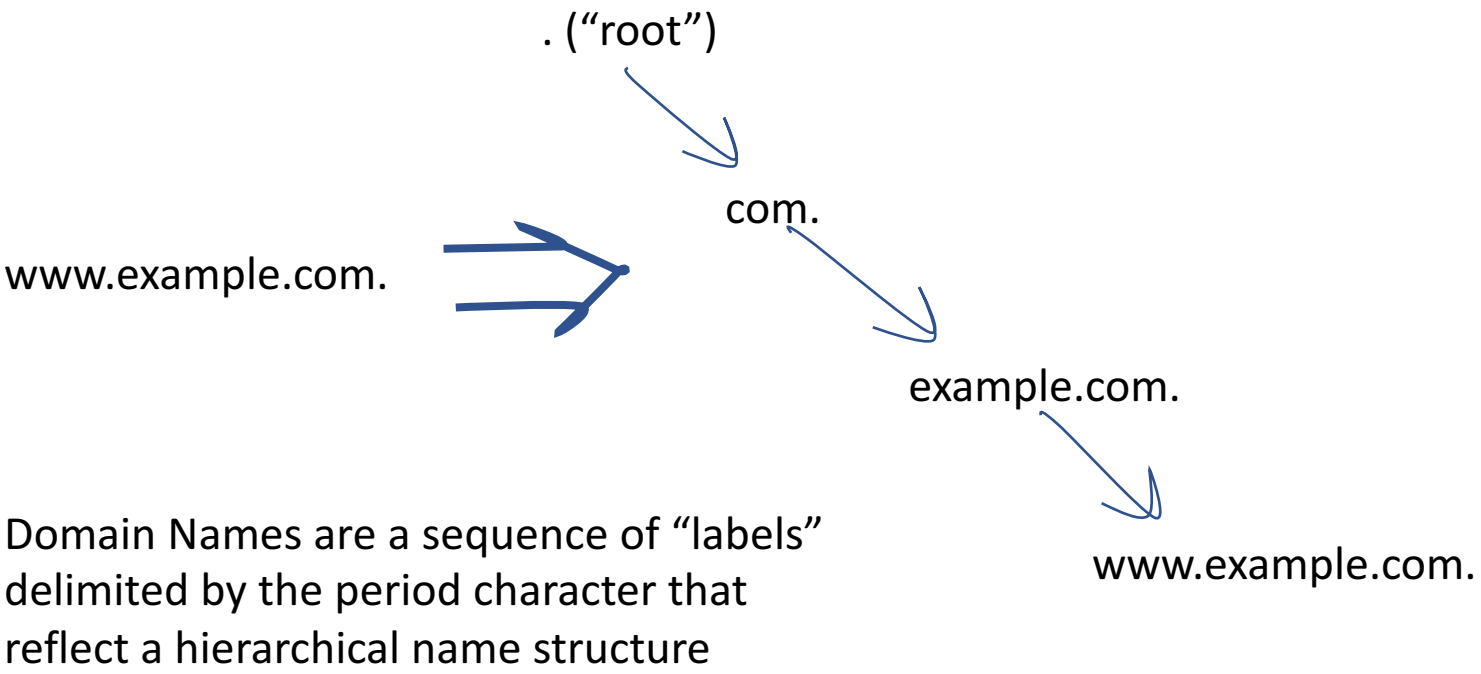
The Root of the DNS

Geoff Huston
APNIC

March 2017

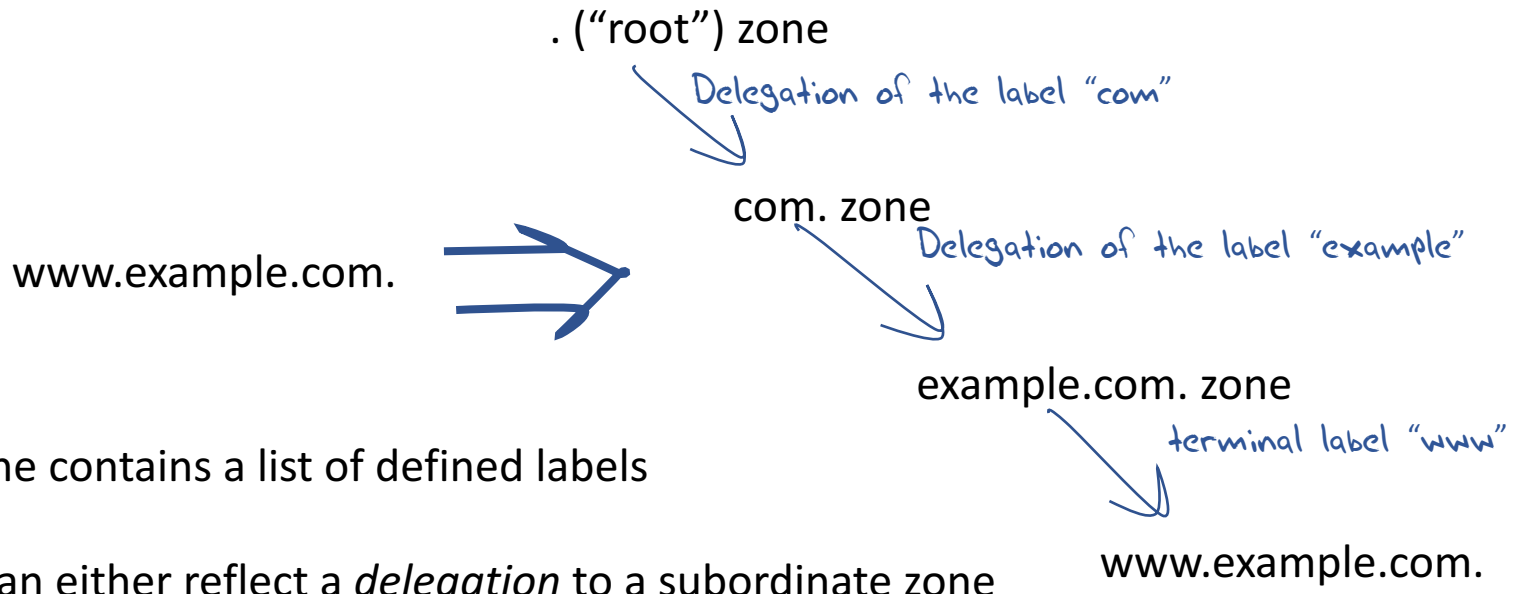
The Structure of the Domain Name *Space*



Domain Names are a sequence of “labels” delimited by the period character that reflect a hierarchical name structure

The Structure of the Domain Name *System*

The Domain Name System (DNS) is a distributed data collection using a delegation hierarchy that reflects the internal hierarchical structure of domain names. At each level in the name hierarchy each label represents a potential point of administrative delegation



