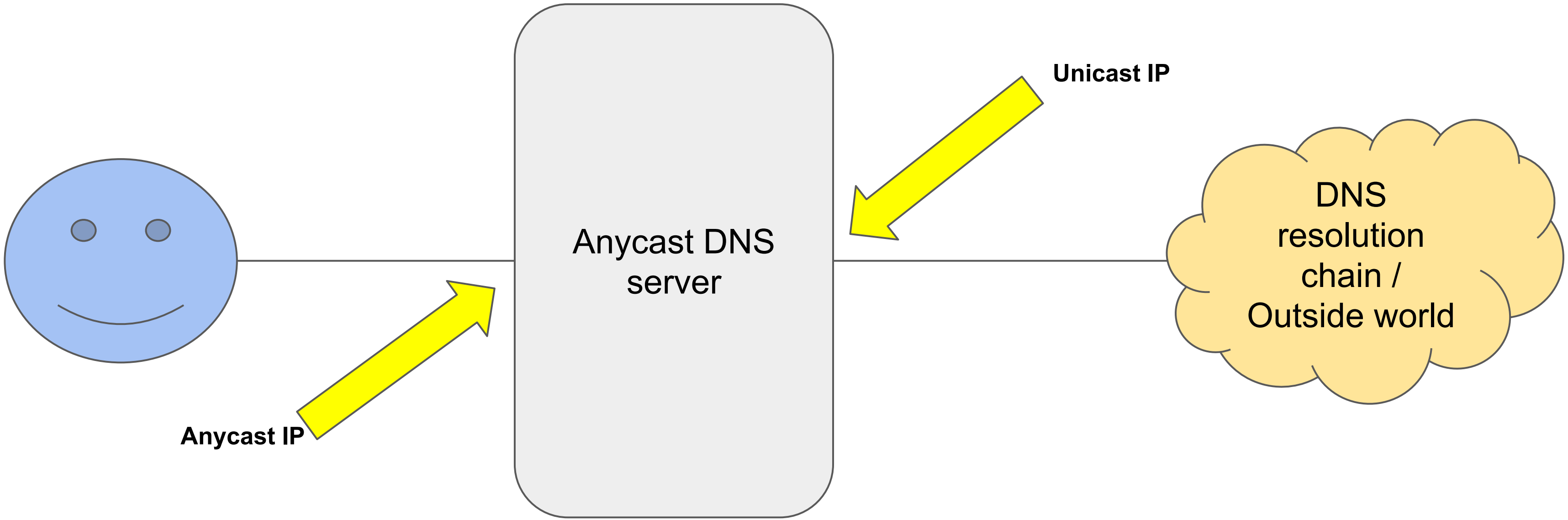
Why unicast IP needed in anycast setup?

- Needed for management interface of the server so that it can be accessed for management

- Used for running dynamic routing announcement (like with BGP) to the router

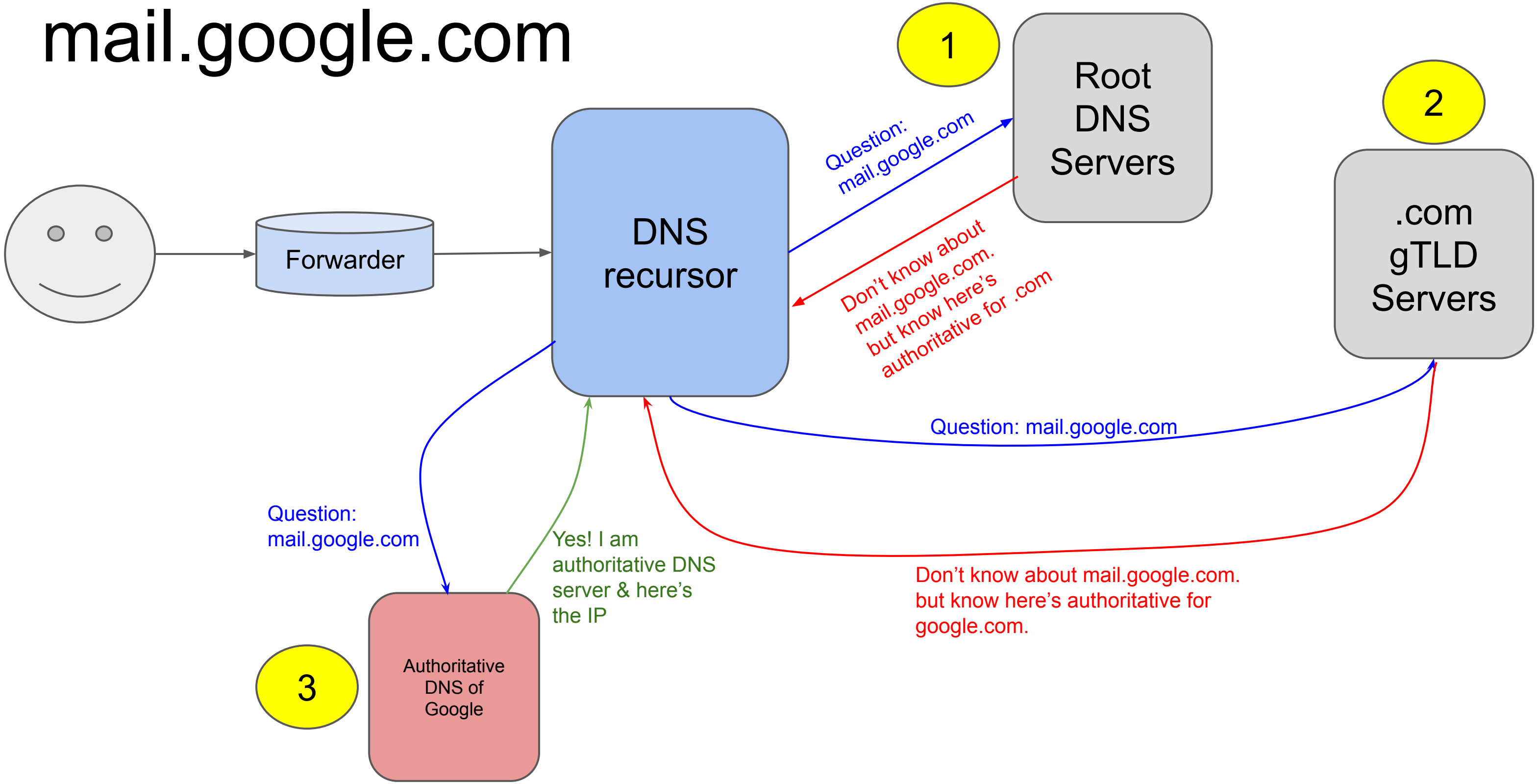
- Clients contact DNS server on anycast IP but server sends queries out using unicast IP (so return can be guaranteed)

