

Optimistic DNS

draft-gakiwate-dnsop-optimistic-dns

Gautam Akiwate, Phil Flack, Stuart Cheshire
DNSOP WG
July 24, 2026

Optimistic DNS

A client-side stub resolver mechanism
that immediately returns
expired cached DNS records
while simultaneously
refreshing it with a network query.*

Optimistic DNS

A **client-side stub resolver mechanism**
that immediately returns
expired cached DNS records
while simultaneously
refreshing it with a network query.*

Optimistic DNS

A **client-side stub resolver mechanism**

that immediately returns

expired cached DNS records

while simultaneously

refreshing it with a network query.*

Complementary to RFC 8767 - Serving State Data

Motivation

Post-Expiry Latency Spike

Time (Seconds)	Response Latency	Notes
t=0	150 ms	First Lookup (TTL = 60s)
t=15	~0 ms	Cache Hit
t=59	~0 ms	Cache Hit
t=61	150 ms	Expired (New Query)

- Record is valid for 60 seconds, then invalid for the fraction of a second it takes to refresh it.
- Yet that fraction of a second is the one the user notices because it causes intermittent slowness.
- *In most common scenario, the answer has not changed.* We unnecessarily take the latency hit every time a record expires.
- Proactive refreshing does not work at the stub resolver.

“Why Not Try” Philosophy

- We have an expired record. Send DNS query to refresh the record.
- Why not use the expired record while we wait?

“Why Not Try” Philosophy

- We have an expired record. Send DNS query to refresh the record.
- Why not use the expired record while we wait?
 - It might work, it might not.

“Why Not Try” Philosophy

- We have an expired record. Send DNS query to refresh the record.
- Why not use the expired record while we wait?
 - It might work, it might not.
 - *If it does work*, then visibly better user experience.

“Why Not Try” Philosophy

- We have an expired record. Send DNS query to refresh the record.
- Why not use the expired record while we wait?
 - It might work, it might not.
 - *If it does work*, then visibly better user experience.
 - *If it does not work*, then we wait for DNS query and use that record.
 - No worse than just waiting for the DNS query to finish.

“Why Not Try” Philosophy

- We have an expired record. Send DNS query to refresh the record.
- Why not use the expired record while we wait?
 - It might work, it might not.
 - *If it does work*, then visibly better user experience.
 - *If it does not work*, then we wait for DNS query and use that record.
 - No worse than just waiting for the DNS query to finish.

Client should gracefully fallback and use the “fresh” answer to connect if expired answer does not work.

