Tutorial on Reverse DNS: A Primer for Network Engineers

David Huberman
ICANN's Office of the CTO

RIPE79
14 October 2019

Purpose of this Tutorial

- Understand what reverse DNS is and why its useful for network operations
- Discover its conventions
- Learn about the zone boundaries for IPv4 and IPv6 reverse DNS
- Know how to resolve reverse DNS queries and understand the reverse DNS resolution hierarchy

- This tutorial does not cover how to *delegate* reverse DNS authority to a name server for the IP address blocks you operate. Each RIR publishes a really good “How-to Guide” that you can follow. This tutorial will, however, help you understand the principles behind how the RIRs manage reverse DNS.

Tutorial Agenda

1. Introduction
2. Resolving DNS Queries: Important Concepts
3. Resolving Reverse DNS Queries
4. More About Zone Delegations
5. Some Closing Thoughts

Introduction

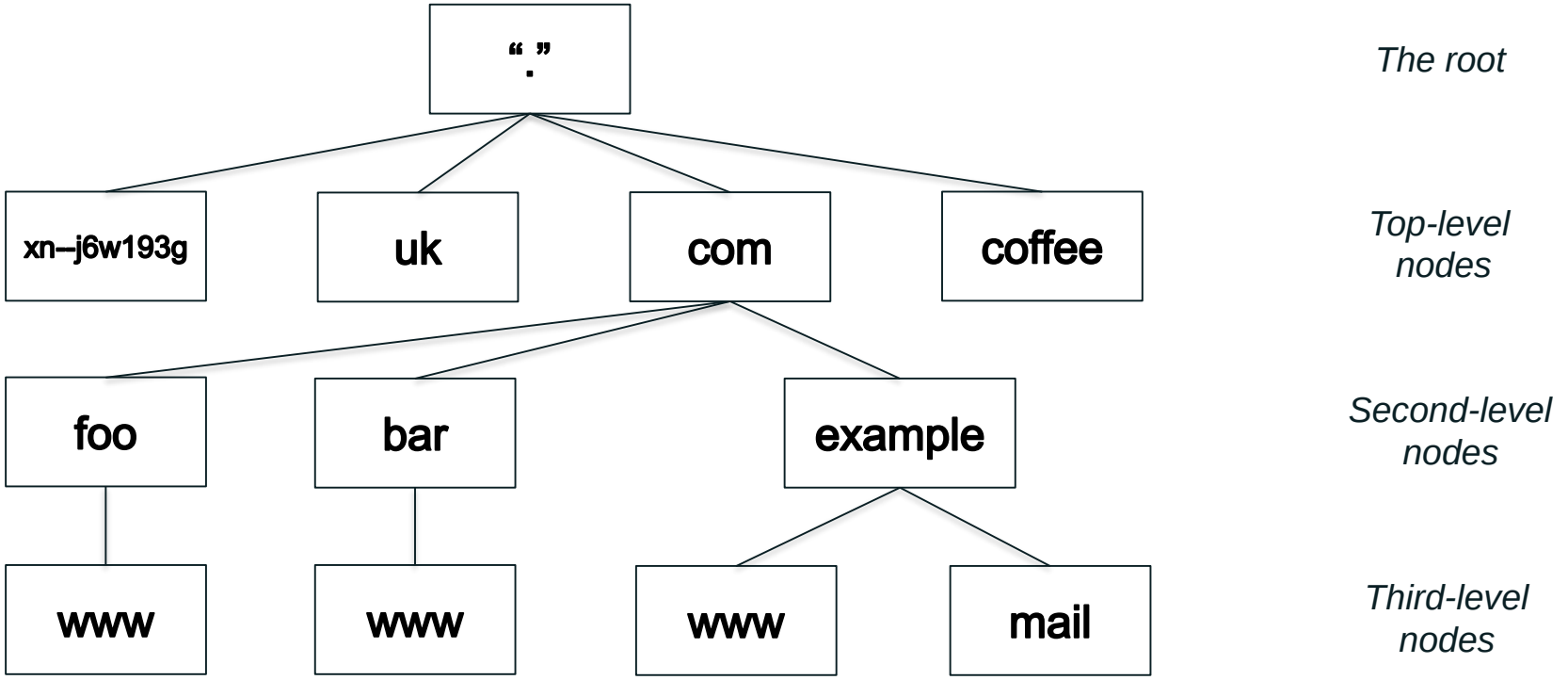
What is Forward DNS?

