Filtering address records in stub resolvers

draft-caletka-aaaa-filtering-01

What and why

- There are still people **stuck on IPv4-only networks**
- We want to run internal services **IPv6-only** via **VPN**
- We have fully working **IPv6 reachability via VPN**
- Yet **DNS says No!**

It even pings!

```
$ ping6 www.doesnotwork.eu
PING6(56=40+8+8 bytes) 2001:67c:2e8:1200::b20 --> 2001:1528:132:70::d0e5
16 bytes from 2001:1528:132:70::d0e5, icmp_seq=0 hlim=56 time=99.158 ms
16 bytes from 2001:1528:132:70::d0e5, icmp_seq=1 hlim=56 time=148.704 ms
^C
--- www.doesnotwork.eu ping6 statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 99.158/123.931/148.704/24.773 ms

$ curl https://www.doesnotwork.eu
curl: (6) Could not resolve host: www.doesnotwork.eu
```

Why are some resolvers not doing AAAA queries?

Possible reasons:

- If your IPv6 connectivity is Teredo then you rather not use it (Windows)
- Some poorly written resolvers might crash when they see AAAA query (Chrome)
    - This claim didn’t age well with Chrome now **doing queries for HTTPS RRs**

This draft sets the rules to do the query filtering **properly**

- Not only **AAAA** but also **A** records (for pure IPv6-only networks)
- Two algorithms:
  - Look into the **routing table** (including **connected routes**)
  - Look at **IP addresses** configured on all interfaces
- Do not consider **just default route** and/or **default interface**

Next steps

- Questions?
- Comments?
- Adoption?