Each zone contains a list of defined labels

Labels can either reflect a *delegation* to a subordinate zone or they can be a *terminal* label that contains attribute information associated with that label

Resolving a DNS Name

Your resolver needs need to ask a DNS server for the zone that contains the terminal label for the associated information (resource record) associated with the DNS name

But…

- Where exactly is the zone cut?
- Who are the servers?

So resolvers *discover* this information by performing a top-down iterative search…

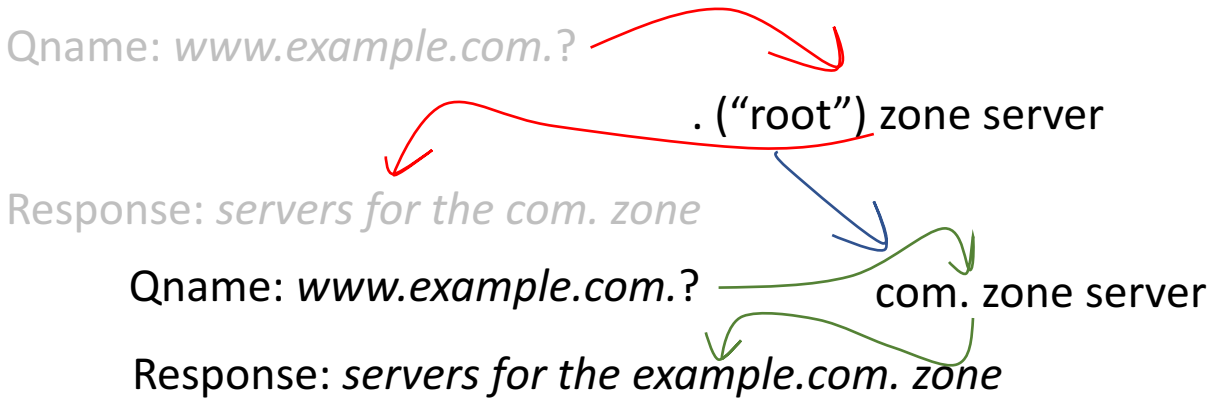
Resolving a DNS Name

Qname: *www.example.com.*?