Forward DNS is what most people are describing when they talk about “the DNS”:

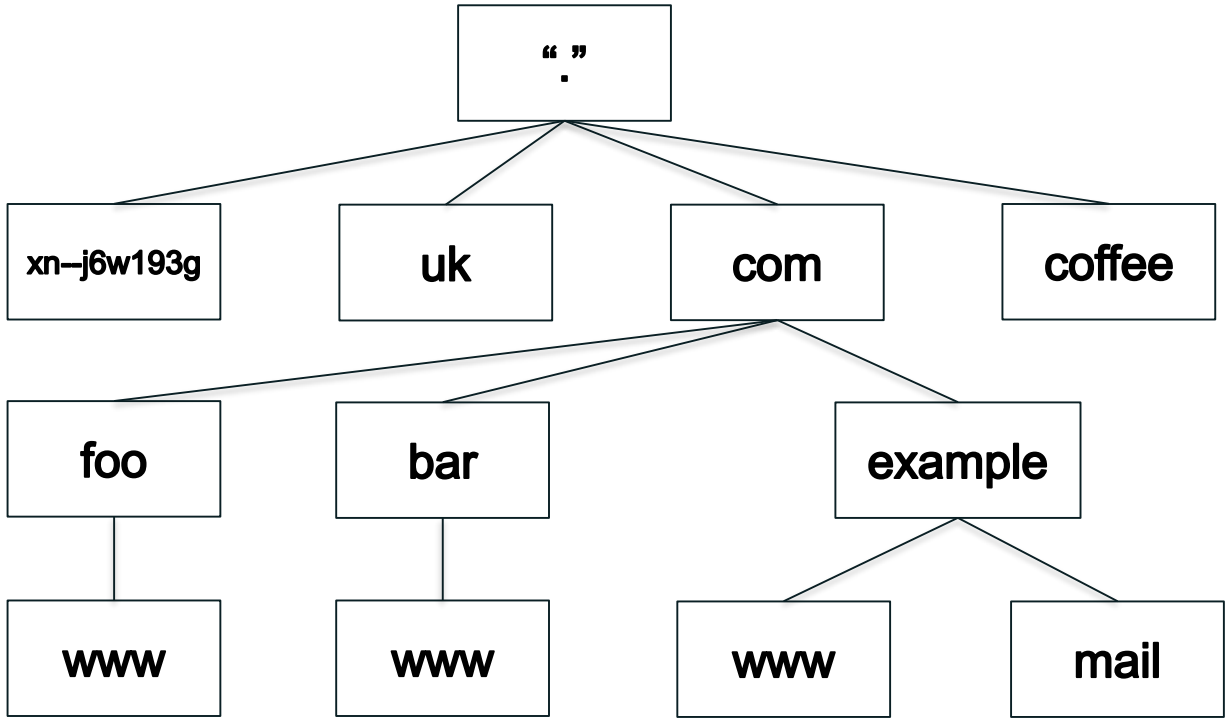
- **www.icann.org** is hosted at the IP address 192.0.32.7
- Humans do not want to have to memorize IP addresses
- The Domain Name System (DNS) maps semantic names (easily understood by humans) to these IP addresses

The Name Space

- DNS database structure is an inverted tree called the ***name space***
- Each node has a label
- The root node (and only the root node) has a null label



The Domain Name Label



- **RFC1034 section 3.1**:

  “When a user needs to type a domain name, the length of each label is omitted and the labels are separated by dots ("."). Since a complete domain name ends with the root label, this leads to a printed form which ends in a dot.”

- **www.example.com.**

- **mail.bar.uk.**

What is Reverse DNS

Reverse DNS is the opposite of forward DNS:

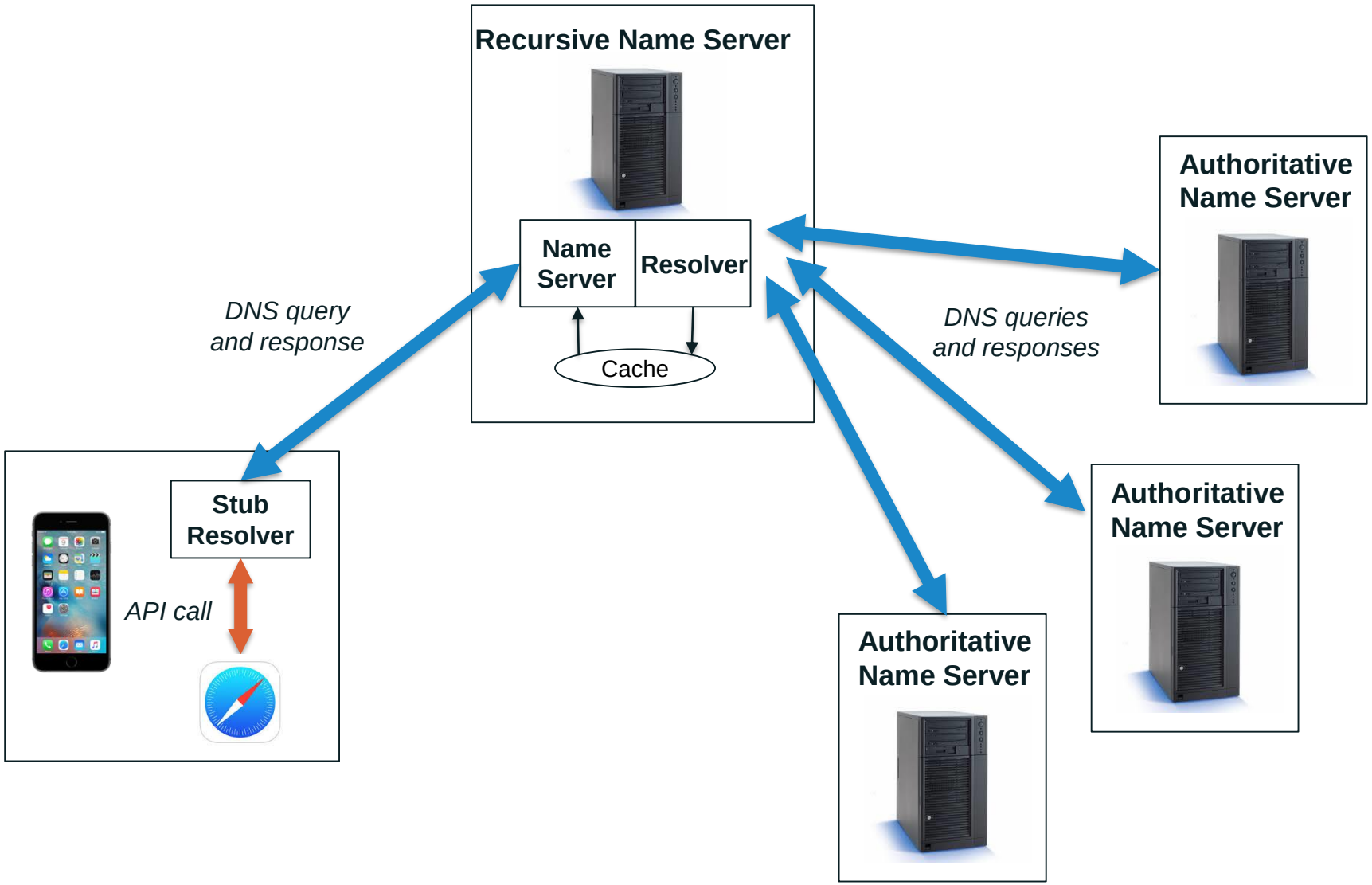
- 64.78.40.7 maps to ICANN's mail server **out.west.pexch112.icann.org.**

