

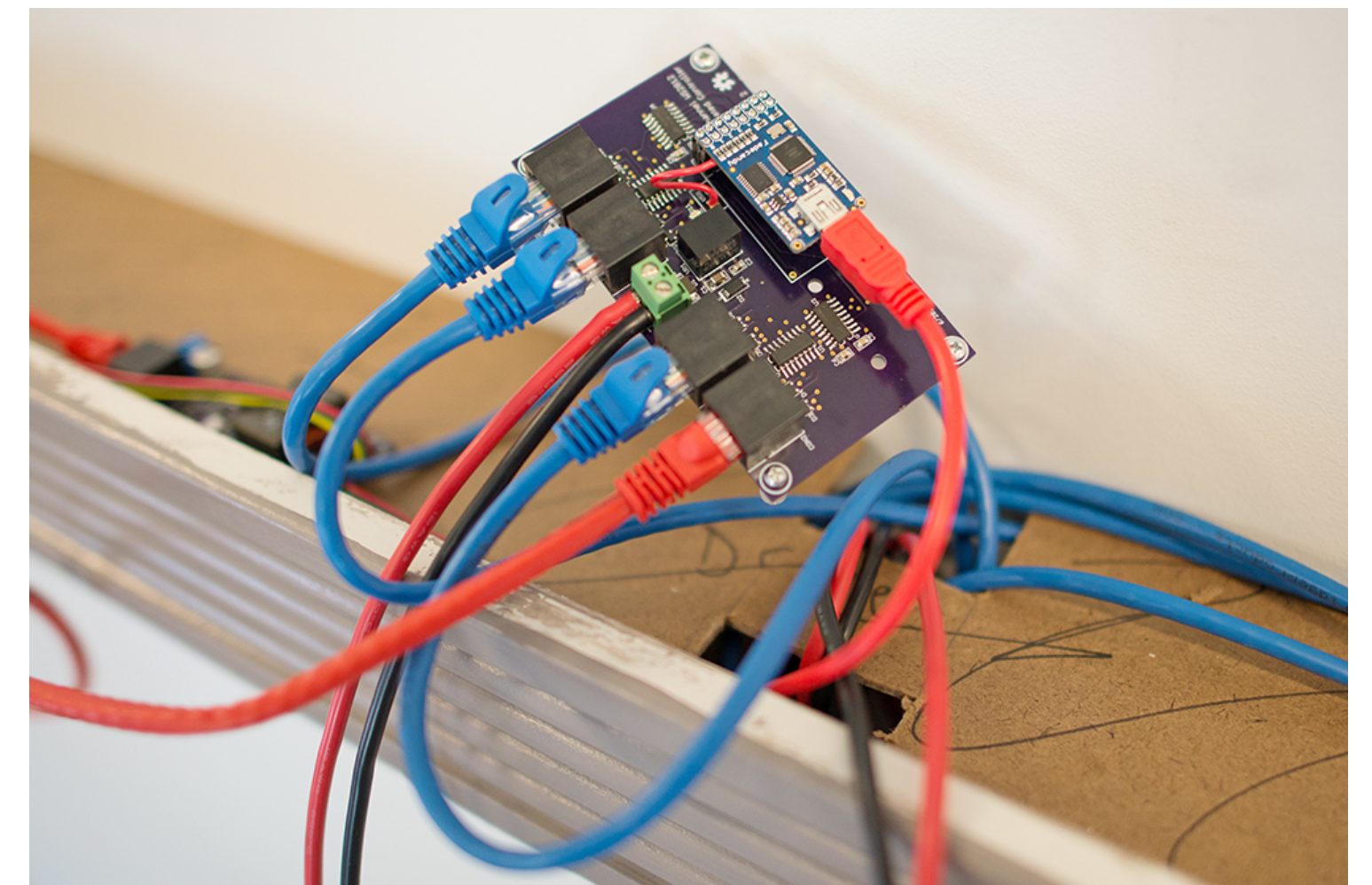