DNS resolution communication



EDNS client subnet

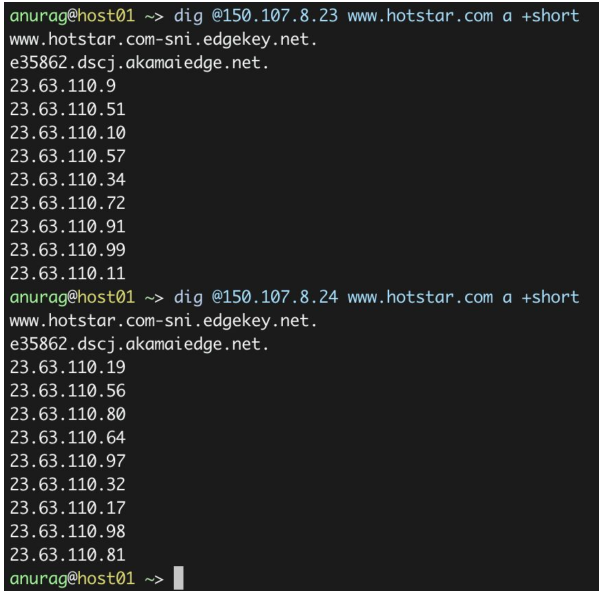
Reminder: Typical DNS resolution chain -> mail.google.com


1
Root DNS Servers
2
.com gTLD Servers
Forwarder
DNS recursor
Question: mail.google.com
Don't know about mail.google.com. but know here's authoritative for .com
Question: mail.google.com
Don't know about mail.google.com. but know here's authoritative for google.com.
Question: mail.google.com
Yes! I am authoritative DNS server & here's the IP
3
Authoritative DNS of Google

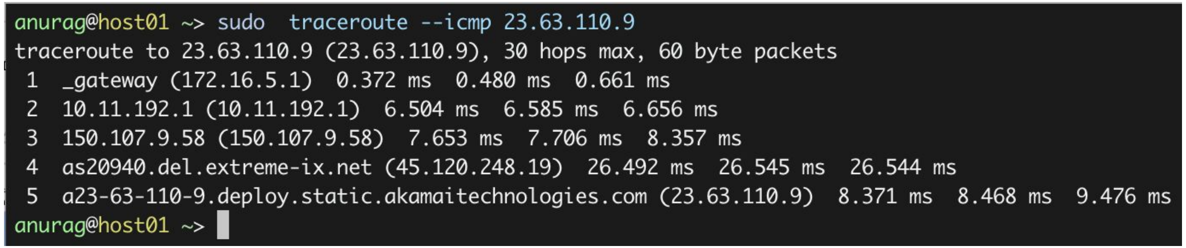
EDNS client subnet <- Understanding the problem

- Authoritative DNS server only “see” the DNS resolver and not the end user
- Some CDN providers use DNS resolver’s IP to map user to “nearest” CDN server
- If end user uses a server far away from them, CDN auth DNS can reply back with a server far off from the user

Resolving & trace to www.hotstar.com using my ISPs DNS resolver



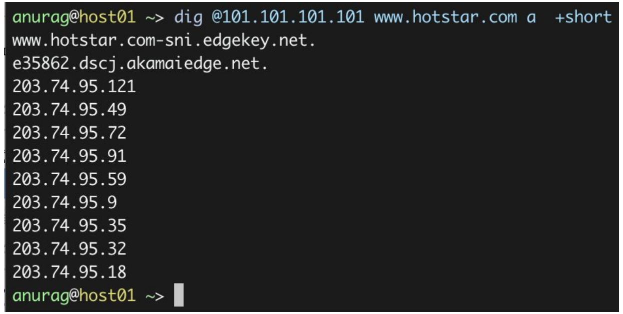
```
anurag@host01 ~> dig @150.107.8.23 www.hotstar.com a +short
www.hotstar.com-sni.edgekey.net.
e35862.dscj.akamaiedge.net.
23.63.110.9
23.63.110.51
23.63.110.10
23.63.110.57
23.63.110.34
23.63.110.72
23.63.110.91
23.63.110.99
23.63.110.11
anurag@host01 ~> dig @150.107.8.24 www.hotstar.com a +short
www.hotstar.com-sni.edgekey.net.
e35862.dscj.akamaiedge.net.
23.63.110.19
23.63.110.56
23.63.110.80
23.63.110.64
23.63.110.97
23.63.110.32
23.63.110.17
23.63.110.98
23.63.110.81
anurag@host01 ~>
```