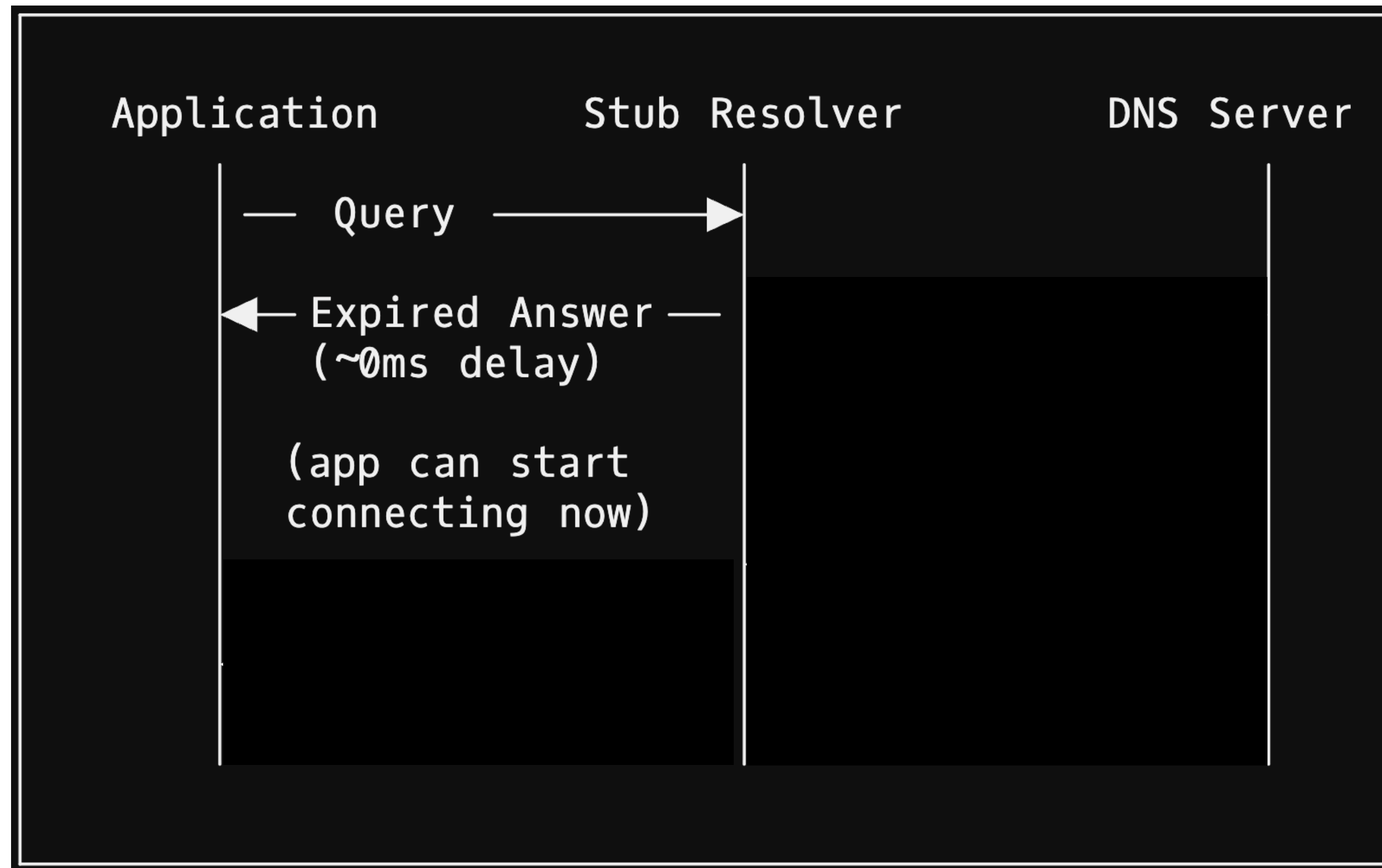
Enabling Technologies

Prerequisites for Optimistic DNS

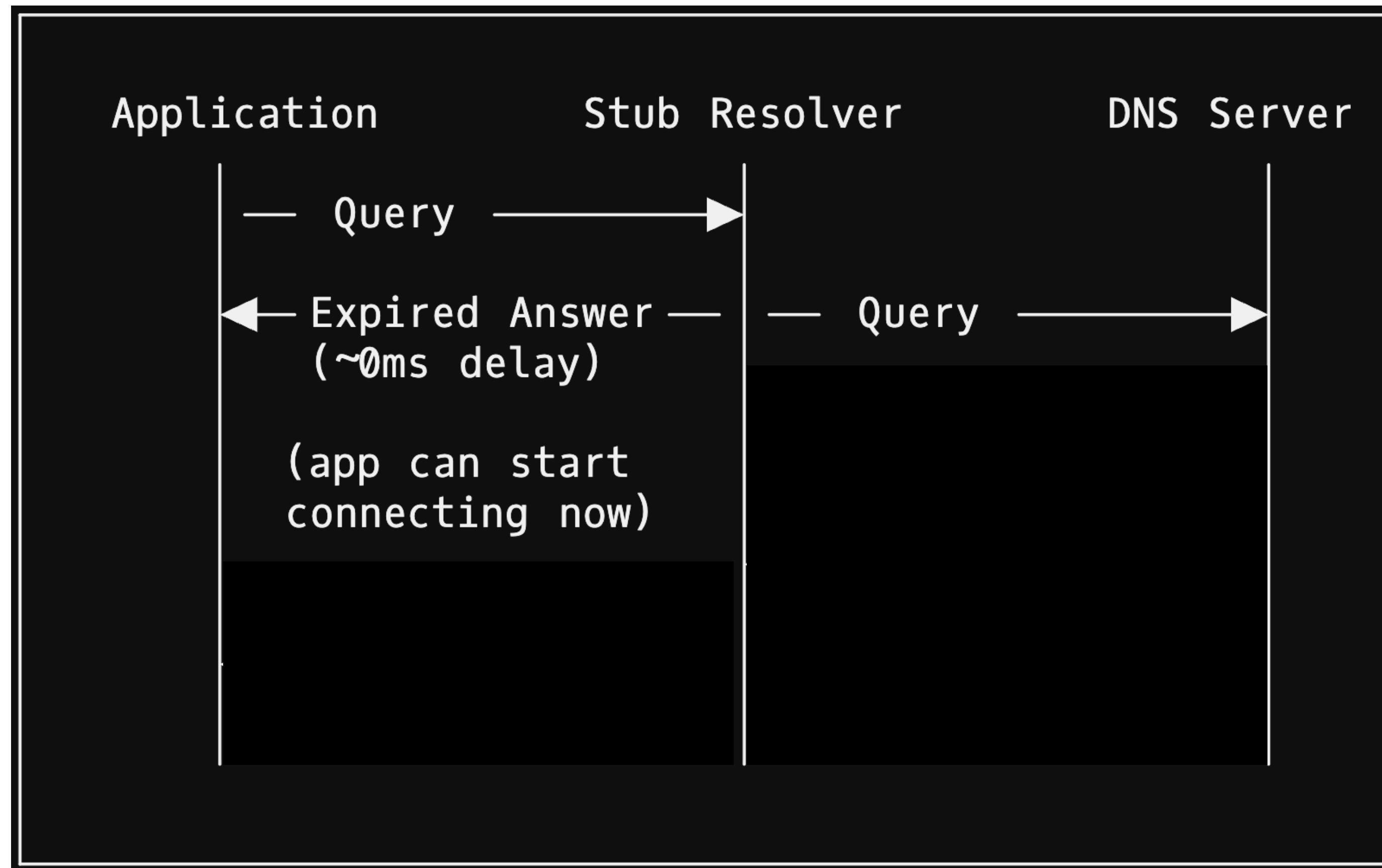
- Asynchronous DNS
 - The application continues getting updates to the DNS query.
- Happy Eyeballs
 - The application continues trying new addresses till a connection is successfully made i.e., a failed attempt is not fatal to the application.

If a client cannot gracefully use the new addresses do not use expired records.