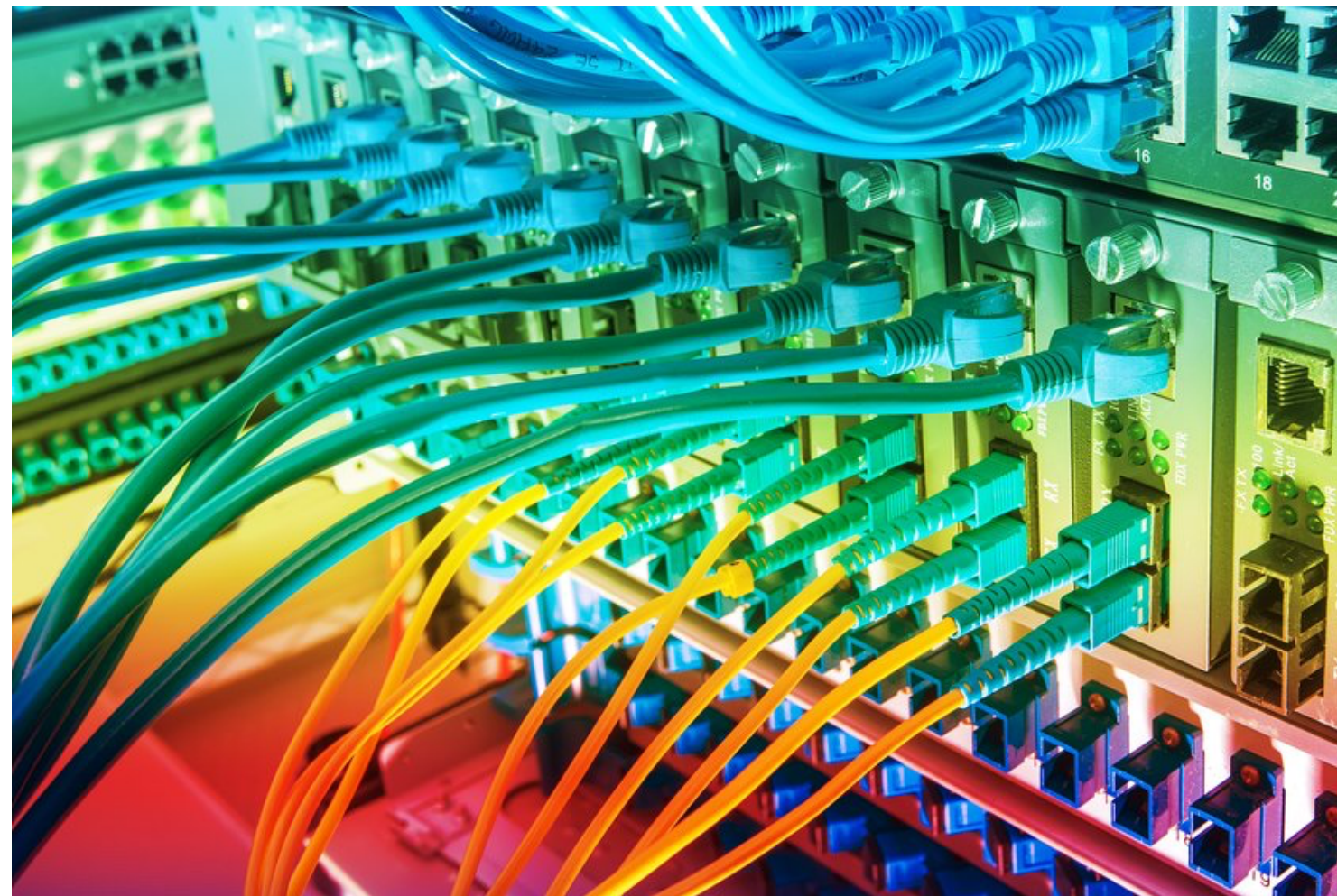