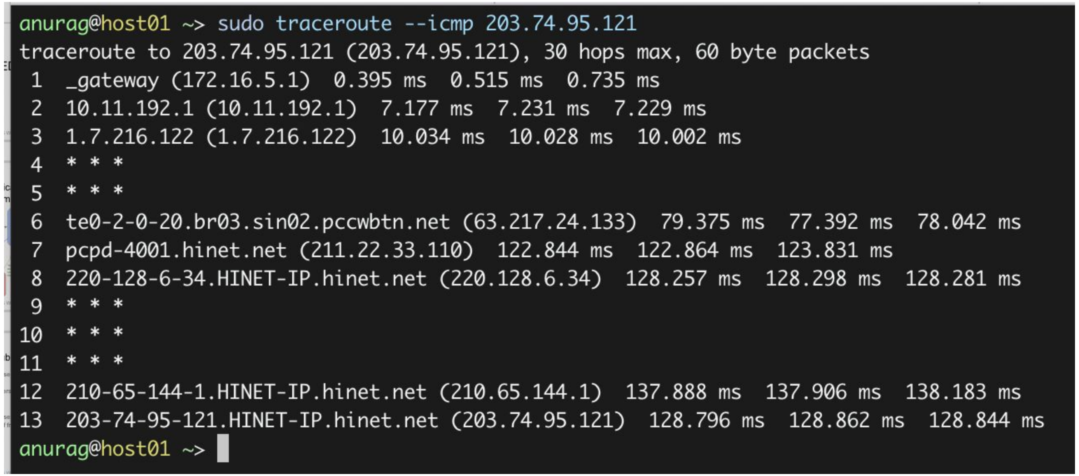
```
anurag@host01 ~> sudo  traceroute --icmp 23.63.110.9
traceroute to 23.63.110.9 (23.63.110.9), 30 hops max, 60 byte packets
 1  _gateway (172.16.5.1)  0.372 ms  0.480 ms  0.661 ms
 2  10.11.192.1 (10.11.192.1)  6.504 ms  6.585 ms  6.656 ms
 3  150.107.9.58 (150.107.9.58)  7.653 ms  7.706 ms  8.357 ms
 4  as20940.del.extreme-ix.net (45.120.248.19)  26.492 ms  26.545 ms  26.544 ms
 5  a23-63-110-9.deploy.static.akamaitechnologies.com (23.63.110.9)  8.371 ms  8.468 ms  9.476 ms
anurag@host01 ~>
```

Hitting CDN node (located at peering IX) here with 8.3ms

Resolving & trace to www.hotstar.com using my Taiwan’s 101.101.101.101



```
anurag@host01 ~> dig @101.101.101.101 www.hotstar.com a  +short
www.hotstar.com-sni.edgekey.net.
e35862.dscj.akamaiedge.net.
203.74.95.121
203.74.95.49
203.74.95.72
203.74.95.91
203.74.95.59
203.74.95.9
203.74.95.35
203.74.95.32
203.74.95.18
anurag@host01 ~>
```



```
anurag@host01 ~> sudo traceroute --icmp 203.74.95.121
traceroute to 203.74.95.121 (203.74.95.121), 30 hops max, 60 byte packets
 1  _gateway (172.16.5.1)  0.395 ms  0.515 ms  0.735 ms
 2  10.11.192.1 (10.11.192.1)  7.177 ms  7.231 ms  7.229 ms
 3  1.7.216.122 (1.7.216.122)  10.034 ms  10.028 ms  10.002 ms
 4  * * *
 5  * * *
 6  te0-2-0-20.br03.sin02.pccwbtn.net (63.217.24.133)  79.375 ms  77.392 ms  78.042 ms
 7  pcpd-4001.hinet.net (211.22.33.110)  122.844 ms  122.864 ms  123.831 ms
 8  220-128-6-34.HINET-IP.hinet.net (220.128.6.34)  128.257 ms  128.298 ms  128.281 ms
 9  * * *
10  * * *
11  * * *
12  210-65-144-1.HINET-IP.hinet.net (210.65.144.1)  137.888 ms  137.906 ms  138.183 ms
13  203-74-95-121.HINET-IP.hinet.net (203.74.95.121)  128.796 ms  128.862 ms  128.844 ms
anurag@host01 ~>
```

Hitting CDN node (located in Taiwan) here which is 128ms away!

101 DNS resolver - https://101.101.101.101

Client subnet in EDNS - solution

1. DNS resolver passes /24 equivalent of client to the authoritative DNS resolver

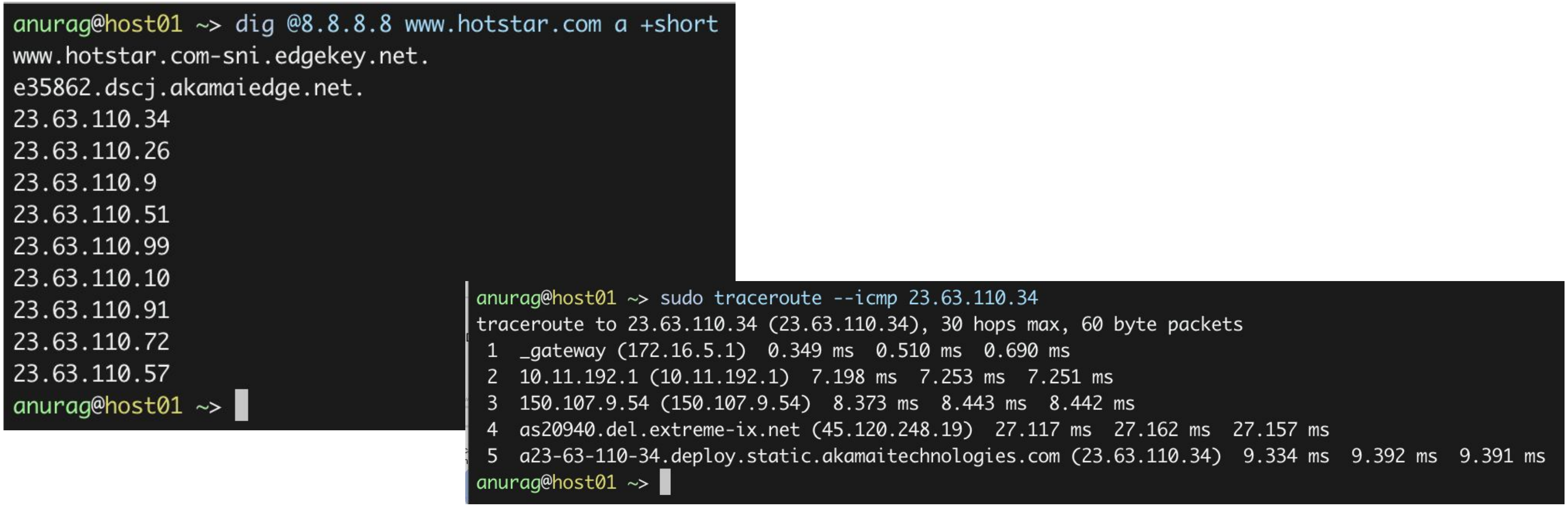
2. Was done in collaboration of popular DNS resolvers like OpenDNS, Google DNS, CDNs like Akamai etc.