In operational networks, reverse DNS has two use cases relevant to network engineers:

- Authentication of mail server hostnames
- Marking network elements to make tools like traceroute more useful to humans

Resolving DNS Queries: Important Concepts

DNS Resolution Components at a Glance



Hierarchical Resolution

Assuming our resolver caches don’t help us here, let’s look up www.example.com.

- We have to resolve “.”
- We have to resolve .com
- We have to resolve example.com
- We have to resolve www.example.com

Resolving “.”

- When a recursive resolver boots up, it only has pre-packed config data about the root zone NSset (a root hints file). But that list of NSes may not be current. So the first thing it needs to do is obtain the current root zone NSset.
- During start up, it initializes its cache with a **priming query**. It reads the pre-packaged config data and performs a lookup of the root zone from one of the name servers in the config file.
- It receives back a NS RRset for the root node (“.”) from the name server it asked
- The priming query fills the resolver’s cache with the NS records for the 13 root name servers [**a.root-servers.net.** through **m.root-servers.net.**], as well as some or all of A and AAAA records associated with those

The Root Zone

$ dig . ns

```
;; ANSWER SECTION:
.                       518400  IN      NS      e.root-servers.net.
.                       518400  IN      NS      h.root-servers.net.
.                       518400  IN      NS      l.root-servers.net.
.                       518400  IN      NS      i.root-servers.net.
.                       518400  IN      NS      a.root-servers.net.
.                       518400  IN      NS      d.root-servers.net.
.                       518400  IN      NS      c.root-servers.net.
.                       518400  IN      NS      b.root-servers.net.
.                       518400  IN      NS      j.root-servers.net.
.                       518400  IN      NS      k.root-servers.net.
.                       518400  IN      NS      g.root-servers.net.
.                       518400  IN      NS      m.root-servers.net.
.                       518400  IN      NS      f.root-servers.net.

;; ADDITIONAL SECTION:
e.root-servers.net.     518400  IN      A       192.203.230.10
e.root-servers.net.     518400  IN      AAAA    2001:500:a8::e
h.root-servers.net.     518400  IN      A       198.97.190.53
h.root-servers.net.     518400  IN      AAAA    2001:500:1::53
l.root-servers.net.     518400  IN      A       199.7.83.42
l.root-servers.net.     518400  IN      AAAA    2001:500:9f::42
i.root-servers.net.     518400  IN      A       192.36.148.17
i.root-servers.net.     518400  IN      AAAA    2001:7fe::53
a.root-servers.net.     518400  IN      A       198.41.0.4
a.root-servers.net.     518400  IN      AAAA    2001:503:ba3e::2:30
d.root-servers.net.     518400  IN      A       199.7.91.13
d.root-servers.net.     518400  IN      AAAA    2001:500:2d::d
c.root-servers.net.     518400  IN      A       192.33.4.12
c.root-servers.net.     518400  IN      AAAA    2001:500:2::c
b.root-servers.net.     518400  IN      A       199.9.14.201
b.root-servers.net.     518400  IN      AAAA    2001:500:200::b
j.root-servers.net.     518400  IN      A       192.58.128.30
j.root-servers.net.     518400  IN      AAAA    2001:503:c27::2:30
k.root-servers.net.     518400  IN      A       193.0.14.129
k.root-servers.net.     518400  IN      AAAA    2001:7fd::1
g.root-servers.net.     518400  IN      A       192.112.36.4
g.root-servers.net.     518400  IN      AAAA    2001:500:12::d0d
m.root-servers.net.     518400  IN      A       202.12.27.33
m.root-servers.net.     518400  IN      AAAA    2001:dc3::35
f.root-servers.net.     518400  IN      A       192.5.5.241
f.root-servers.net.     518400  IN      AAAA    2001:500:2f::f
```

