GitHub for Working Groups

WG Chairs Lunch, IETF 102
Martin Thomson
Menu

Why?

Managing Discussion

Established Practices
Why

Source control
Issue management
Transparency
Large community
Source Control

Internet-Drafts that make up the base QUIC specification [https://quicwg.github.io/](https://quicwg.github.io/)

- quic
- http
- tls
- transport
- protocol
- ietf
- standards

- 2,058 commits
- 22 branches
- 62 releases
- 41 contributors

Latest commit f0217bc 2 hours ago

- .cirdeci
- .editorconfig
- .gitignore
- .lint.py

- Defer artifact saving
- Add trim_trailing_whitespace
- Ignore upload files
- And this is what you get when you run things as well
Source Control

TLS 1.3 Specification

2,174 commits
31 branches
30 releases
69 contributors
Issue and Change Management

Issue tracking
Change management (pull requests)
Reviewing tools
Both integrated with revision control
Issues

- **Initial size of dynamic table** - qpack, design
  - #1530 opened 2 hours ago by martinthomson

- **QPACK could use some examples** - qpack, editorial
  - #1454 opened 20 days ago by afrind

- **encoder stream can deadlock** - qpack, design
  - #1420 opened on Jun 7 by martinthomson

- **Static table negotiation** - qpack, design, quicv2
  - #1343 opened on May 10 by MikeBishop

- **Referencing strategy** - qpack, editorial
  - #1129 opened on Feb 21 by martinthomson
Threaded Discussion

HTTP header compression: static table fit for QUIC #904

Lpardue commented on Oct 31, 2017 • edited

The HTTP header compression mechanism is still TBC but as I understand it, the proposals keep the same HPACK static table. This was formed from the "most frequent header fields used by popular web sites", and I wonder if that is still the case for today's web. For example, perhaps we'll see more usage of alt-svc in HTTP/QUIC, in order to keep the session active.

One benefit to keeping the static table would be to allow implementations to reuse of their object representation of the table, however I don't know how realistic that would be.

ianswett commented on Oct 31, 2017

Given it looks like the header compression will share little to no code with H2, I think this is worth considering.
Pull Requests (PRs)
MikeBishop 14 hours ago  Member

"Either ... cannot’ feels awkward. Maybe "Neither packet can be..."?"

Regardless of urgency, I think this covers both cases a little better. It might still be worth mentioning that the previous flight of 0-RTT packets could still be consumed by the server, so the client needs to consider that data as having been sent (and therefore can’t change it).
Transparency

Progress on issues and text is open

History of changes is maintained

Editors can get rapid feedback on changes
Large Community

Tools that are well understood
Makes contributions more accessible
Improved feedback cycle
Managing Discussion

Discussion has a tendency to fracture... badly

Chairs need to decide how they want to manage this
All Discussion on <wg>@ietf.org

This is the “safest” policy

Limit to editorial comments on issues and PRs

Needs constant reminders to steer discussion

Used effectively for several groups
Discussion on GitHub

More natural

Note well implications

Always take big items to the list

Take care to avoid substantive discussion on PRs

... these turn into a giant mess
Discussion Policy

Be very clear with the working group
Send email explaining expectations
Monitor discussion and steer it as needed
Use CONTRIBUTING.md
Sample Contribution Policy

Contributing to QUIC

Anyone can contribute to QUIC; you don’t have to join the Working Group, because there is no “membership” -- anyone who participates in the work, as outlined below, is part of the QUIC Working Group.

Before doing so, it’s a good idea to familiarize yourself with our current issues list and charter. If you’re new to this, you may also want to read the Tao of the IETF.

Be aware that all contributions fall under the “NOTE WELL” terms outlined below.

- Following Discussion
- Raising Issues
- Resolving Issues
- Pull Requests
- Code of Conduct
- NOTE WELL

Following Discussion
Established Practices

Organizations
Ownership
Repository Management
Issues
Continuous Integration
Working Group == Organization
Ownership and Permissions

Chairs (or responsible AD) own the organization
Chairs (and responsible AD) have admin privileges
Each separate work item has a repository
Editors are added to teams
Teams are given commit access to their repository
Repositories

Editors are responsible for maintaining the repository
Chairs should not commit changes, merge PRs, etc...
One draft per repository is easier to manage
...unless drafts are closely related (use judgment)
Use Markdown

It is easier to use

...especially for IETF newcomers

You get more and better contributions

Both variants are good: [kramdown-rfc2629](https://kramdown-sys.org) and [mmark](https://mmark.org)
Issue Management

Labels can be very useful

A design or editorial distinction can help

For multiple docs, labels can help with sorting

Issues can be assigned; review can be requested for PRs

Milestones and the project board might help
Continuous Integration

Draft about consolidation of Internet services and the impacts it may have on architectures etc.

13 commits
1 branch
0 releases
3 contributors

This is GREAT
Continuous Integration

...but reading this is awful
Continuous Integration

CI systems are designed to check your code

Tools exist to both check that the draft is valid

... and maintain a readable copy of the latest
Example README.md

QUIC Protocol

This is the working area for the IETF QUIC Working Group documents for the QUIC protocol suite.

Core Transport Protocol

- Editor's copy
- Working Group Draft
- Compare Working Group Draft and Editor's copy

Loss Detection & Congestion Control

- Editor's copy
- Working Group Draft
- Compare Working Group Draft and Editor's copy

TLS Mapping
Version-Independent Properties of QUIC

draft-ietf-quic-invariants-latest

Abstract
This document defines the properties of the QUIC transport protocol that are expected to remain unchanged over time as new versions of the protocol are developed.

Note to Readers
Discussion of this draft takes place on the QUIC working group mailing list
Previews of Branches

Editor's drafts for master branch of quicwg/base-drafts

View saved issues, or the latest GitHub issues issues and pull requests.

draft-ietf-quic-http     html  plain text  diff with last submission
draft-ietf-quic-invariants html  plain text  diff with last submission
draft-ietf-quic-qpack    html  plain text  diff with last submission
draft-ietf-quic-recovery html  plain text  diff with last submission
draft-ietf-quic-spin-exp html  plain text  diff with last submission
draft-ietf-quic-tls      html  plain text  diff with last submission
draft-ietf-quic-transport html plain text  diff with last submission

Preview for branch stream_types

draft-ietf-quic-http     html  plain text  diff with master  diff with last submission
draft-ietf-quic-invariants html  plain text  diff with master  diff with last submission
draft-ietf-quic-qpack    html  plain text  diff with master  diff with last submission
draft-ietf-quic-recovery html  plain text  diff with master  diff with last submission
draft-ietf-quic-spin-exp html  plain text  diff with master  diff with last submission
draft-ietf-quic-tls      html  plain text  diff with master  diff with last submission
draft-ietf-quic-transport html plain text  diff with master  diff with last submission
Toolchain

https://github.com/martinthomson/i-d-template

Build changes

CI integration for validating changes and previews

Automated submission to datatracker

Also produces diffs and manages back for issue status
Contributing

Time permitting, we will “fix” an issue in a draft
More Information

https://unicorn-wg.github.io/github-bcp/

Or discuss: wgchairs@ietf.org