DNS Deep Dive, IETF 108

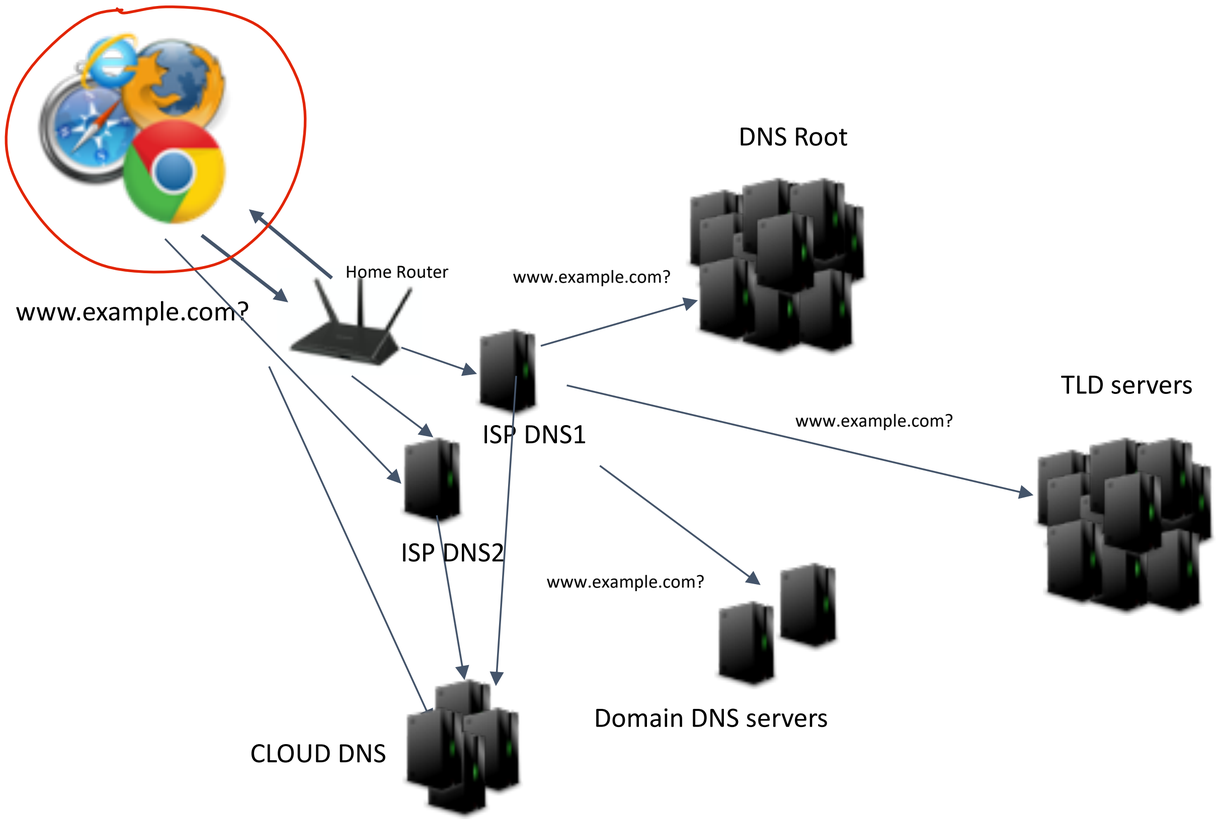
Part 1, section 3, Software

João Damas, APNIC

DNS Software

- Stub resolver - what ships with every OS, different for every OS
- Forwarder - most commonly found in home routers
- Recursive resolver software - the hardest part of DNS software
- Authoritative servers - the source of the data
- All-in-one

Stub resolvers (API)