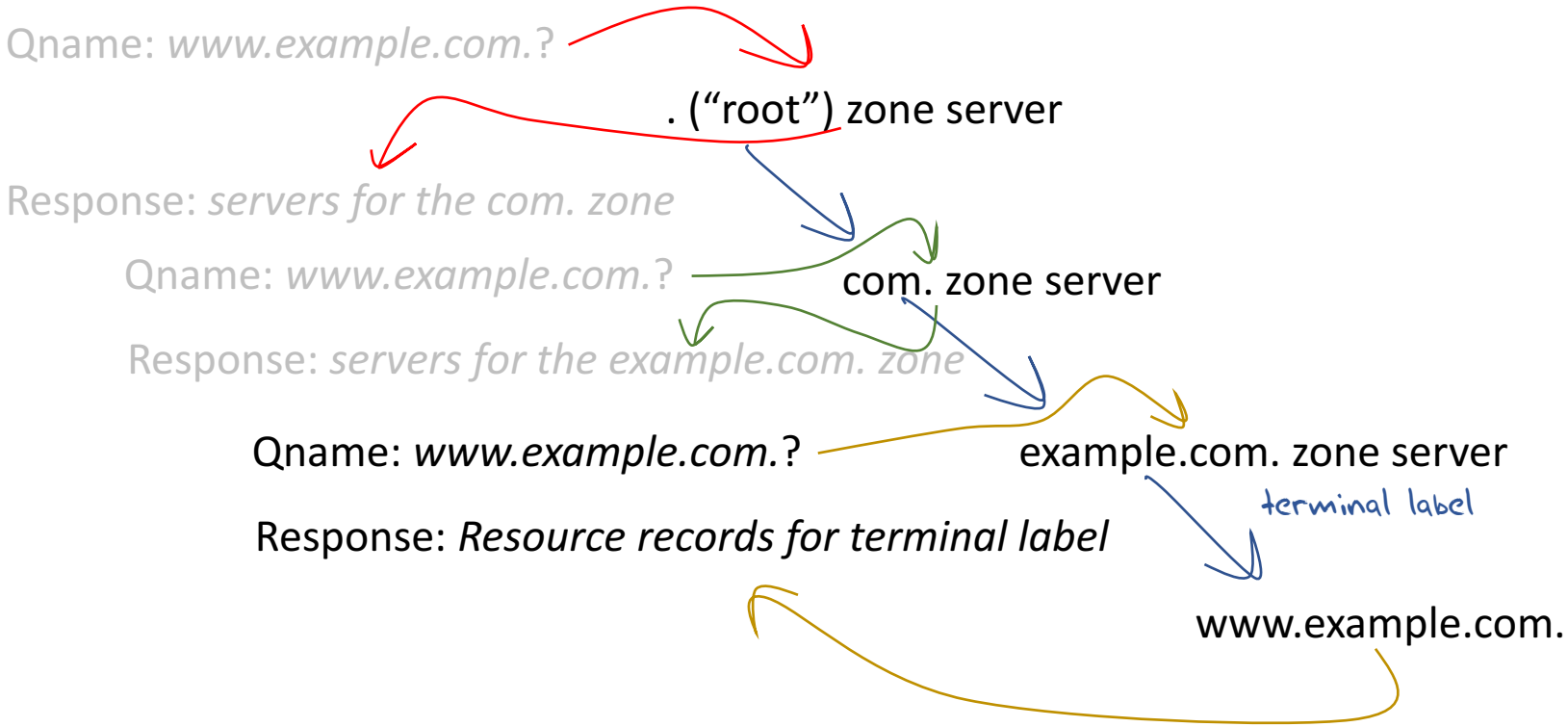
. (“root”) zone server

Response: *servers for the com. zone*

Resolving a DNS Name



Resolving a DNS Name



How do we use the Root Service?

In the next few slides we'll describe DNS behaviours and their use of the Root Service.

The DNS is full of variations in behaviour, and we can't describe them all, so we'll stick to what we observe as common behaviour here

Every DNS Resolver starts up by asking a root server a *priming query*

Resolvers typically have a “hints” file to bootstrap into the DNS, but they use this to refresh the list of root name servers by asking one of the roots for the list of name servers for the root zone (the *priming query)*

Root Server Role: Answer priming queries about the root zone

Every Name Resolution query starts by asking a root server

Every Name Resolution query starts by asking a root server

In theory!