3. (Based on experience) works well between trusted resolvers & authoritative servers only & not by any random DNS resolver.

4. Considered bad for privacy as /24 can leak “*too much information*” to authoritative DNS servers.

5. Some support it like Google 8.8.8.8 and some intentionally do not support it like Cloudflare’s 1.1.1.1

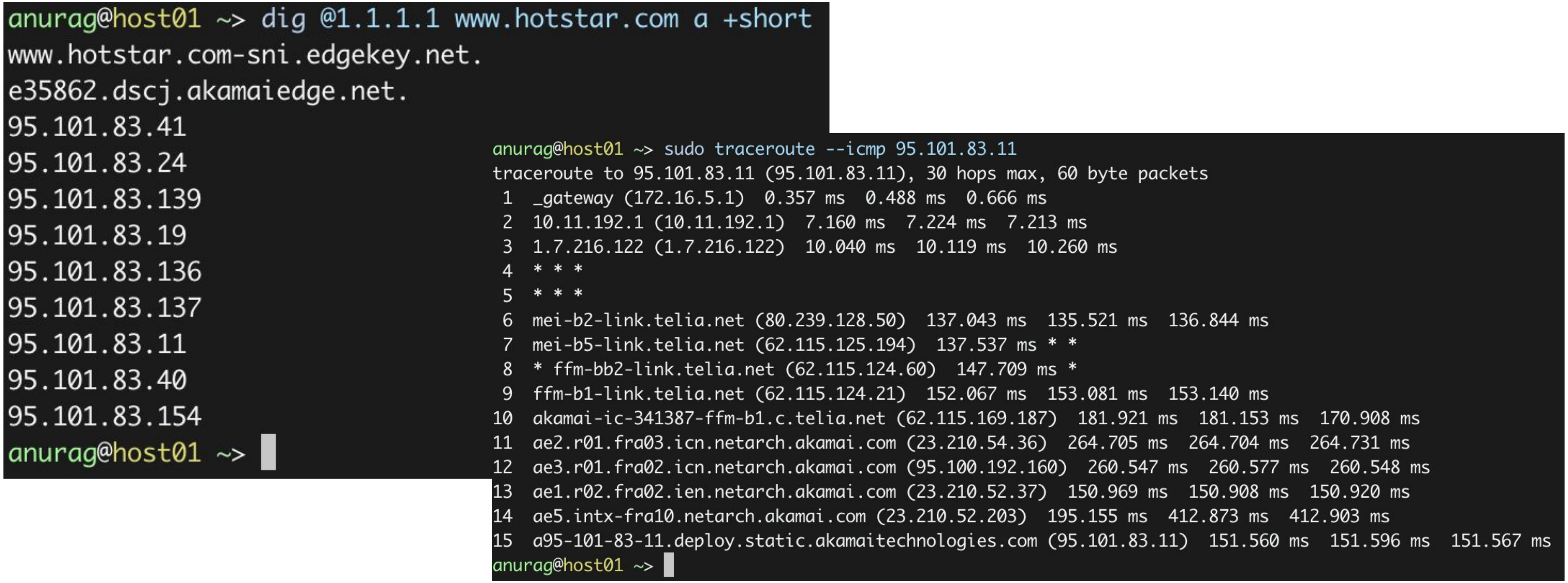
Google DNS Vs Cloudflare for www.Hotstar.com



```
anurag@host01 ~> dig @8.8.8.8 www.hotstar.com a +short
www.hotstar.com-sni.edgekey.net.
e35862.dscj.akamaiedge.net.
23.63.110.34
23.63.110.26
23.63.110.9
23.63.110.51
23.63.110.99
23.63.110.10
23.63.110.91
23.63.110.72
23.63.110.57
anurag@host01 ~>
```

```
anurag@host01 ~> sudo traceroute --icmp 23.63.110.34
traceroute to 23.63.110.34 (23.63.110.34), 30 hops max, 60 byte packets
 1  _gateway (172.16.5.1)  0.349 ms  0.510 ms  0.690 ms
 2  10.11.192.1 (10.11.192.1)  7.198 ms  7.253 ms  7.251 ms
 3  150.107.9.54 (150.107.9.54)  8.373 ms  8.443 ms  8.442 ms
 4  as20940.del.extreme-ix.net (45.120.248.19)  27.117 ms  27.162 ms  27.157 ms
 5  a23-63-110-34.deploy.static.akamaitechnologies.com (23.63.110.34)  9.334 ms  9.392 ms  9.391 ms
anurag@host01 ~>
```

Google DNS Vs Cloudflare for www.Hotstar.com



```
anurag@host01 ~> dig @1.1.1.1 www.hotstar.com a +short
www.hotstar.com-sni.edgekey.net.
e35862.dscj.akamaiedge.net.
95.101.83.41
95.101.83.24
95.101.83.139
95.101.83.19
95.101.83.136
95.101.83.137
95.101.83.11
95.101.83.40
95.101.83.154
anurag@host01 ~>
```

