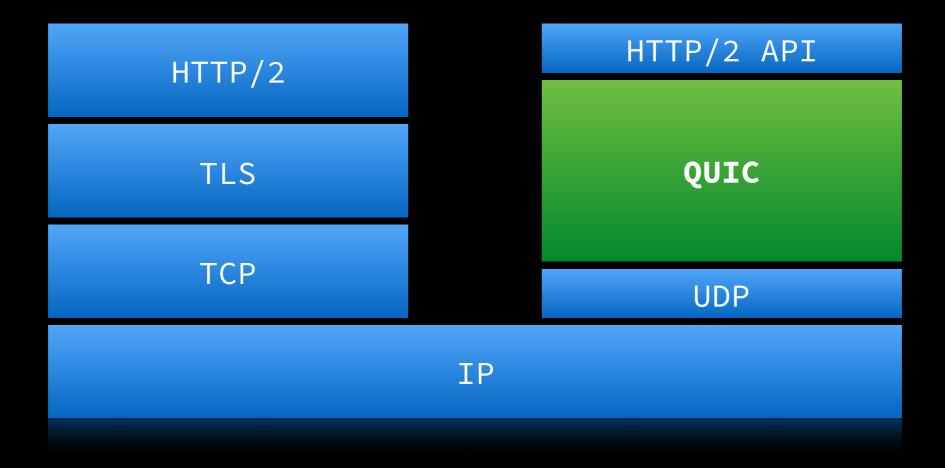# github.com/lucas-clemente/ quic-go

A QUIC server implementation in Go
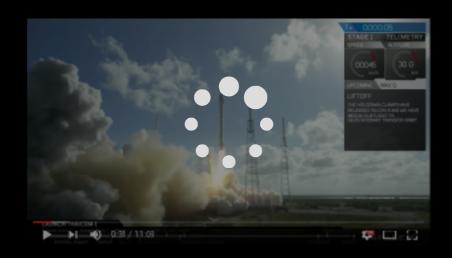
Lucas Clemente
Marten Seemann

you've probably been using QUIC for ~ 3 years

An experimental UDP-based network protocol by Google

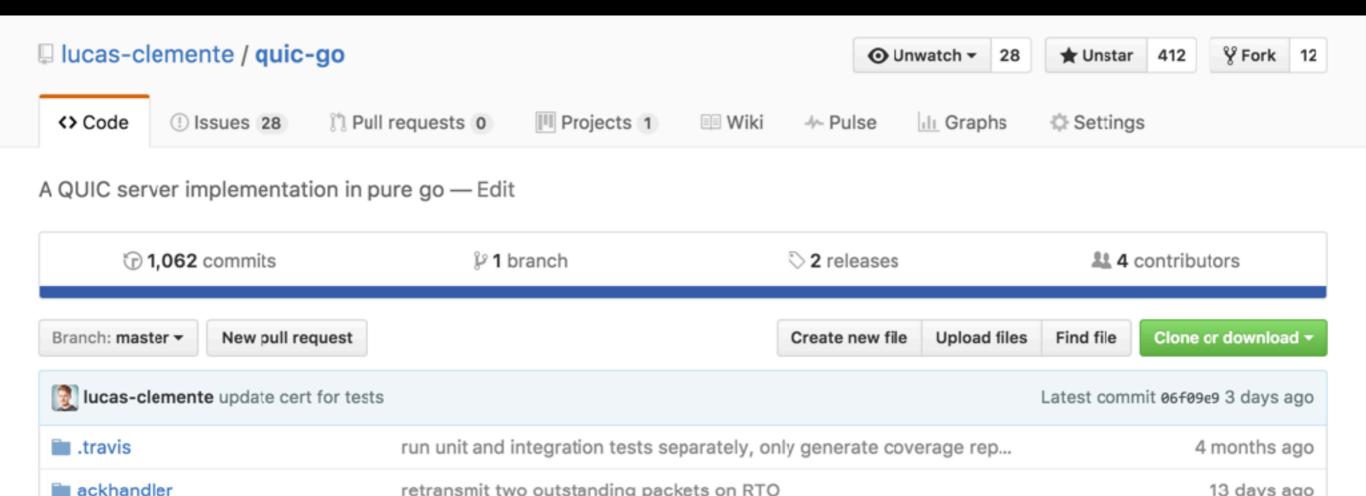0-RTT handshakes

Faster on bad connections

Easier protocol iteration than TCP

30% fewer YouTube rebufferings

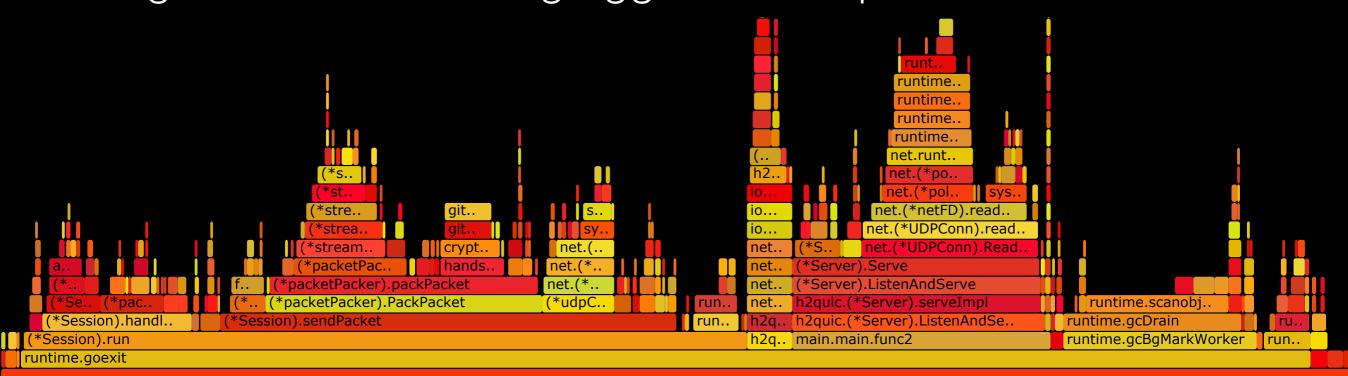**Google Search 1s faster for slowest 1% connections**

quic-go

- Half a year old, 10k LOC, 14k lines of test

- TODOs: Client, more network tests, bugs

lucas-clemente / **quic-go**

👁 Unwatch ▾ 28 ★ Unstar 412 ⑂ Fork 12

<> Code   ⊙ Issues 28   ⑂ Pull requests 0   ⊞ Projects 1   🗏 Wiki   ⌁ Pulse   📊 Graphs   ⚙ Settings

A QUIC server implementation in pure go — Edit

⊙ **1,062** commits   ⑂ **1** branch   🏷 **2** releases   👥 **4** contributors

Branch: master ▾   New pull request   Create new file   Upload files   Find file   **Clone or download ▾**

lucas-clemente update cert for tests   Latest commit 06f09e9 3 days ago

📁 .travis   run unit and integration tests separately, only generate coverage rep…   4 months ago

📁 ackhandler   retransmit two outstanding packets on RTO   13 days ago

Lessons learned

- Network programming is hard 😱

- Integration tests help (dockerized Chrome & UDP proxy)

- Go 1.7 framepointers for optimizing (prof / dtrace)

- github.com/brendangregg/FlameGraph

Try it / help out

- Use Caddy with `-quic` (thanks to Matt Holt!)

- In your own program, replace

```
http.ListenAndServeTLS(":443", "chain.pem", "privkey.pem", nil)
h2quic.ListenAndServe (":443", "chain.pem", "privkey.pem", nil)
```

github.com/lucas-clemente/quic-go