In practice, resolvers **cache** the responses they receive, and then use the cache for subsequent queries

This holds for the root zone as much as all other zones in the DNS, so in practice most queries for delegated domain names do **not** start with queries to the root zone

Every DNS Resolver asks a root server about unknown names

DNS resolvers do not conventionally cache the entire root zone, but populate a local cache incrementally based on the names they are tasked with resolving

When a query cannot be answered from the local cache, the resolver will query a root server

Root Server Role: Answer cache miss queries from resolvers

Root servers are *promiscuous responders*

Root servers do not “know” the reason for receiving a query, and have no policy about whether or not they should respond and with what information

Root servers respond using a common current copy of the root zone to form their response

Root Server Role: Answer all queries presented to root servers in a uniform and consistent manner based only on information in the root zone

Root Service MUST be available

Not every root server needs to be reachable from every resolver that asks DNS queries, but at least one server must be available to respond to queries

If a resolver cannot reach ANY root servers then when its local cache expires it will be unable to answer queries. It will “go dark”

Root Server Role: The aggregate system of root servers must be available to respond to queries at all times

Root Service needs to be ”fast enough”

An available Root Server should respond to queries in a timely manner

The faster the root server can respond the faster the overall DNS resolution time when the local resolver(s) have cache misses on the query, but there are no pre-defined target response times

Root Server Role: The aggregate system of root servers must respond to queries within a reasonable time

Anycast Root Servers

12 of the 13 root server “letters” operate some form of “anycast” server constellation. All the servers in a constellation respond to the same public IP addresses. The routing system will direct resolvers to pass their query to a particular root letter to the “closest” member of the letter’s anycast constellation.

Anycast provides:

- faster responses to queries to the root for many DNS resolvers
- Greater resilience to hostile traffic by load sharing widely distributed attacks across the entire anycast constellation, and absorbing a single point attack on a single server instance

Using the Root “Service”

The main role of the root server system is to answer queries that are not cached in local name caches

There are many more well-formed top level labels that are not delegated labels in the root zone than those that are (1,530)

That means that the vast majority of the queries that are passed to the root zone servers generate a “no-such-name” (NXDOMAIN) response from the root system

Which Root?

There is no generic “any root server” address for a resolver to use. Resolvers need to send their queries to a specific letter.

Which letter they pick is up to the resolver. Some do round robin, some latch on to the one they think is faster. There are no particular rules that resolvers use here.

It’s not clear that resolvers use any particular heuristic to guide their choice of root server letter, nor is it clear that it matters in any case.

If there is no response, then the resolvers will switch to another root letter and repeat the query.

Futures for the way we use the Root Service

As the traffic levels to the root servers increases both as steady state query levels and instances of attacks, we keep on building bigger servers and add more instances to the existing anycast clouds

Can we improve the behaviour of the total system to improve its overall scaling properties without singling out the root server system?

DNSSEC changes Everything

Before DNSSEC we relied on the assumption that if we asked an IP address of a root server then the response was genuine

With DNSSEC we can ask anyone, and then use validation to assure ourselves that the answer is genuine

How can we use this?

DNSSEC-Enabled Directions for the Root Service

DNSSEC opens some fascinating possibilities, allowing us to explore other options in how the root zone is distributed towards DNS resolvers in addition to the convention collection of Root Server Letter anycast constellations

The underlying ability provided by DNSSEC is that no matter how you obtain a response from the root zone, you can validate its authenticity with DNSSEC

This allows us to enlist DNSSEC-aware DNS resolvers to provide authoritative DNS responses from their local cache that would've otherwise required a query to a root server. This can be used to provide a significant augmentation to the capability of the root system without actually changing the scale or capability of the dedicated root zone servers themselves

Local Root Secondaries - RFC7706

Enlist DNS resolvers to offer a root zone secondary service

If resolvers use this approach then they only need to query a root server infrequently and perform a zone transfer of the current state of the root zone (IXFR from a root server), and use this validated copy of the root zone to directly answer all queries that refer to the root zone

“Aggressive” NSEC caching

Most of the queries seen at the root are for non-existent domains

Resolvers cache the nonexistence of a given name

But a DNSSEC-signed NXDOMAIN response from the root zone actually describes a range of labels that do not exist, and it’s the range that is signed, not the actual query name

If resolvers cached this range and the signed response, then they could use the same signed response to locally answer a query for any name that falls within the same label range

This has a similar effect to RFC7706, but without any configuration overhead, nor is there any requirement for supporting root zone transfers.

This approach increases the effectiveness of the local cache by allowing the local resolver to learn entire ranges of non-existent names in the root

Research and Analysis

Can we peer inside the interactions between DNS resolvers and root servers to look at how the use the root system? Can we see to what extent resolvers spread their queries across the entire collection of root servers and to what extent they express a preference to use what they see as the “fastest” such server?

What is the interaction between V4 and V6 transports for the root servers and the anycast distributions?

What local caching parameters are used by DNS resolvers for root zone data?

To what extent do DNS resolvers ask for DNSSEC signatures for root zone data?

Questions?