

Happy Eyeballs v3 in Firefox

Max Inden - Mozilla - IETF 125

Available in Firefox Nightly

You can try it out by:

1. Installing [Firefox Nightly](#)
2. Enabling Happy Eyeballs v3 by visiting `about:config` setting `network.http.happy_eyeballs_enabled` to `true`
3. Visiting any website, e.g. <https://www.happy-eyeballs.net/>

(Consider [enabling DoH](#) to get DNS HTTPS RR on all platforms.)

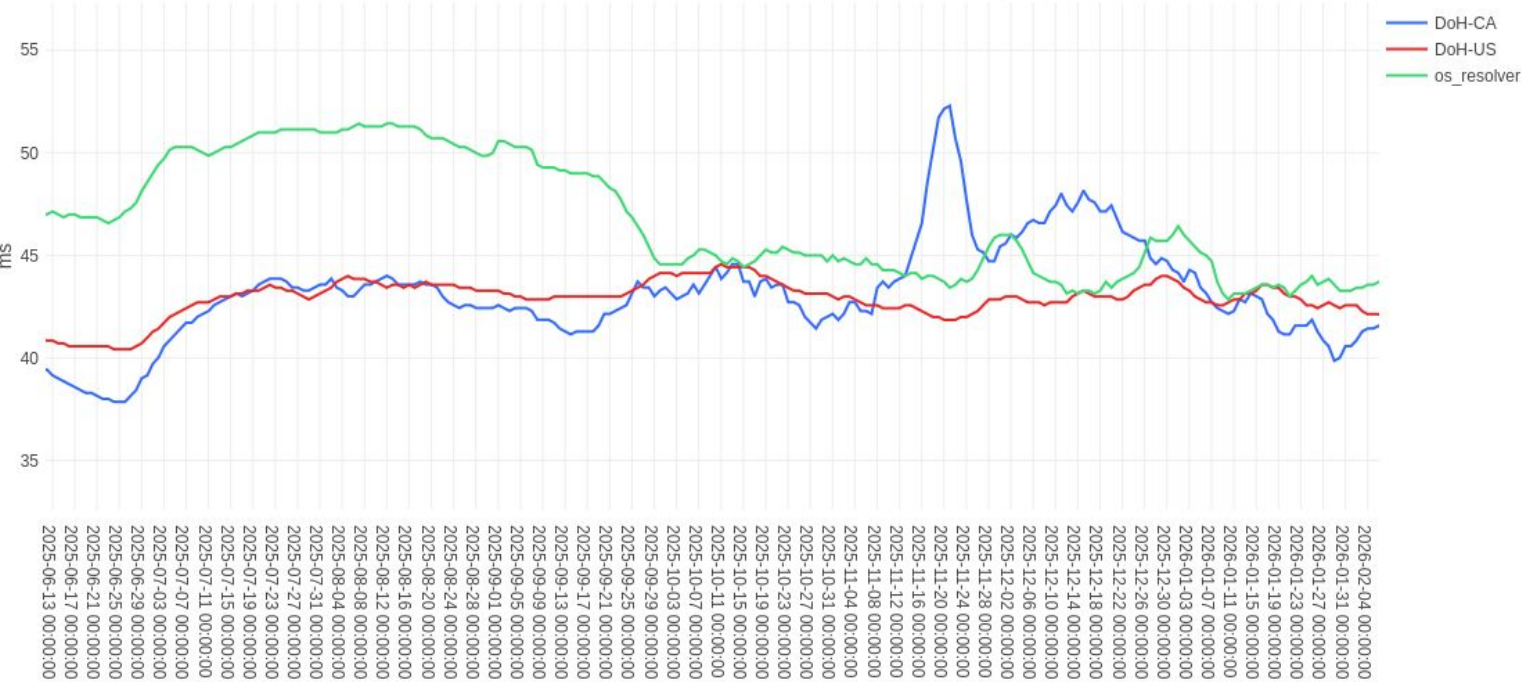
Status

- In Firefox Nightly, disabled by default
- No major deviation from the Draft
- Likely to tune the resolution delay and connection establishment latency based on Firefox ~99th percentile
- Considering adding Proxy awareness
 - a. Proxy connection
 - b. Proxy protocol
 - c. Target connection

Firefox Telemetry

- Public dataset
- Upcoming metrics
 - Resolution delay by record type
 - Number of connection attempts
 - Connection outcome (http version, IP version, ech, ...)
 - Suggestions?
- Required to align with Mozilla's [privacy stance](#)