Resolving “.com”

- The recursive resolver asks the best* root server to help it resolve www.example.com
- The root server answers with the NS RRset for .com, and the resolver fills its cache for .com

* Best is defined subjectively. For some resolver software, it’s the topologically closest server. Other resolver packages might use round robin to randomly choose a root server to query.

The .com Domain

$ dig +norecurse @L.root-servers.net www.example.com A

```
;; AUTHORITY SECTION:
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.

;; ADDITIONAL SECTION:
a.gtld-servers.net.     172800  IN      A       192.5.6.30
b.gtld-servers.net.     172800  IN      A       192.33.14.30
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
h.gtld-servers.net.     172800  IN      A       192.54.112.30
i.gtld-servers.net.     172800  IN      A       192.43.172.30
j.gtld-servers.net.     172800  IN      A       192.48.79.30
k.gtld-servers.net.     172800  IN      A       192.52.178.30
l.gtld-servers.net.     172800  IN      A       192.41.162.30
m.gtld-servers.net.     172800  IN      A       192.55.83.30
a.gtld-servers.net.     172800  IN      AAAA    2001:503:a83e::2:30
b.gtld-servers.net.     172800  IN      AAAA    2001:503:231d::2:30
c.gtld-servers.net.     172800  IN      AAAA    2001:503:83eb::30
d.gtld-servers.net.     172800  IN      AAAA    2001:500:856e::30
e.gtld-servers.net.     172800  IN      AAAA    2001:502:1ca1::30
f.gtld-servers.net.     172800  IN      AAAA    2001:503:d414::30
g.gtld-servers.net.     172800  IN      AAAA    2001:503:eea3::30
h.gtld-servers.net.     172800  IN      AAAA    2001:502:8cc::30
i.gtld-servers.net.     172800  IN      AAAA    2001:503:39c1::30
j.gtld-servers.net.     172800  IN      AAAA    2001:502:7094::30
k.gtld-servers.net.     172800  IN      AAAA    2001:503:d2d::30
l.gtld-servers.net.     172800  IN      AAAA    2001:500:d937::30
m.gtld-servers.net.     172800  IN      AAAA    2001:501:b1f9::30
```

Resolving the Rest of Our Query

- The resolver asks .com's authoritative name servers for information about www.example.com

  $ dig +norecurse @a.gtld-servers.net www.example.com A

- The authoritative name servers for .com respond with the NS records for example.com

- The resolver asks example.com's authoritative name servers for information on www.example.com

  $ dig +norecurse @AUTHNS.FOO www.example.com A

- The response is the A record for the host www.example.com

Resolution Redux

For our www.example.com resolution:

- Ask the root
- Ask the TLD auth NS
- Ask the domain’s auth NS
- Get a final answer
- Our query involved three classes of resource records:
  - NS records
  - A records
  - AAAA records

Resolving Reverse DNS Queries

A Practical Use of Reverse DNS: Mail Server Authentication

- ICANN's mail server is **out.west.pexch112.icann.org** and it is hosted on two IP addresses:
  - 64.78.40.7
  - 64.78.40.10
- When mail is delivered via SMTP, the originating mail server presents its credentials to the destination mail server. ["Hi! I'm the mail server out.west.pexch112.icann.org!"]
- One of the useful mechanisms to combat spam is to authenticate those credentials, to ensure the mail server credentials are genuine (not forged)
- To authenticate the credentials, the destination mail server performs a reverse DNS lookup of the IP address of the originating mail server it is connected to. ["I know that the host 64.78.40.7 is connected to me and is trying to deliver mail to me. It claims it is the mail server out.west.pexch112.icann.org. Let me perform a reverse DNS lookup to see if these two statements reconcile."]

So how do we do that?

Reverse DNS Resolution: ARPA

- The reverse DNS TLD is in the **.ARPA** TLD
- It has an IPv4 version: **in-addr.arpa**
- It has an IPv6 version: **ip6.arpa**
- The reverse domain is appended to the IP address:
  - 64.78.40.7.in-addr.arpa … right?
  - No!
  - **7.40.78.64.in-addr.arpa.**
  - But why???

