How Stuff Works: The Plumbing of the Internet

- 10 minutes presentation on a small and specific topic, maybe overlooked but important for the functioning of the internet
DNS Upstream Server Selection
Resolving Names: At First Sight

Diagram:
- Resolver
  - www.example.com?
  - .example.com?
  - .com?
  - .com
  - .example.com
Resolving Names: After a Closer Look

multiple name servers for a domain
Multiple Name Servers for a Domain

```bash
pi@raspberrypi:~ $ drill -t ns .
;; -----BEGIN NS-- opcode: QUERY, rcode: NOERROR, id: 54130
;; flags: qr rd ra ; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;
;; ANSWER SECTION:
.
  508090 IN  NS  f.root-servers.net.
  508090 IN  NS  l.root-servers.net.
  508090 IN  NS  h.root-servers.net.
  508090 IN  NS  b.root-servers.net.
  508090 IN  NS  a.root-servers.net.
  508090 IN  NS  d.root-servers.net.
  508090 IN  NS  k.root-servers.net.
  508090 IN  NS  e.root-servers.net.
  508090 IN  NS  g.root-servers.net.
  508090 IN  NS  m.root-servers.net.
  508090 IN  NS  c.root-servers.net.
  508090 IN  NS  i.root-servers.net.
  508090 IN  NS  j.root-servers.net.
;
;; AUTHORITY SECTION:
;
;; ADDITIONAL SECTION:
```
DNS Upstream Server Selection

- Different requirements, not always aligned
- performance: obvious
- probing: discovering (new) name servers
- benign: no DDoS by unintended aggressive querying of name servers
- security: variance in server selection for on-path DNS spoofing attacks

DNSSEC?
Unbound Server Selection

- Basic upstream selection process
  - sort IP-addresses (IPv4, IPv6) by timeout (RTO)
  - pick server random within RTT band (400 ms)
- If timeout occurs
  - update timeout values (next slide)
  - start server selection again

Finding IP addresses
- A, AAAA in NS records
- fetches promiscuously for future use
- servers get default value 376 ms

```c
/* overwritten by config: infra_cache_min_rtt: */
int RTT_MIN_TIMEOUT = 50;
/** calculate RTO from rtt information */
static int
calc_rto(const struct rtt_info* rtt)
{
    /* From Stevens, Unix Network Programming, Voll, 3rd ed., p.598 */
    int rto = rtt->srtt + 4*rtt->rttvar;
    if(rto < RTT_MIN_TIMEOUT)        rto = RTT_MIN_TIMEOUT;
    if(rto > RTT_MAX_TIMEOUT)        rto = RTT_MAX_TIMEOUT;
    return rto;
}
Timeout Parameter Calculation

- Complexity of upstream name server selection is a multidimensional problem mapped to one metric

- Timeout values are determined with regular queries to name server
  - smoothed average RTT times of successful queries ✅
  - increases due to packet loss, doubling timeout ❌
  - penalty for servers that exhibit problems ❌
  - e.g., no-DNSSEC (missing RRSIGs), no EDNS0 support
Probing Unresponsive Servers

- Probing regime when server becomes unresponsive
  - timeout (with exponential backoff) exceeds 12 seconds
  - two (or more) consecutive exponential backoffs have just been done
- In probing regime, server is queried cautiously with regular requests (+ self-exclusion)
  - with little traffic, single request will probe the different servers
  - moderate traffic, several requests will pick up different address and probe
  - high traffic, all servers for domain probed at the same time
- if more queries arrive, resolver answers with SERVFAIL
Blocking Regime

• Upstream is in blocking regime when timeout reached 120 seconds

• If no other working server exists, all queries are answered with SERVFAIL

• Blocking status is cached until infra-ttl (default 15 mins)

• After infra-ttl expired, **one** probe query is send to server in 15 mins

  • full probe sequence would take about 240 seconds (sequence of exponential backoffs until it is 120 seconds)
Config Options to Tweak Selection Process

• Default behaviour of Unbound upstream selection process
  • robust and safe
  • behaves “well” in the grand scheme of things (aka The Internet)
• But Unbound is also used in/with
  • CDNs/Clouds/Datacenters: performance preference with fast-server-permil & fast-server-num
  • satellite uplinks with 500 msec delays: increase initial timeout for packets
Concluding

• Proper upstream server selection is important for
  • DNS performance (end-user experience, responsiveness, etc.)
  • stability of the Internet (millions of instances of software!)
• Not standardised
  • Implementations for the various resolvers (e.g. BIND, Knot Resolver, PowerDNS Recursor, and Unbound) take different considerations into their design decisions
• Background article on Unbound timeout: https://www.nlnetlabs.nl/documentation/unbound/info-timeout/


• Moritz Müller, Giovane Moura, Ricardo de O. Schmidt, and John Heidemann, “Recursives in the Wild: Engineering Authoritative DNS Servers”, in Proceedings of ACM Internet Measurement Conference (IMC ’17), London, United Kingdom, November 2017