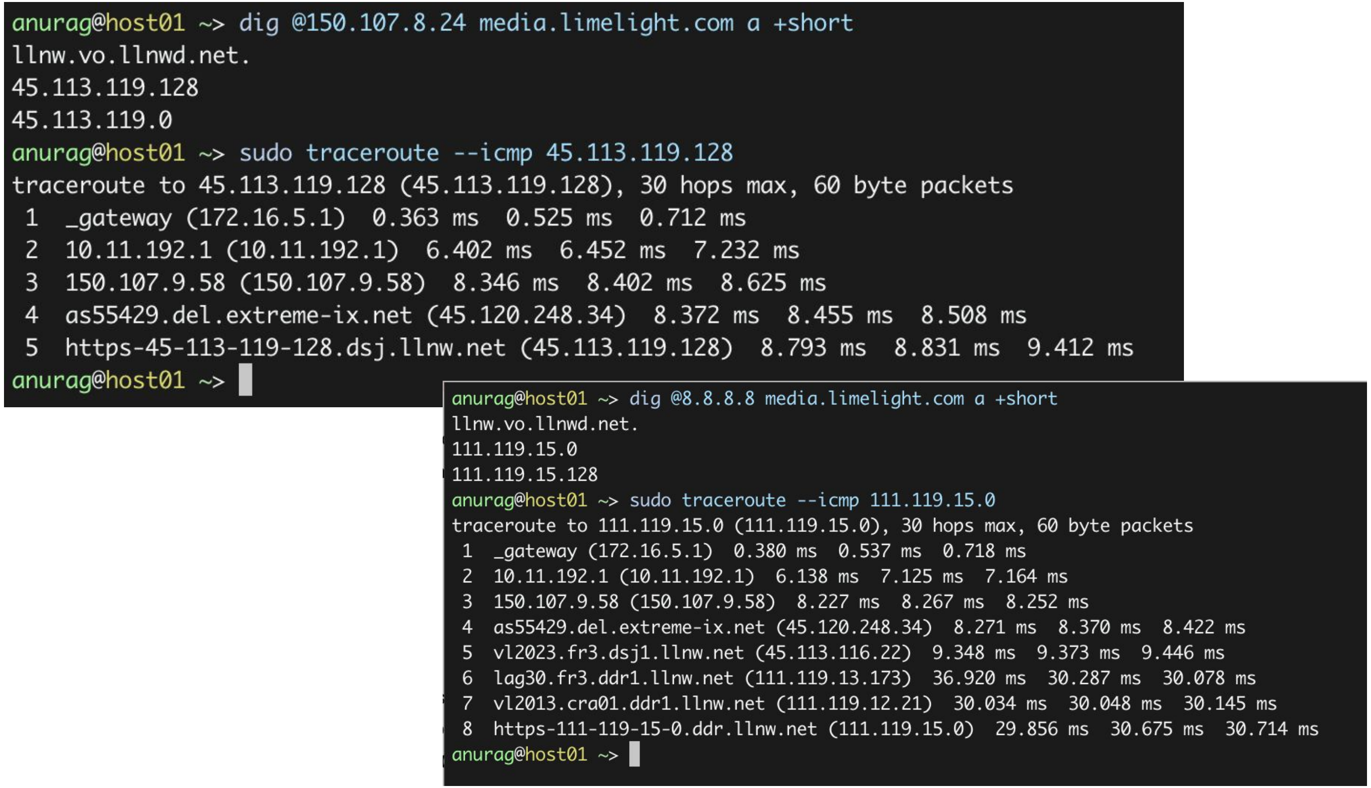
```
anurag@host01 ~> sudo traceroute --icmp 95.101.83.11
traceroute to 95.101.83.11 (95.101.83.11), 30 hops max, 60 byte packets
 1  _gateway (172.16.5.1)  0.357 ms  0.488 ms  0.666 ms
 2  10.11.192.1 (10.11.192.1)  7.160 ms  7.224 ms  7.213 ms
 3  1.7.216.122 (1.7.216.122)  10.040 ms  10.119 ms  10.260 ms
 4  * * *
 5  * * *
 6  mei-b2-link.telia.net (80.239.128.50)  137.043 ms  135.521 ms  136.844 ms
 7  mei-b5-link.telia.net (62.115.125.194)  137.537 ms * *
 8  * ffm-bb2-link.telia.net (62.115.124.60)  147.709 ms *
 9  ffm-b1-link.telia.net (62.115.124.21)  152.067 ms  153.081 ms  153.140 ms
10  akamai-ic-341387-ffm-b1.c.telia.net (62.115.169.187)  181.921 ms  181.153 ms  170.908 ms
11  ae2.r01.fra03.icn.netarch.akamai.com (23.210.54.36)  264.705 ms  264.704 ms  264.731 ms
12  ae3.r01.fra02.icn.netarch.akamai.com (95.100.192.160)  260.547 ms  260.577 ms  260.548 ms
13  ae1.r02.fra02.ien.netarch.akamai.com (23.210.52.37)  150.969 ms  150.908 ms  150.920 ms
14  ae5.intx-fra10.netarch.akamai.com (23.210.52.203)  195.155 ms  412.873 ms  412.903 ms
15  a95-101-83-11.deploy.static.akamaitechnologies.com (95.101.83.11)  151.560 ms  151.596 ms  151.567 ms
anurag@host01 ~>
```

So I should just use 8.8.8.8 instead of 1.1.1.1?

So I should just use 8.8.8.8 instead of 1.1.1.1?

NO!

Limelight on ISPs local DNS Vs Google



```
anurag@host01 ~> dig @150.107.8.24 media.limelight.com a +short
llnw.vo.llnwd.net.
45.113.119.128
45.113.119.0
anurag@host01 ~> sudo traceroute --icmp 45.113.119.128
traceroute to 45.113.119.128 (45.113.119.128), 30 hops max, 60 byte packets
 1  _gateway (172.16.5.1)  0.363 ms  0.525 ms  0.712 ms
 2  10.11.192.1 (10.11.192.1)  6.402 ms  6.452 ms  7.232 ms
 3  150.107.9.58 (150.107.9.58)  8.346 ms  8.402 ms  8.625 ms
 4  as55429.del.extreme-ix.net (45.120.248.34)  8.372 ms  8.455 ms  8.508 ms
 5  https-45-113-119-128.dsj.llnw.net (45.113.119.128)  8.793 ms  8.831 ms  9.412 ms
anurag@host01 ~>
```

```
anurag@host01 ~> dig @8.8.8.8 media.limelight.com a +short
llnw.vo.llnwd.net.
111.119.15.0
111.119.15.128
anurag@host01 ~> sudo traceroute --icmp 111.119.15.0
traceroute to 111.119.15.0 (111.119.15.0), 30 hops max, 60 byte packets
 1  _gateway (172.16.5.1)  0.380 ms  0.537 ms  0.718 ms
 2  10.11.192.1 (10.11.192.1)  6.138 ms  7.125 ms  7.164 ms
 3  150.107.9.58 (150.107.9.58)  8.227 ms  8.267 ms  8.252 ms
 4  as55429.del.extreme-ix.net (45.120.248.34)  8.271 ms  8.370 ms  8.422 ms
 5  vl2023.fr3.dsj1.llnw.net (45.113.116.22)  9.348 ms  9.373 ms  9.446 ms
 6  lag30.fr3.ddr1.llnw.net (111.119.13.173)  36.920 ms  30.287 ms  30.078 ms
 7  vl2013.cra01.ddr1.llnw.net (111.119.12.21)  30.034 ms  30.048 ms  30.145 ms
 8  https-111-119-15-0.ddr.llnw.net (111.119.15.0)  29.856 ms  30.675 ms  30.714 ms
anurag@host01 ~>
```