… Because it's Just Like Forward DNS

- Forward DNS resolution goes from right to left:
  - **www.example.com.**
- Conceptually, note that it is also going from least specific to more specific
  - "." encompasses the entire name space
  - .com represents the entire .com TLD
  - example.com covers the entire example domain
  - www.example.com is a specific host

Reverse DNS from Least to Most Specific: IPv4

- The least specific zone in an IPv4 name space is a **/8 zone**:
  64.in-addr.arpa covers all reverse DNS zones for addresses in 64.0.0.0/8
- The next least specific zone in an IPv4 name space is a **/16 zone**:
  78.64.in-addr.arpa covers all reverse DNS zones for addresses in 64.78.0.0/16
- As expected, the next zone boundary in IPv4 is for the **/24 zone**:
  40.78.64.in-addr.arpa covers all reverse DNS zones for addresses in 64.78.40.0/24

***By the way …* notice that we left out all leading .0 octets in our zone names**

- 0.0.0.64.in-addr.arpa is incorrect
- 64.in-addr.arpa is correct

Zone Boundaries in IPv6

- In IPv6 reverse DNS, zone boundaries are defined on the *nibble boundary*
  - In classical computer science, a “nibble” is any four-bit aggregation
  - IPv6 lends itself nicely to nibble boundaries, because it is represented as eight 16-bit quartets:

    _ _ _ _: _ _ _ _: _ _ _ _: _ _ _ _: _ _ _ _: _ _ _ _: _ _ _ _: _ _ _ _
  - Each underscore represents 4-bits:

    2 0 0 1: D B 8 0: 0 0 0 0: 0 0 0 0: 0 0 0 0: 0 0 0 0: 0 0 0 0: 0 0 0 0
  - Each number, therefore, is a nibble boundary, so each number signifies a zone boundary for reverse DNS
  - Similar to IPv4, we reverse the numbers when defining reverse DNS zones.

    0.8.b.d.1.0.0.2.ip6.arpa is the reverse zone for 2001:db8::/32

- Leading zeros are important, because the reverse zone’s domain must mathematically match the number of bits it represents

IPv6 Reverse DNS in Practice

- Under the current allocation practices agreed upon by IANA and the RIRs, the IANA delegates /12 blocks to each RIR as needed
- In 2006, IANA delegated 2400::/12 for APNIC to issue to its customers
- APNIC is therefore responsible for the delegation of reverse DNS at the **/12 boundary**:

  0.4.2.ip6.arpa covers all reverse DNS zones for addresses in 2400::/12
- APNIC often issues /32s to customers
- A customer is delegated reverse DNS at the **/32 boundary**:

  0.8.b.7.0.0.4.2.ip6.arpa covers all reverse DNS zones for addresses in 2400:7b8::/32

Reverse DNS Resolution: the PTR Record

- When discussing forward DNS, we were resolving A records and AAAA records, which resolve a domain name to an IP address
- In reverse DNS, the RR type is PTR:
  - “Pointer” record
  - Resolves a domain name that ends in in-addr.arpa or ip6.arpa that represents an IP address into a fully-qualified domain name
  - (It requires a FQDN, so the trailing dot is necessary in the zone file!)

Going Back to Our Mail Server Example

- ICANN's mail server is **out.west.pexch112.icann.org** and it is hosted on two IP addresses:
  - 64.78.40.7
  - 64.78.40.10
- We have to query ".”
- We have to query ".arpa”
- We have to query "in-addr.arpa”
- We have to query "64.in-addr.arpa”
- We have to query "78.64.in-addr.arpa”
- We have to query "40.78.64.in-addr.arpa”
- We have to query "7.40.78.64.in-addr.arpa” or "10.40.78.64.in-addr.arpa”

Going Back to Our Mail Server Example

- ICANN's mail server is **out.west.pexch112.icann.org** and it is hosted on two IP addresses:
  - 64.78.40.7
  - 64.78.40.10
- We have to query ".”
- We have to query ".arpa" ← not true
- We have to query "in-addr.arpa"
- We have to query "64.in-addr.arpa"
- We have to query "78.64.in-addr.arpa" ← not true
- We have to query "40.78.64.in-addr.arpa"
- We have to query "7.40.78.64.in-addr.arpa" or "10.40.78.64.in-addr.arpa"