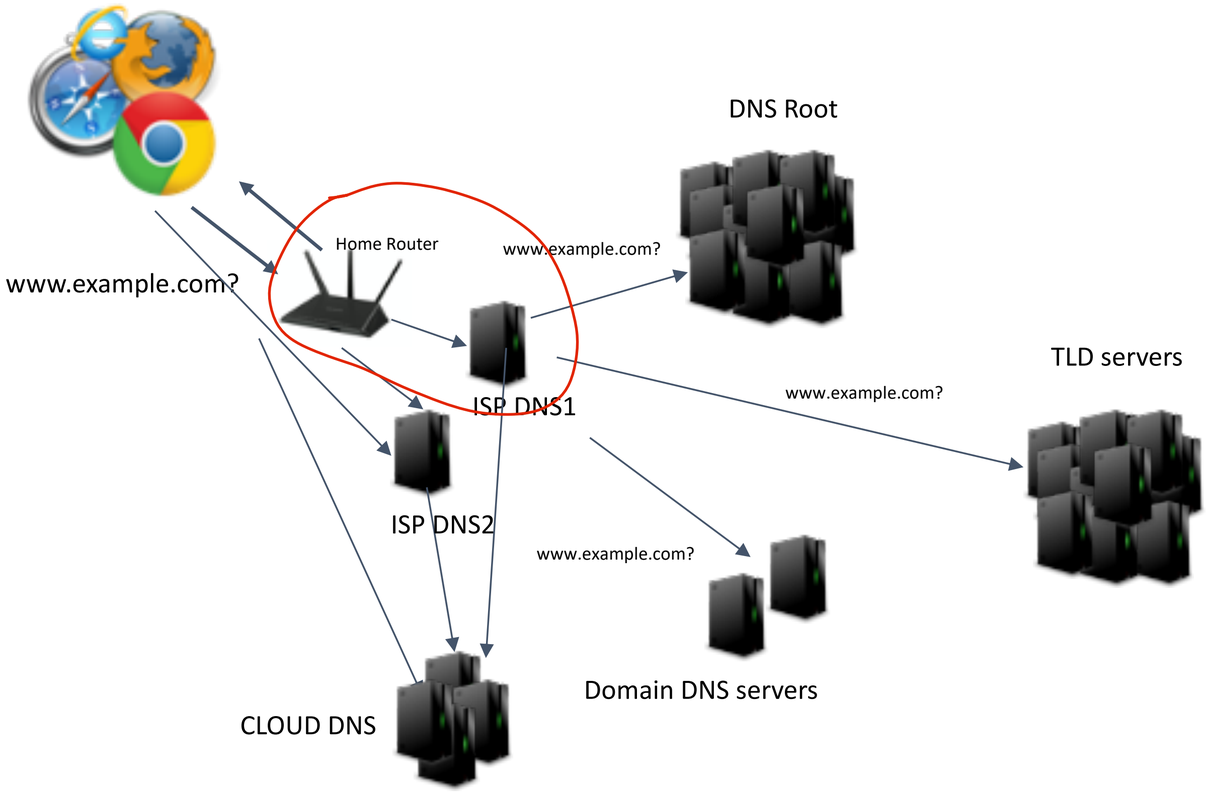
Stub resolvers (API)

- Initially just a part of the BSD Unix API
- gethostbyaddr(), gethostbyname(), gethostent(), and sethostent() functions appeared in 4.2BSD [1983, at the time of RFC 882/883]
- Still the most frequently used calls when working on Unix
- Windows has this and its own
- macOS/iOS also wrap these in more abstract calls/services
- More modern implementations like getDNS and its stub implementation, stubby, and language bindings implement client side features such as DNSSEC validation
- Linux has recently seen the introduction of systemd resolved.service
- Some apps now have their own stub (typically web browsers or “things” that use web “engines”)
- Some may include a host (or application) level cache

More on APIs

- As things evolve we may see other APIs or API-like forms for DNS transactions
  - e.g. DNS Over HTTP, is currently a way of transporting DNS over HTTPS but if you look at it with web app developer eyes, it is pretty much on the way to an API

Forwarders


DNS Root
Home Router
www.example.com?
www.example.com?
TLD servers
www.example.com?
ISP DNS1
ISP DNS2
www.example.com?
Domain DNS servers
CLOUD DNS

Forwarders

- Forwarders fall in between stubs and recursive resolvers, both from a network topology and a functionality point of view.
- They have some features typical of recursive resolvers such as a local cache but do not perform the recursive lookups against authoritative servers, they “forward” the queries they get to a full-service recursive resolver that performs the heavy duty work.