Good practice for CDN traffic optimisation

1. Offer highly available, superfast DNS resolvers to your end customers so that they do not move to outside DNS resolvers.

2. Try picking cool simple (anycast) IP addresses & make ground teams remember that instead of putting 1.1.1.1 everywhere.

3. Ensure that IP prefix covering DNS resolver’s unicast is announced to CDN players (if have CDNs within your network).

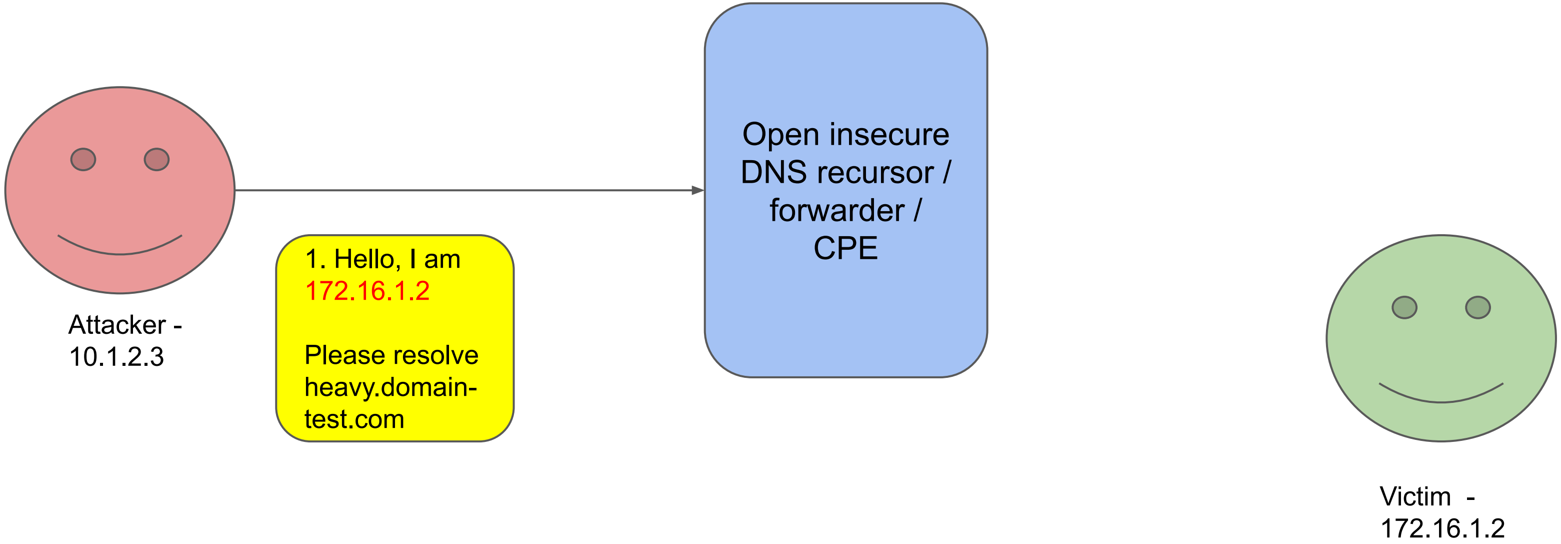
4. Run full fledged DNS resolver and not just a DNS forwarder towards 1.1.1.1 or even 8.8.8.8.

DNS Amplification attacks

Some characteristics of DNS (and NTP)

1. Based on UDP - connectionless and hence no TCP handshake is needed
2. For small light questions (based on size), replies can be very large (upto 70 times)
3. Large number of networks miss anti-spoofing protection
4. Many networks run DNS recursors open to the world and without any rate limitation

DNS amplification attacks - how it happens?