The Root Zone

$ dig +norecurse @L.root-servers.net . ns

```
;; ANSWER SECTION:
.                       518400  IN      NS      e.root-servers.net.
.                       518400  IN      NS      h.root-servers.net.
.                       518400  IN      NS      l.root-servers.net.
.                       518400  IN      NS      i.root-servers.net.
.                       518400  IN      NS      a.root-servers.net.
.                       518400  IN      NS      d.root-servers.net.
.                       518400  IN      NS      c.root-servers.net.
.                       518400  IN      NS      b.root-servers.net.
.                       518400  IN      NS      j.root-servers.net.
.                       518400  IN      NS      k.root-servers.net.
.                       518400  IN      NS      g.root-servers.net.
.                       518400  IN      NS      m.root-servers.net.
.                       518400  IN      NS      f.root-servers.net.

;; ADDITIONAL SECTION:
e.root-servers.net.     518400  IN      A       192.203.230.10
e.root-servers.net.     518400  IN      AAAA    2001:500:a8::e
h.root-servers.net.     518400  IN      A       198.97.190.53
h.root-servers.net.     518400  IN      AAAA    2001:500:1::53
l.root-servers.net.     518400  IN      A       199.7.83.42
l.root-servers.net.     518400  IN      AAAA    2001:500:9f::42
i.root-servers.net.     518400  IN      A       192.36.148.17
i.root-servers.net.     518400  IN      AAAA    2001:7fe::53
a.root-servers.net.     518400  IN      A       198.41.0.4
a.root-servers.net.     518400  IN      AAAA    2001:503:ba3e::2:30
d.root-servers.net.     518400  IN      A       199.7.91.13
d.root-servers.net.     518400  IN      AAAA    2001:500:2d::d
c.root-servers.net.     518400  IN      A       192.33.4.12
c.root-servers.net.     518400  IN      AAAA    2001:500:2::c
b.root-servers.net.     518400  IN      A       199.9.14.201
b.root-servers.net.     518400  IN      AAAA    2001:500:200::b
j.root-servers.net.     518400  IN      A       192.58.128.30
j.root-servers.net.     518400  IN      AAAA    2001:503:c27::2:30
k.root-servers.net.     518400  IN      A       193.0.14.129
k.root-servers.net.     518400  IN      AAAA    2001:7fd::1
g.root-servers.net.     518400  IN      A       192.112.36.4
g.root-servers.net.     518400  IN      AAAA    2001:500:12::d0d
m.root-servers.net.     518400  IN      A       202.12.27.33
m.root-servers.net.     518400  IN      AAAA    2001:dc3::35
f.root-servers.net.     518400  IN      A       192.5.5.241
f.root-servers.net.     518400  IN      AAAA    2001:500:2f::f
```

The in-addr.arpa Domain

$ dig +norecurse @L.root-servers.net in-addr.arpa ns

```
;; ANSWER SECTION:
in-addr.arpa.           3600    IN      NS      c.in-addr-servers.arpa.
in-addr.arpa.           3600    IN      NS      a.in-addr-servers.arpa.
in-addr.arpa.           3600    IN      NS      f.in-addr-servers.arpa.
in-addr.arpa.           3600    IN      NS      b.in-addr-servers.arpa.
in-addr.arpa.           3600    IN      NS      e.in-addr-servers.arpa.
in-addr.arpa.           3600    IN      NS      d.in-addr-servers.arpa.

;; ADDITIONAL SECTION:
a.in-addr-servers.arpa. 78012   IN      A       199.180.182.53
a.in-addr-servers.arpa. 67140   IN      AAAA    2620:37:e000::53
b.in-addr-servers.arpa. 78012   IN      A       199.253.183.183
b.in-addr-servers.arpa. 67140   IN      AAAA    2001:500:87::87
c.in-addr-servers.arpa. 78012   IN      A       196.216.169.10
c.in-addr-servers.arpa. 67140   IN      AAAA    2001:43f8:110::10
d.in-addr-servers.arpa. 78012   IN      A       200.10.60.53
d.in-addr-servers.arpa. 67140   IN      AAAA    2001:13c7:7010::53
e.in-addr-servers.arpa. 78012   IN      A       203.119.86.101
e.in-addr-servers.arpa. 67140   IN      AAAA    2001:dd8:6::101
f.in-addr-servers.arpa. 78012   IN      A       193.0.9.1
f.in-addr-servers.arpa. 67140   IN      AAAA    2001:67c:e0::1
```

The 64.in-addr.arpa Domain

$ dig +norecurse @a.in-addr-arpa-servers.arpa 64.in-addr.arpa ns

```
;; ANSWER SECTION:
64.in-addr.arpa.        66882   IN      NS      u.arin.net.
64.in-addr.arpa.        66882   IN      NS      arin.authdns.ripe.net.
64.in-addr.arpa.        66882   IN      NS      y.arin.net.
64.in-addr.arpa.        66882   IN      NS      r.arin.net.
64.in-addr.arpa.        66882   IN      NS      z.arin.net.
64.in-addr.arpa.        66882   IN      NS      x.arin.net.

;; ADDITIONAL SECTION:
x.arin.net.             26781   IN      A       199.71.0.63
u.arin.net.             26780   IN      A       204.61.216.50
u.arin.net.             61323   IN      AAAA    2001:500:14:6050:ad::1
z.arin.net.             26781   IN      A       199.212.0.63
y.arin.net.             26781   IN      A       192.82.134.30
```

The 40.78.64.in-addr.arpa Domain

$ dig +norecurse @U.arin.net 40.78.64.in-addr.arpa ns

```
;; ANSWER SECTION:
40.78.64.in-addr.arpa.  68413   IN      NS      ns3.intermedia.net.
40.78.64.in-addr.arpa.  68413   IN      NS      ns2.intermedia.net.

;; ADDITIONAL SECTION:
ns2.intermedia.net.     596804  IN      A       64.78.22.176
ns3.intermedia.net.     596804  IN      A       162.244.196.45
```

The PTR Response for Our Mail Relay

$ dig +norecurse @ns2.intermedia.net 7.40.78.64.in-addr.arpa PTR

```
;; ANSWER SECTION:
7.40.78.64.in-addr.arpa. 81957  IN      PTR     out.west.pexch112.icann.org.

;; AUTHORITY SECTION:
40.78.64.in-addr.arpa.   68357  IN      NS      ns2.intermedia.net.
40.78.64.in-addr.arpa.   68357  IN      NS      ns3.intermedia.net.

;; ADDITIONAL SECTION:
ns2.intermedia.net.      596748 IN      A       64.78.22.176
ns3.intermedia.net.      596748 IN      A       162.244.196.45
```

