

Happy Eyeballs, Version 3

draft-pauly-v6ops-happy-eyeballs-v3

Tommy Pauly, David Schinazi,
Nidhi Jaju, Kenichi Ishibashi
V6OPS
IETF 118, November 2023, Prague



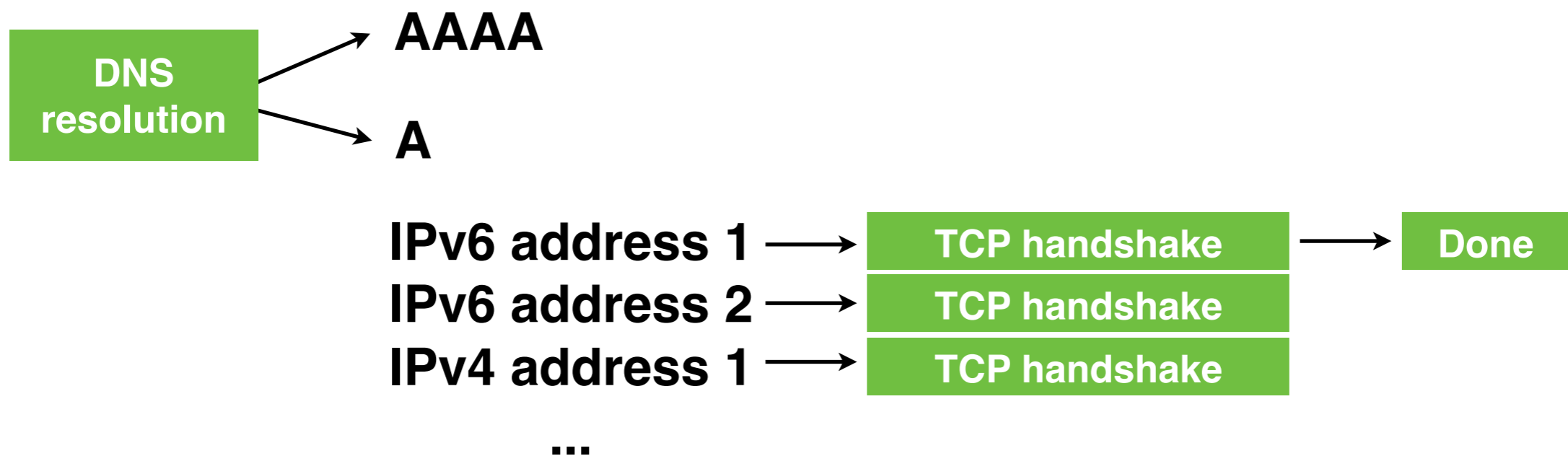
Why a new version?

Lots of changes to DNS and transports since RFC 8305!

- SVCB / HTTPS records
 - Address hints
 - Priorities
 - ALPN
- QUIC standardization
- Encrypted Client Hello

Existing algorithm

- Query AAAA/A asynchronously
- Prioritize AAAA over A
- Sort addresses and stagger connection attempts
- Race until TCP handshake completes



Algorithm with SVCB

- Query SVCB/A/AAAA/A RRs in parallel
- Use SVCB priority to sort associated A/AAAA answers and address hints
 - A/AAAA answers not associated with SVCB are prioritized last
 - Prefer ECH keys if present
- Sort endpoints (address, protocol, ECH keys) and stagger connection attempts
- Race until full handshakes complete (TCP / TLS / QUIC, etc)

Sorting Algorithm

Preferences:

1. Prefer ECH keys, if present
2. SvcPriority, if present
3. Preferred ALPNs, if present
4. Destination Address Selection (RFC6724)
5. Historical RTTs
6. Adjust to sort (2) IPv6 addresses to the front

Generalizing for QUIC

- Prefer services with QUIC-capable ALPNs when sorting endpoints, after ECH keys and SvcPriority
 - QUIC provides improved delivery and congestion control, connection migration, etc.
- Adjust connection establishment logic to not just mention TCP
 - Race until QUIC completes
 - Also allow racing until TLS above TCP completes

ECH considerations

- If client is SVCB-optional,
 - May start a TCP handshake, but not TLS/QUIC
 - Wait until a timeout for "ech" SvcParamKey
 - Is it reasonable to proceed if the timer expires?
- If client is/becomes SVCB-reliant,
 - Wait until "ech" SvcParamKey to start TLS/QUIC handshake
 - Is it safe to start a TCP handshake?

Next steps

Continue to refine logic around ECH, ALPN, etc

Implementations

Apple implements the basic SVCB logic already
(for a couple years)

Let's implement and get data!

Adopt in v6ops?