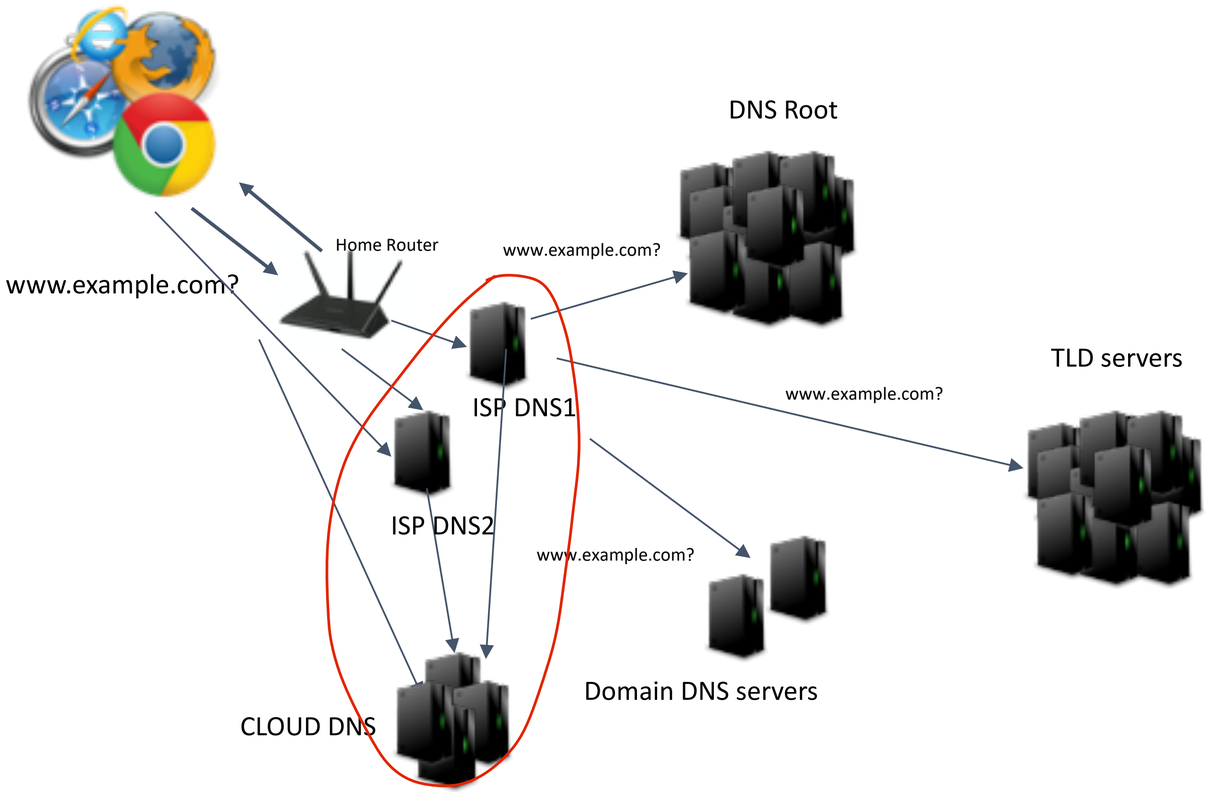
Forwarder implementations

- dnsmasq
  - by far the most common version of DNS forwarder, with various versions, of very diverse vintage, in use in the wild
  - Included as part of heaps of home router software images as it also includes a DHCP server
  - Includes added support for DNSSEC validation since v2.69 (2014)

Forwarder implementations

- BIND, MaraDNS, other recursive servers
  - Configurable as a forwarders if you wish
- These days it is common to see ISPs use full resolver being used with sizeable caches and forwarding queries to open DNS resolvers such as the OpenDNS or the quads (1,8,9)

Recursive resolvers



Recursive resolvers

- Even though today installing and running a local (to your device) recursive server is not a difficult task (at least for non-mobile devices), most people just use what is provided by the network configuration (DHCP, etc) and don’t bother with actual software deployment.
- ISPs and enterprises tend to use a mix of Open Source and proprietary software in their services.
- An increasing trend is the use of “Public Open Resolvers” where users or network administrators change their network configuration so that applications send queries to well-known servers on the Internet
  - These may be running proprietary implementations (e.g. Google, OpenDNS), or be based on DNS recursive software (e.g. Quad1, based on Knot Recursive, Quad9 based on BIND 9, etc)

Recursive resolver

- Some servers (FLOSS or not) extend DNS service to include internal or externally provided policies as part of domain blocking orders, security policies, censorship, etc
- A lot of these feeds are proprietary, at least one option has been put to the IETF (DNS RPZ)