More About Zone Delegations

Your Delegations and Your RIR

Reminder:

- In IPv4, zones are delegated at either the /8, the /16, or the /24 boundary
- In IPv6, zones are delegated on the nibble boundary

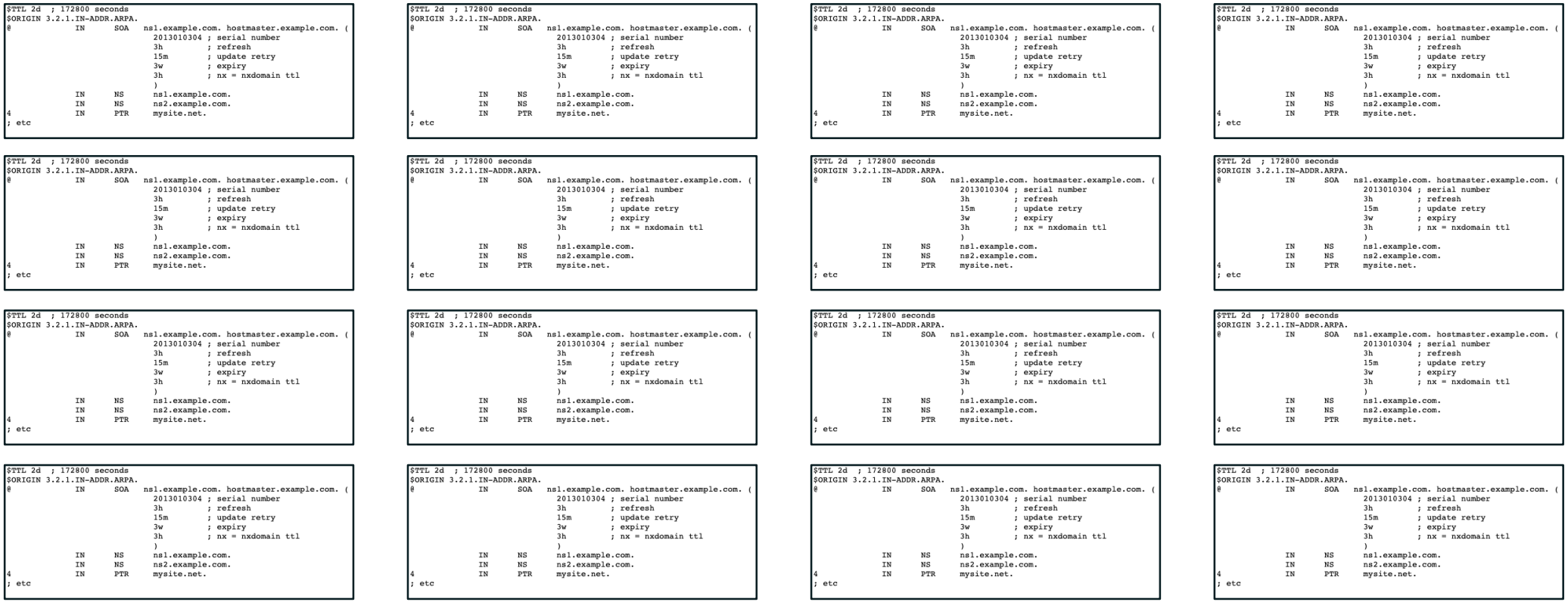
You are the Registrant of a /24

- If you have a /24, you configure a reverse domain in your zone files for the /24

```
$TTL 2d  ; 172800 seconds
$ORIGIN 3.2.1.IN-ADDR.ARPA.
@               IN      SOA    ns1.example.com. hostmaster.example.com. (
                                 2013010304 ; serial number
                                 3h         ; refresh
                                 15m        ; update retry
                                 3w         ; expiry
                                 3h         ; nx = nxdomain ttl
                                 )
                IN      NS       ns1.example.com.
                IN      NS       ns2.example.com.
4               IN      PTR      mysite.net.
; etc
```

You are the Registrant of a /20

- If you have a /20, you configure sixteen reverse domains – one for each /24 - in your zone files



Delegations Smaller than a /16

- /24 = 1 reverse domain
- /23 = 2 reverse domains
- /22 = 4 reverse domains
- [. . .]
- /18 = 64 reverse domains
- /17 = 128 reverse domains

Your Delegation of a /16 (or larger)

- If you are the registrant of a /16, simply insert a new level of hierarchy:
  - Configure a /16 reverse domain with authoritative NSes
- Then configure the individual /24 reverse domain zone files
- If you are the registrant of a /15, configure two /16 reverse domains, and then configure two sets of 256 /24 reverse domain zone files

IPv6 Delegations

- A /32 is the default size for LIRs. It acts just like a /16 in IPv4. You answer for the /32, then define individual reverse domains as necessary – respecting the nibble boundaries.
- If you have a /48 or something smaller, the RIR will respond to the /32, and delegate the individual /48s or /44s (etc.) to your name servers

Fun Fact: Very Large Address Blocks

- In the early days of IPv4, Jon Postel issued /8s to large companies for their networks, which were larger than 256 subnets. Some companies still have /8s.
- We can envision a possible IPv6 network that is huge – that might use something as large as a /12
- For these types of fundamentally large address blocks, you configure your zone files just like you would in our previous examples – hierarchy at each zone boundary. But DNS resolution actually bypasses your RIR. ARPA delegates to IANA, which delegates directly to your authoritative NSes.