How Stuff Works: DNS Upstream Server Selection

Benno Overeinder
NLnet Labs

How Stuff Works: The Plumbing of the Internet

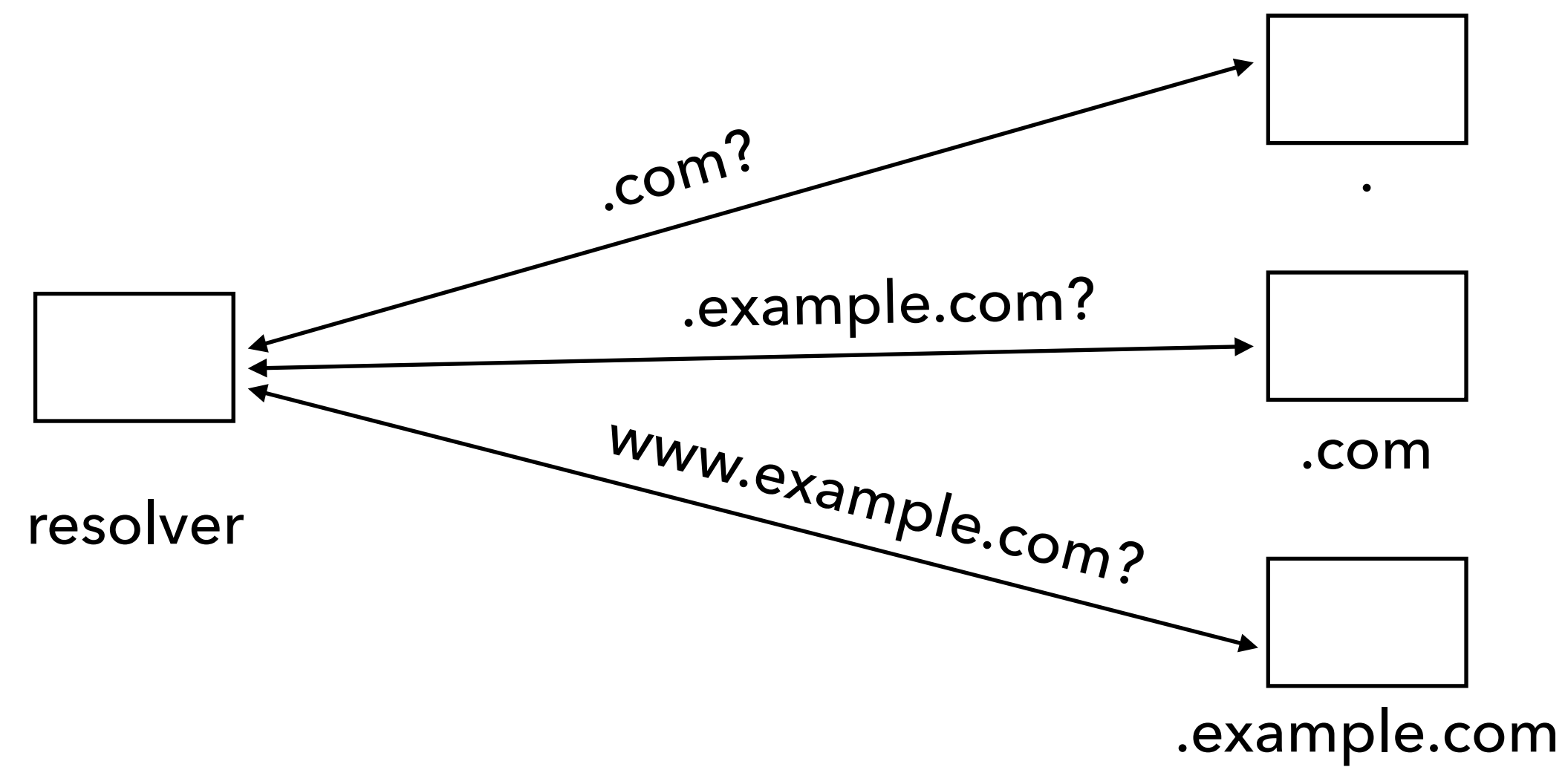
- 10 minutes presentation on a small and specific topic, maybe overlooked but important for the functioning of the internet