DNS Lookup - 75th percentile

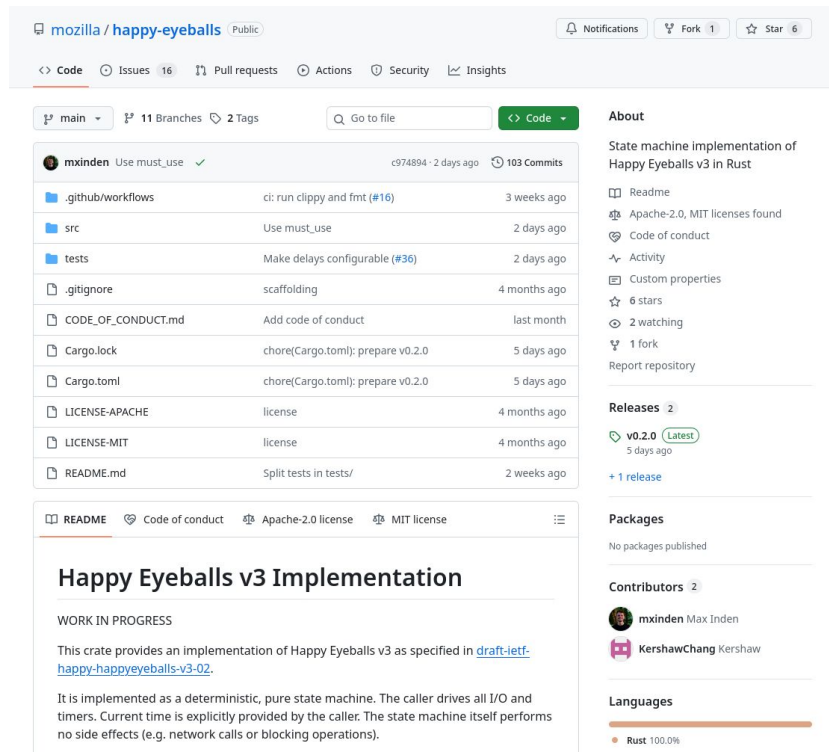


Connection Establishment Latency

5th	11 ms
25th	37 ms
50th	73 ms
75th	123 ms
95th	537 ms
99th	1.7 s
99.9th	11 s

Independent Library

- Rust library [mozilla/happy-eyeballs](https://github.com/mozilla/happy-eyeballs)
- Deterministic, no side-effects, abstract over I/O and time
- No dependencies on Firefox
- Permissive license (MIT and Apache)
- Input, questions and contributions welcome.



The screenshot shows the GitHub repository page for mozilla/happy-eyeballs. The repository is public and has 103 commits and 1 fork. The main branch is selected. The repository contains several files and folders, including .github/workflows, src, tests, .gitignore, CODE_OF_CONDUCT.md, Cargo.lock, Cargo.toml, LICENSE-APACHE, LICENSE-MIT, and README.md. The README file is open, showing the title "Happy Eyeballs v3 Implementation" and the text "WORK IN PROGRESS". The README describes the crate as a deterministic, pure state machine implementation of Happy Eyeballs v3, implemented as a Rust library. The repository also has a "Releases" section with a v0.2.0 release and a "Packages" section with no packages published. The repository is licensed under Apache-2.0 and MIT.

mozilla / happy-eyeballs Public

<> Code Issues 16 Pull requests Actions Security Insights

main 11 Branches 2 Tags Go to file Code

mxinden Use must_use ✓ c974894 · 2 days ago 103 Commits

.github/workflows	ci: run clippy and fmt (#16)	3 weeks ago
src	Use must_use	2 days ago
tests	Make delays configurable (#36)	2 days ago
.gitignore	scaffolding	4 months ago
CODE_OF_CONDUCT.md	Add code of conduct	last month
Cargo.lock	chore(Cargo.toml): prepare v0.2.0	5 days ago
Cargo.toml	chore(Cargo.toml): prepare v0.2.0	5 days ago
LICENSE-APACHE	license	4 months ago
LICENSE-MIT	license	4 months ago
README.md	Split tests in tests/	2 weeks ago

README Code of conduct Apache-2.0 license MIT license

Happy Eyeballs v3 Implementation

WORK IN PROGRESS

This crate provides an implementation of Happy Eyeballs v3 as specified in [draft-ietf-happy-happyeyeballs-v3-02](#).

It is implemented as a deterministic, pure state machine. The caller drives all I/O and timers. Current time is explicitly provided by the caller. The state machine itself performs no side effects (e.g. network calls or blocking operations).

About

State machine implementation of Happy Eyeballs v3 in Rust

- Readme
- Apache-2.0, MIT licenses found
- Code of conduct
- Activity
- Custom properties
- 6 stars
- 2 watching
- 1 fork
- Report repository

Releases 2

- v0.2.0 (Latest) 5 days ago
- + 1 release

Packages

No packages published

Contributors 2

- mxinden Max Inden
- KershawChang Kershaw

Languages

- Rust 100.0%

Test Suite

- Easy to formalize Happy Eyeballs v3 algorithm
- Helpful for working group?

```
#[test]
fn connection_attempt_delay() {
    let (mut now, mut he) = setup();

    // Each tuple is an (input, output) pair:
    // - input: an event fed *into* the state machine
    // - output: the next action the state machine asks *us*
    //   to perform
    he.expect(
        vec![
            // State machine asks us to send DNS queries.
            (None, Some(out_send_dns_https(Id::from(0)))),
            (None, Some(out_send_dns_aaaa(Id::from(1)))),
            (None, Some(out_send_dns_a(Id::from(2)))),
            // Feed HTTPS positive response, state machine waits
            // for more.
            (
                Some(in_dns_https_positive_no_alpn(Id::from(0))),
                Some(out_resolution_delay()),
            ),
            // Feed AAAA positive response, state machine attempts
            // a connection.
            (
                Some(in_dns_aaaa_positive(Id::from(1))),
                Some(out_attempt_v6_h1_h2(Id::from(3))),
            ),
            // Feed A positive response, state machine waits for
            // connection attempt delay.
            (
                Some(in_dns_a_positive(Id::from(2))),
                Some(out_connection_attempt_delay()),
            ),
        ],
        now,
    );
}
```

Next Steps

- Pending work for the Happy Eyeballs v3 Rust library [on GitHub](#)
- Pending work for the Firefox integration [on Mozilla's Bugzilla](#)
- We plan to:
 - enable Happy Eyeballs v3 on Firefox Nightly in the coming weeks
 - ship it in Firefox Release in the coming months
- We can share more telemetry at IETF 126

Questions? Proposals?

- <https://github.com/mozilla/happy-eyeballs/issues/new>
- https://bugzilla.mozilla.org/show_bug.cgi?id=1953459
- mail@max-inden.de
- @mxinden everywhere else