CNAMES and Really Small Zones

- Delegation boundaries are easy to respect as an LIR. But what about your customers who need reverse DNS for small address blocks (e.g. a /28 in IPv4)?
- There can only be one delegation by the RIR for each /24
- To overcome this restriction, we use the **CANONICAL NAME** (CNAME) RR type
  - “An alias name for a host. Causes redirection for a single RR at the owner-name.”
  - www.RipeIsAwesome.net CNAME www.ripe.net
  - Importantly, the existence of a CNAME for a particular domain causes a **new DNS lookup**
  - That is what allows us to get around the “one delegation” rule
- In reverse DNS, **RFC2317** was written to define how to use CNAME RRs in reverse DNS zone files to handle address blocks smaller than a /24

CNAME Resolution Path

- Example network of 1.2.3.0/24
- **3.2.1.in-addr.arpa.** has **NS1.FOO.BAR** as an authoritative name server
- We want to delegate the /28 for 1.2.3.128 through 1.2.3.143 to a different set of authoritative name servers
- In our zone snippet for 3.2.1.in-addr.arpa. we use CNAMES

```
128/28          NS      ns1.SOMEOTHERNAME.bar.
128/28          NS      ns2.SOMEOTHERNAME.bar.
;
129             CNAME   129.128/28.3.2.1.in-addr.arpa.
130             CNAME   130.128/28.3.2.1.in-addr.arpa.
131             CNAME   131.128/28.2.2.1.in-addr.arpa.
```

CNAME Resolution Path

Example network of 1.2.3.0/24:

- Query the .arpa zone
- Query the X.in-addr.arpa. zone (the /8)
- Query the Z.Y.X.in-addr.arpa. zone (the /24)
- Resolve the CNAME, which begins a new resolution path at the specified NSes

Some Closing Thoughts

Traceroute

- Mail server authentication is a big use of rDNS
- The other really useful implementation of rDNS is in tagging network elements for troubleshooting.

```
$ traceroute www.ietf.org
traceroute to www.ietf.org.cdn.cloudflare.net (104.20.1.85), 64 hops max, 40 byte packets
 1  eth2-12.core1.bw.nyc.access.net (166.84.1.28)  0.391 ms  0.418 ms  0.260 ms
 2  gi0-0-0-12.nr11.b001944-1.jfk01.atlas.cogentco.com (38.142.186.73)  1.363 ms  1.056 ms  1.097 ms
 3  te0-3-0-16.rcr24.jfk01.atlas.cogentco.com (154.24.22.233)  3.884 ms  0.938 ms
    te0-3-0-16.rcr23.jfk01.atlas.cogentco.com (154.24.22.229)  0.859 ms
 4  be2600.rcr21.ewr02.atlas.cogentco.com (154.54.40.30)  7.931 ms  1.868 ms  1.578 ms
 5  Delrey_Technologies.demarc.cogentco.com (38.104.44.10)  15.503 ms  30.005 ms  52.285 ms
 6  104.20.1.85 (104.20.1.85)  1.158 ms  2.004 ms  2.577 ms
```

- But this is a good example of rDNS behavior in the real world. It’s out-of-date.

Out-of-date is Common

- If you have a fully integrated OSS or IPAM managing your IP address assignments and all your rDNS, then you likely have good records.
- But many (most?) networks do not enjoy such tooling, and the first thing we see go out-of-date are the PTR records.
  - IP addresses change which customers they are assigned to
  - IP address assignment size changes (think: CNAME zone snippets)
  - Changes to network elements (re-architecture) and someone forgets to update all the PTR records. Because that's not a normal part of a network engineer's day-to-day!
- Another general takeaway: so much of the rDNS tree is unpopulated. If you think about all the traceroutes you do, you see lots of hops have no labels. This is common. The rDNS is often not a reliable source of information about a host.

Go Read RFC8501

- Lee Howard published a very useful RFC:
  - “Reverse DNS in IPv6 for Internet Service Providers”
- Includes a sections on Dynamic DNS and negative responses

RIR How-to Guides

- Each of the RIRs publishes a reverse DNS “how-to guide” which describes the procedures you need to follow to delegate reverse zones.
- Hopefully you now better understand how things work in rDNS and can now ask very good questions of the RIR hostmasters!

One World, One Internet

Visit us at **icann.org**

@icann

facebook.com/icannorg

youtube.com/icannnews

flickr.com/icann

linkedin/company/icann

slideshare/icannpresentations

soundcloud/icann