Attacker - 10.1.2.3

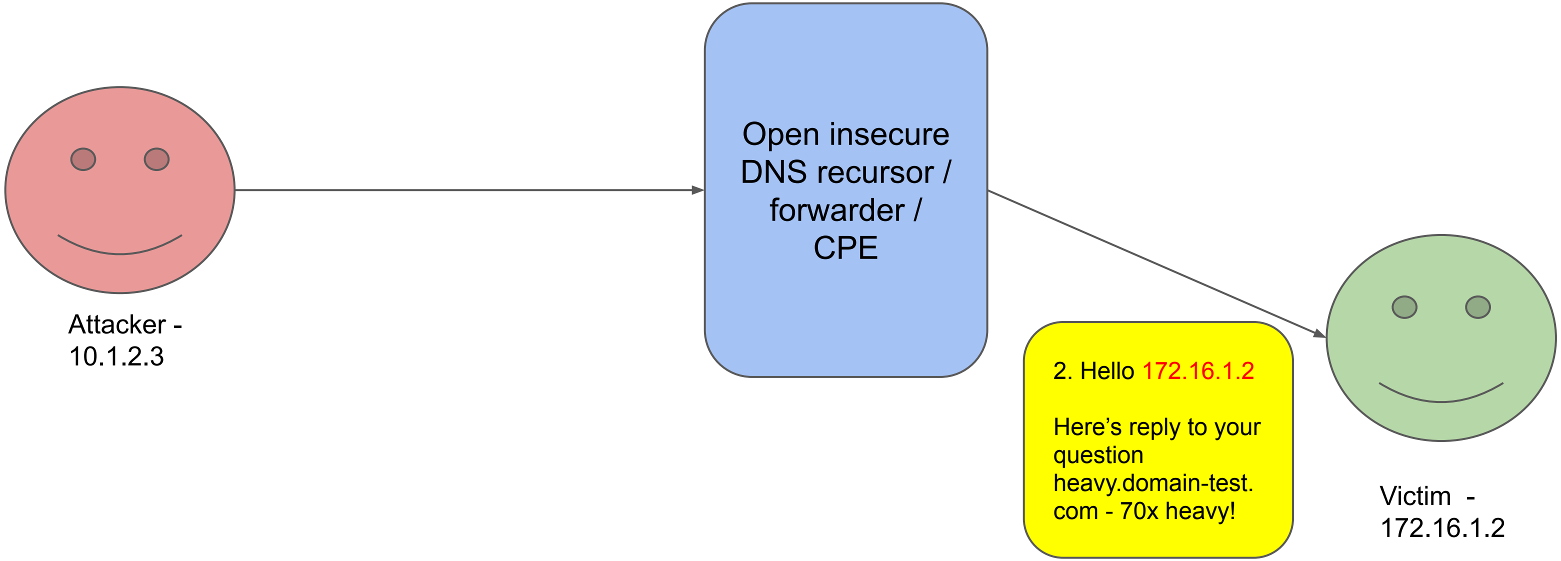
1. Hello, I am 172.16.1.2

Please resolve heavy.domain-test.com

Open insecure DNS recursor / forwarder / CPE

Victim - 172.16.1.2

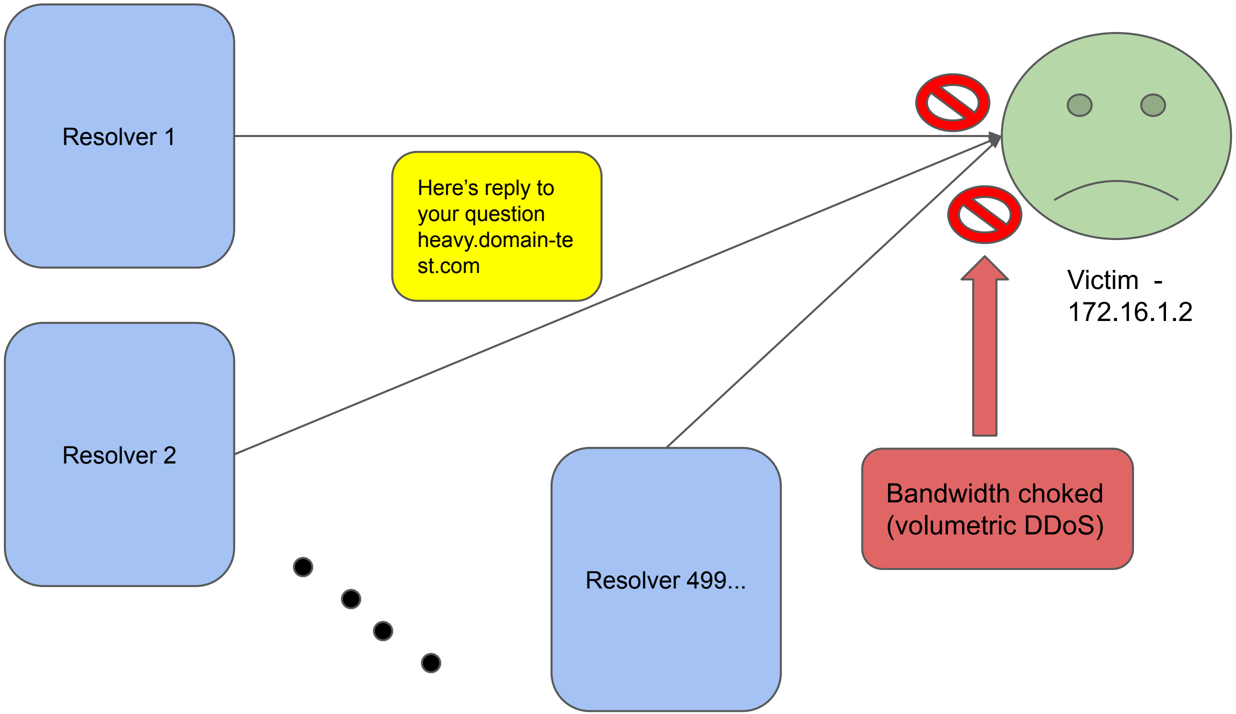
DNS amplification attacks - how it happens?



2. Hello 172.16.1.2

Here's reply to your question heavy.domain-test.com - 70x heavy!

DNS amplification attacks - how it happens?



Resolver 1
Here's reply to your question heavy.domain-test.com
Victim - 172.16.1.2
Resolver 2
Resolver 499...
Bandwidth choked (volumetric DDoS)

The 2013 Spamhaus attack...300Gbps

Excellent write up here:
https://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet/