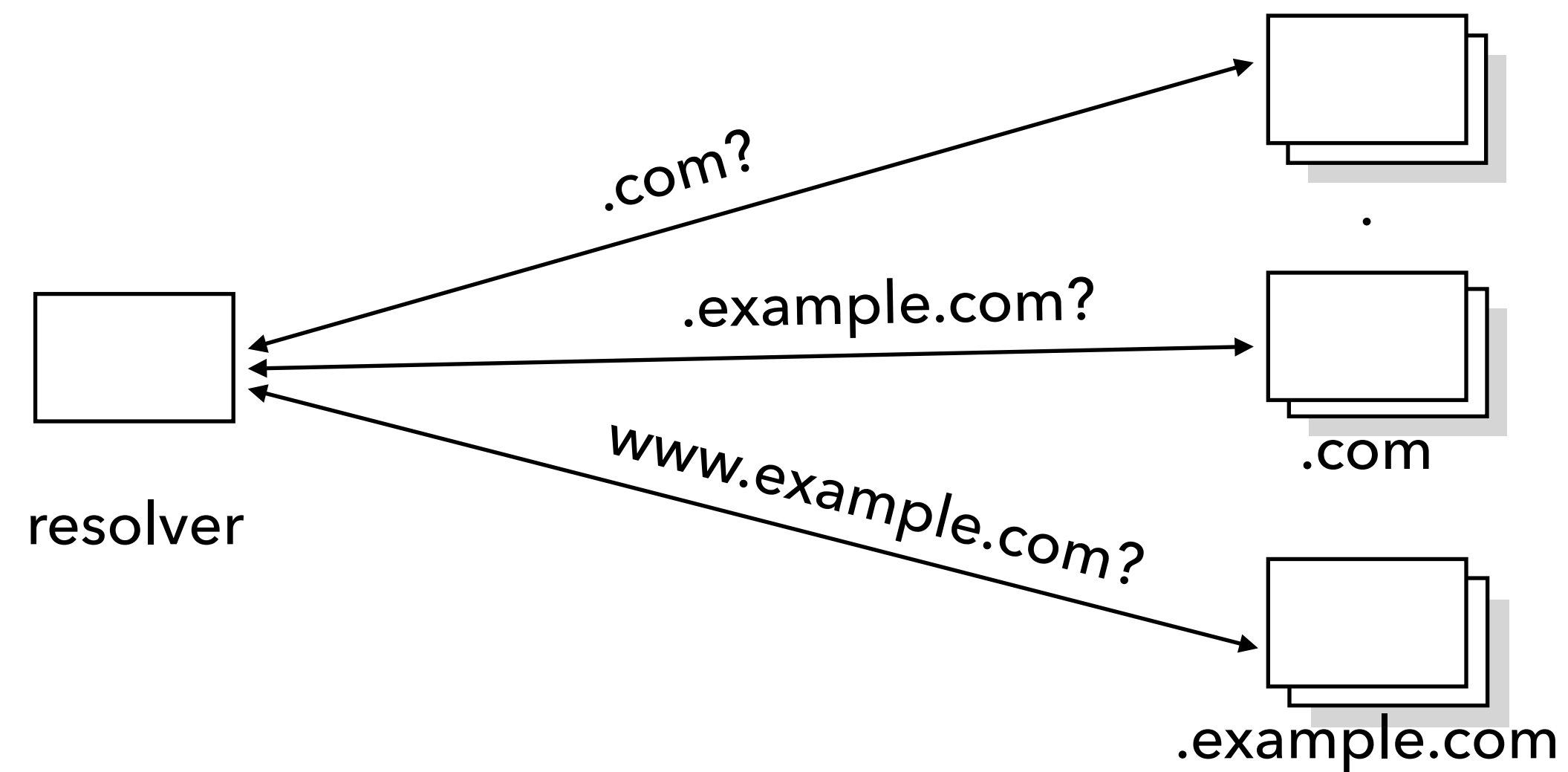
DNS Upstream Server Selection




Resolving Names: At First Sight



Resolving Names: After a Closer Look



multiple
name
servers
for a
domain



Multiple Name Servers for a Domain

```
[pi@raspberrypi:~ $ drill -t ns .  
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 54130  
;; flags: qr rd ra ; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 0  
;; QUESTION SECTION:  
;; .      IN      NS
```

```
;; 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:
```

```
[pi@raspberrypi:~ $ drill -t ns com.  
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 17831  
;; flags: qr rd ra ; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 0  
;; QUESTION SECTION:  
;; com.  IN      NS
```

```
;; ANSWER SECTION:  
com.  172565  IN      NS      d.gtld-servers.net.  
com.  172565  IN      NS      m.gtld-servers.net.  
com.  172565  IN      NS      b.gtld-servers.net.  
com.  172565  IN      NS      l.gtld-servers.net.  
com.  172565  IN      NS      e.gtld-servers.net.  
com.  172565  IN      NS      a.gtld-servers.net.  
com.  172565  IN      NS      i.gtld-servers.net.  
com.  172565  IN      NS      k.gtld-servers.net.  
com.  172565  IN      NS      h.gtld-servers.net.  
com.  172565  IN      NS      c.gtld-servers.net.  
com.  172565  IN      NS      g.gtld-servers.net.  
com.  172565  IN      NS      f.gtld-servers.net.  
com.  172565  IN      NS      j.gtld-servers.net.
```