Recursive resolver implementations

Open source

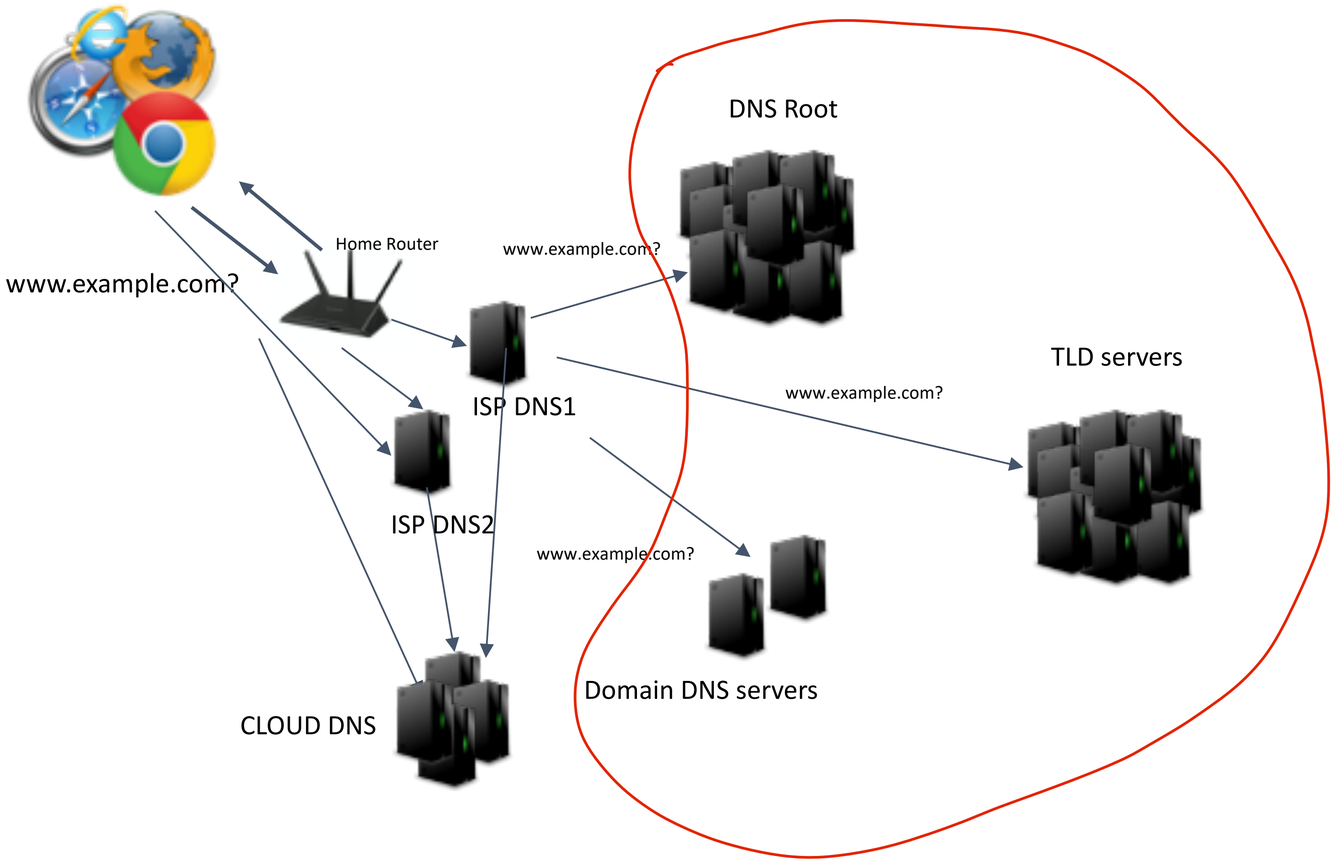
- Traditionally most DNS software has been Open Source software
  - BIND 9
  - Unbound
  - Knot resolver
  - PowerDNS Resolver

Recursive resolver implementations

Closed source

- Microsoft DNS server
- Nominum/Akamai
- Xerocole/Akamai
- Secure64
- Infoblox (mostly a front end on top of BIND, with extensions)

Authoritative servers



Authoritative servers

- In principle these are straightforward: Load zone file and answer requests
- Life is never so simple, especially in such a long-lived protocol
- Years of evolution have led to heaps of configuration options and tunable behaviour
- Split-horizon, ACLs, “Geolocation” and general DNS protocol featuritis
- Storage backends in SQL DBs, etc

Authoritative servers

Open source

- Just like for resolvers most DNS software has been Open Source software
  - BIND 9
  - NSD
  - Knot DNS
  - PowerDNS Authoritative server

Authoritative servers

Closed source

- Akamai (ex-nominum)
- Secure64
- Microsoft DNS: mostly relevant in networks where Active Directory is used as it supports the required (extended) GSS-TSIG mechanism, based on RFC 2078
- Infoblox

Additional bits and pieces

DNS “load balancers”

- Loosely defined devices or applications that aim to present a front towards the network for a series of not-directly accesible backend servers
  - Embedded in traditional load balancer devices (e.g. F5)
    - Most feel like afterthoughts (oh, #!$, we need to support DNS)
  - Open source: dnsdist (sweet!) For the DNS by the DNS
  - Routing techniques (e.g. ECMP) that rely on non-DNS software

Additional bits and pieces

DNS Interceptors

- Because DNS is your gateway to the net, it is frequently used as a control point
  - Captive portals (e.g. hotel networks)
  - Censorship
  - “optimisation”
- All of these are available as part of DNS servers or as extension modules