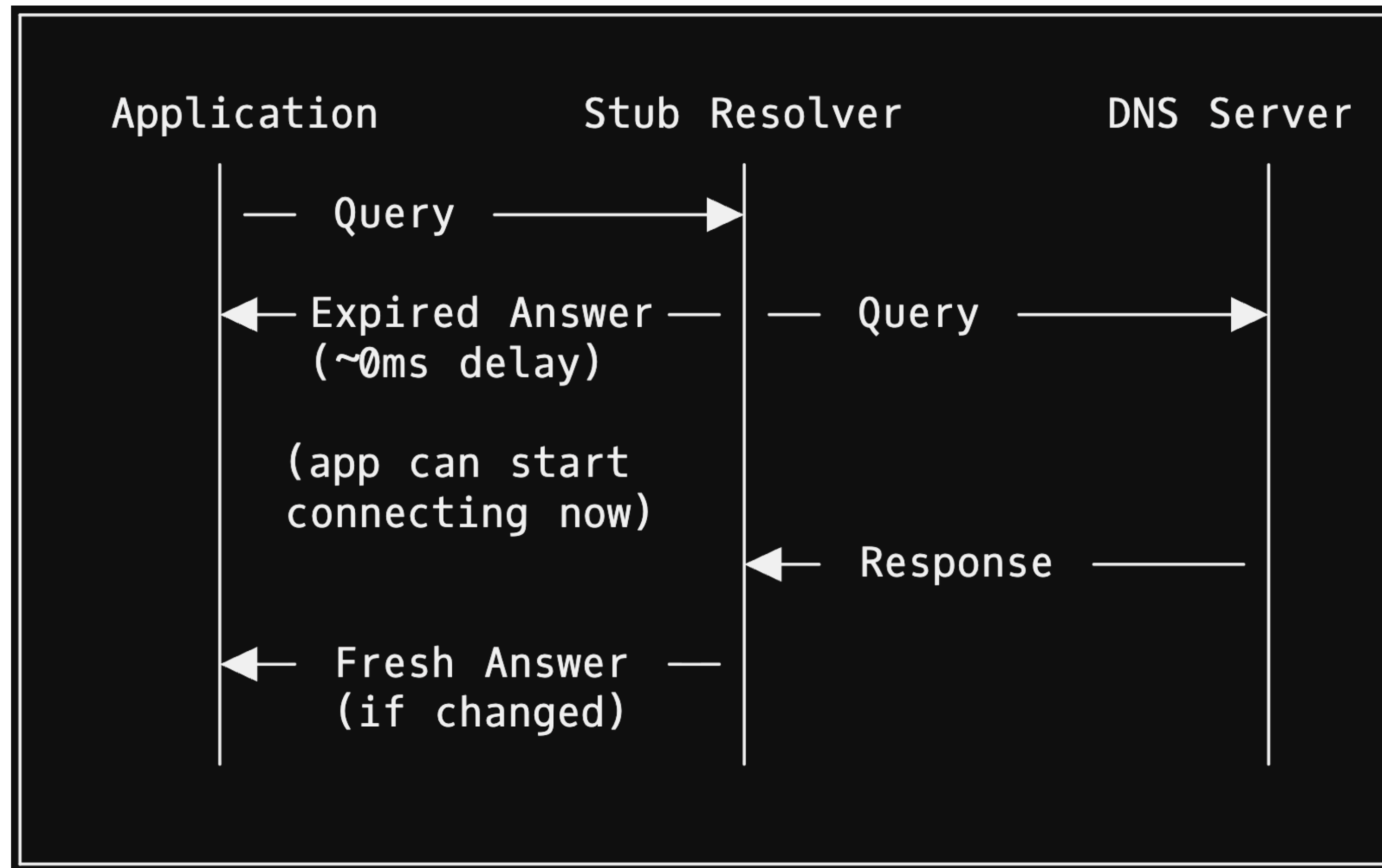
Optimistic DNS



Optimistic DNS



Optimistic DNS



Asynchronous DNS makes
Optimistic DNS *possible*.

Happy Eyeballs makes
Optimistic DNS *safe*.

How Long to Use Expired Records

“Why Not Try” Philosophy

- 5 seconds after — Why Not Try...

How Long to Use Expired Records

“Why Not Try” Philosophy

- 5 seconds after — Why Not Try...
- 5 minutes after — Why Not Try...

How Long to Use Expired Records

“Why Not Try” Philosophy

- 5 seconds after — Why Not Try...
- 5 minutes after — Why Not Try...
- 5 hours later — Why Not Try...

How Long to Use Expired Records

“Why Not Try” Philosophy

- 5 seconds after — Why Not Try...
- 5 minutes after — Why Not Try...
- 5 hours later — Why Not Try...
- 5 days later — Why Not Try...
- ... — Why Not Try

How Long to Use Expired Records

“Why Not Try” Philosophy

- 5 seconds after — Why Not Try...
- 5 minutes after — Why Not Try...
- 5 hours later — Why Not Try...
- 5 days later — Why Not Try...
- ... — Why Not Try

*An expired record is only used once and refreshed immediately.
Caches are purged on network change.
In practice, expired records are not used long after.*

Implementations

- mDNSResponder implements Optimistic DNS
 - September 2018
 - macOS 10.14 (Mojave) and iOS 12
- Mozilla Firefox
 - Uses Expired Records since 2013
- Google
 - In Progress

Operational Considerations

- Optimistic DNS optimizes for the most common scenario.
- Clients will gracefully fallback and use “fresh” answers.
- Servers can rightfully reject clients that are using expired records and clients “will do the right thing”

Operational Considerations

- Optimistic DNS optimizes for the most common scenario.
- Clients will gracefully fallback and use “fresh” answers.
- Servers can rightfully reject clients that are using expired records and clients “will do the right thing”
- In scenarios when operational changes are expected:
 - Maybe a SVCB / HTTPS SvcParam that limits use of optimistic DNS?
 - Explicit Opt Out.

Next Steps

- Dispatch?
 - HAPPY WG or DNSOP WG?