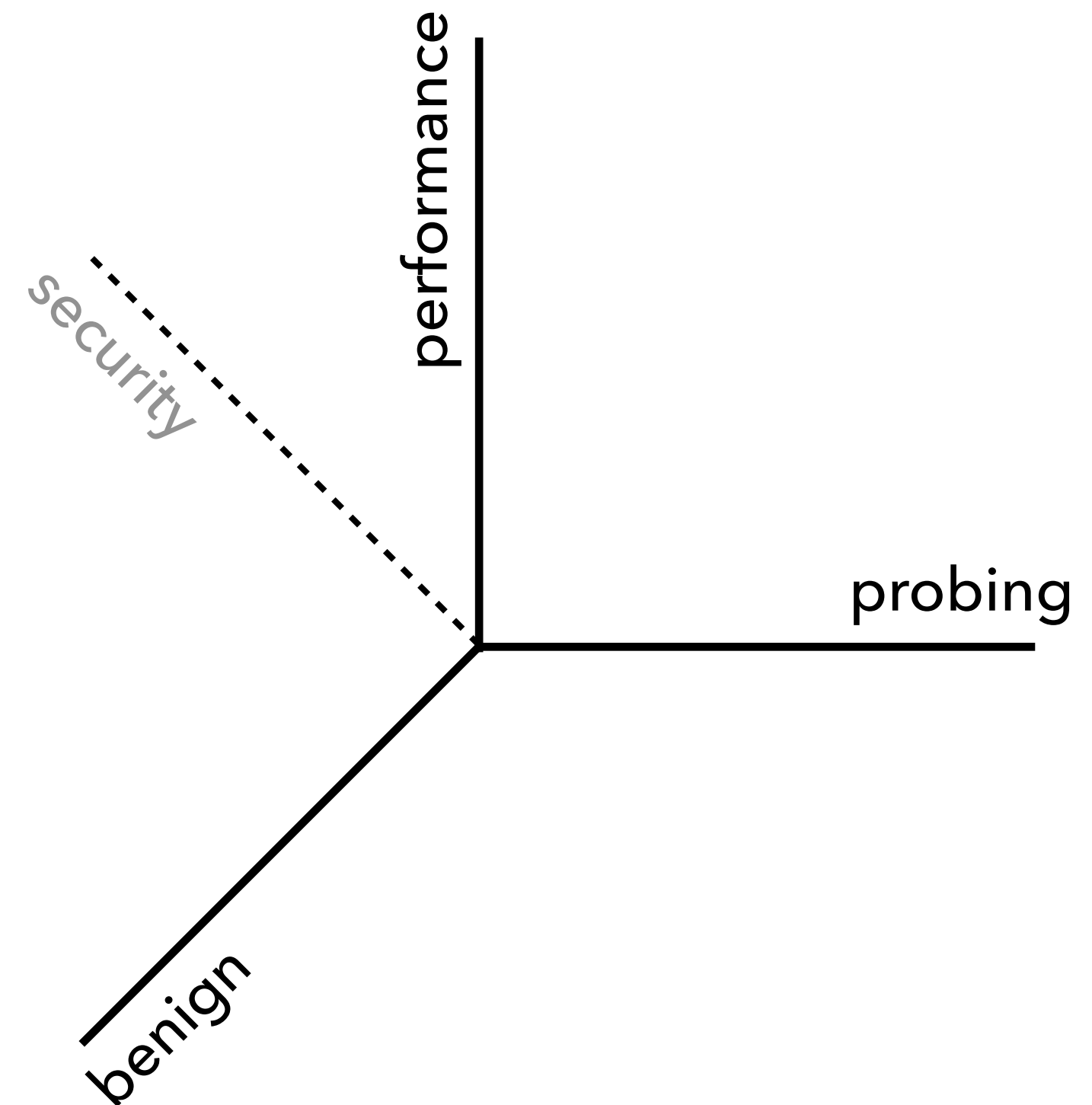
```
;; AUTHORITY SECTION:
```

```
;; 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

```
/* 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, Vol1, 3rd ed., p.598 */
    int rto = rtt->srtd + 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
 - **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

Reading More ...

- Background article on Unbound timeout:
<https://www.nlnetlabs.nl/documentation/unbound/info-timeout/>
- Yingdi Yu, Duane Wessels, Matt Larson, and Lixia Zhang, "Authority Server Selection of DNS Caching Resolvers", *ACM SIGCOMM Computer Communication Review*, Vol. 42, No. 2, April 2012
- 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