Good security practices

1. Do not run open DNS recursors. Only allow your client (and downstream) IPs to query your resolver.

2. Follow bcp38 - Ingress filtering

3. Put anti-spoofing protection using uRPF in strict mode across your edge routers and loose mode across core routers.

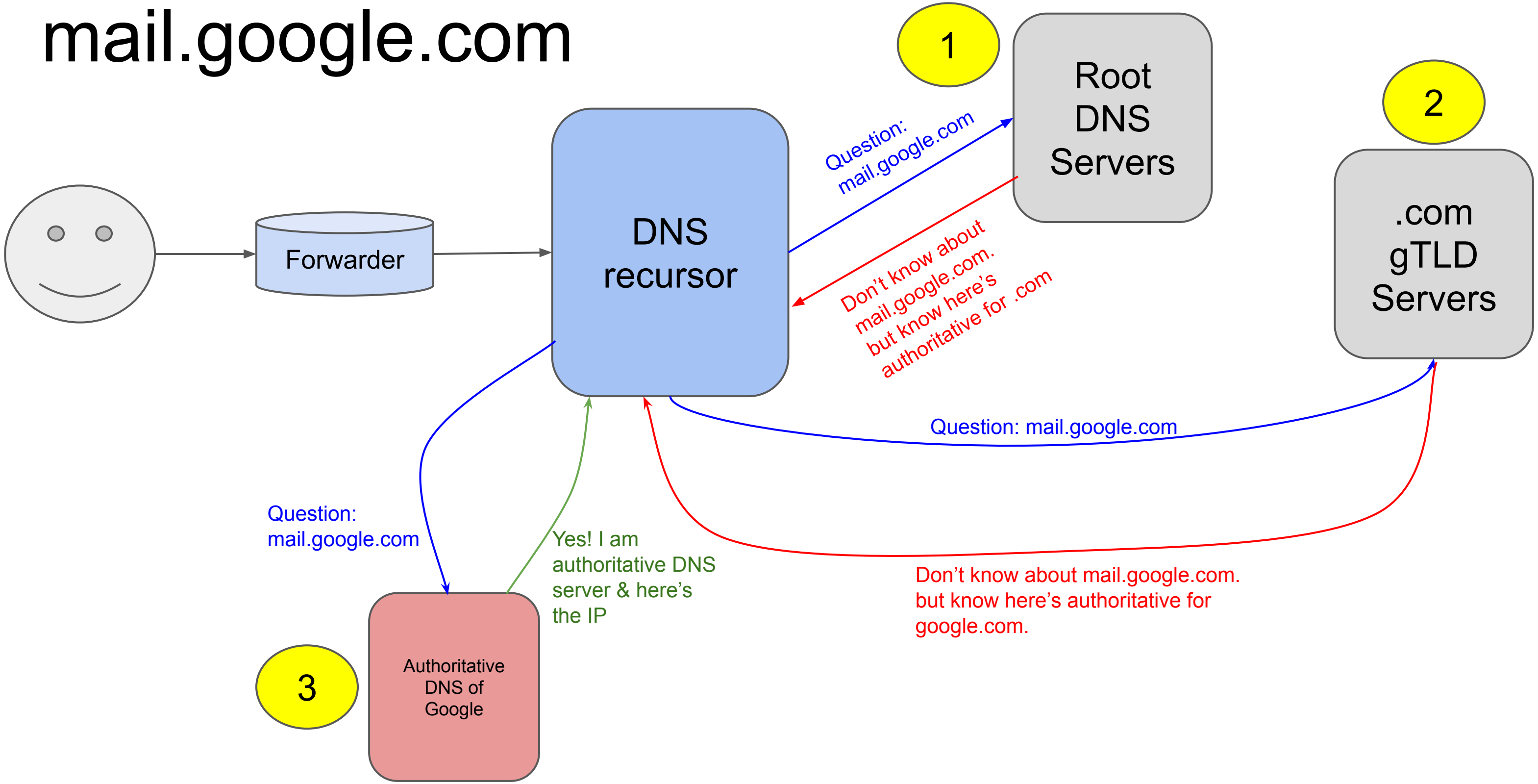
4. Have setup of BGP blackholing with your upstream provider via BGP communities

5. Have some sort of automated filtering mechanism in place if ever get targeted by DDoS from DNS amplification attack (e.g Opensource Fastnetmon)

6. Watch Out for your address space for any possible open DNS resolvers or CPE listening on WAN interface instead of (only) LAN interface.

DNSSEC

Reminder: Typical DNS resolution chain -> mail.google.com



1
Root DNS Servers
2
.com gTLD Servers
Forwarder
DNS recursor
Question: mail.google.com
Don't know about mail.google.com. but know here's authoritative for .com
Question: mail.google.com
Don't know about mail.google.com. but know here's authoritative for google.com.
Question: mail.google.com
Yes! I am authoritative DNS server & here's the IP
3
Authoritative DNS of Google

DNSSEC

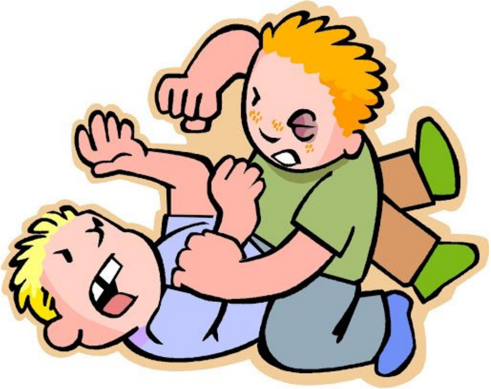
- Communication between DNS resolvers & root DNS servers, TLD servers, zone auth servers by default by default isn’t validated

- Any attacker sitting in middle can give fake reply & resolver won’t know

- DNSSEC solves this problem by using using public-private key cryptography to “sign” zones.

- Signing has to happen at every part of the chain.

- If you have domain name, consider signing the zone!



DNS over TLS & DoH

DNS over TLS

- Idea is to secure communication between client & DNS resolvers using encryption
- Uses port 853
- Can ensure integrity as well as privacy when communication with a trusted resolver.
- ISPs as well as many external public resolvers offer DNS over TLS.
- Some newer operating systems like Android 9 support it out of box.
- Some of DNS software vendors give option to offer DoT support for ISPs (do it!)

DNS over HTTPS

- (Same!) Idea is to secure communication between client & DNS resolvers using encryption
- Uses TCP port 443 (it’s HTTPS) instead of a newer port
- Can ensure integrity as well as privacy when communication with a trusted resolver.
- ISPs as well as many external public resolvers offer DNS over HTTPS
- Cloud companies like it a lot & may turn our DNS resolver ecosystem upside down.
- Some of DNS software vendors give option to offer DoH support for ISPs (do it!)

Questions?

anurag@he.net
Twitter: @anurag_bhatia