Requirements Traceability for RFCs

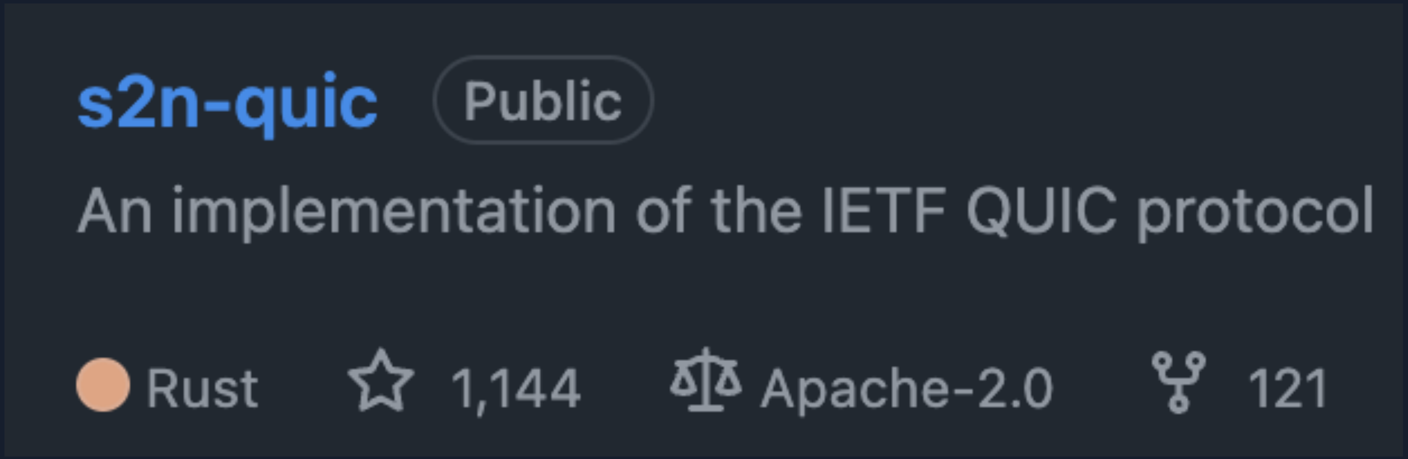
Cameron Bytheway

Principal Engineer
AWS Cryptography

Goals

- What is traceability?
- How can we apply traceability to CA/B Forum efforts?

A story

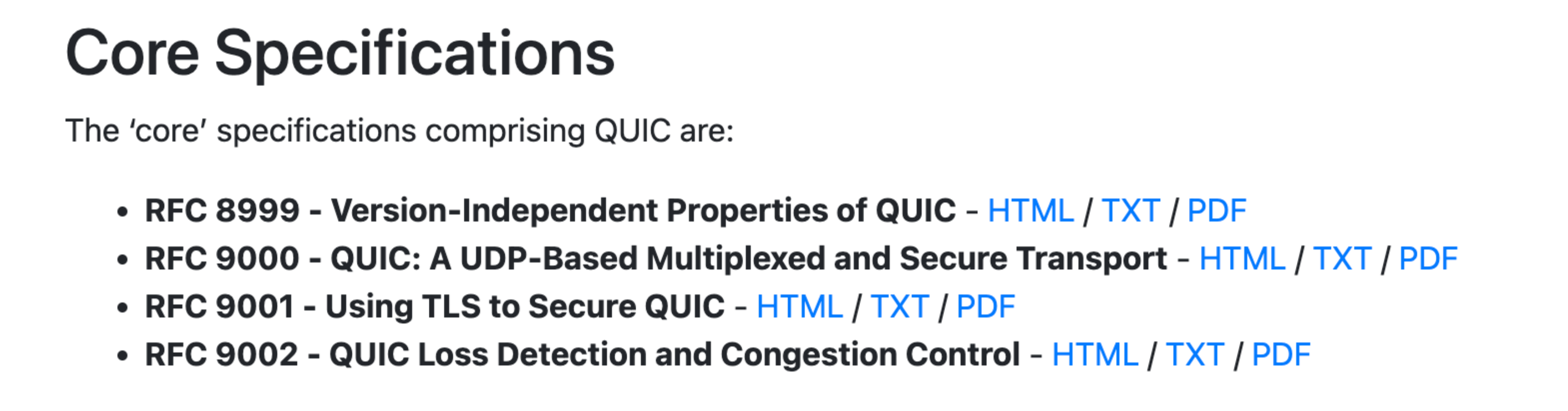


s2n-quic Public

An implementation of the IETF QUIC protocol

● Rust ☆ 1,144 ⚖ Apache-2.0 ⑂ 121

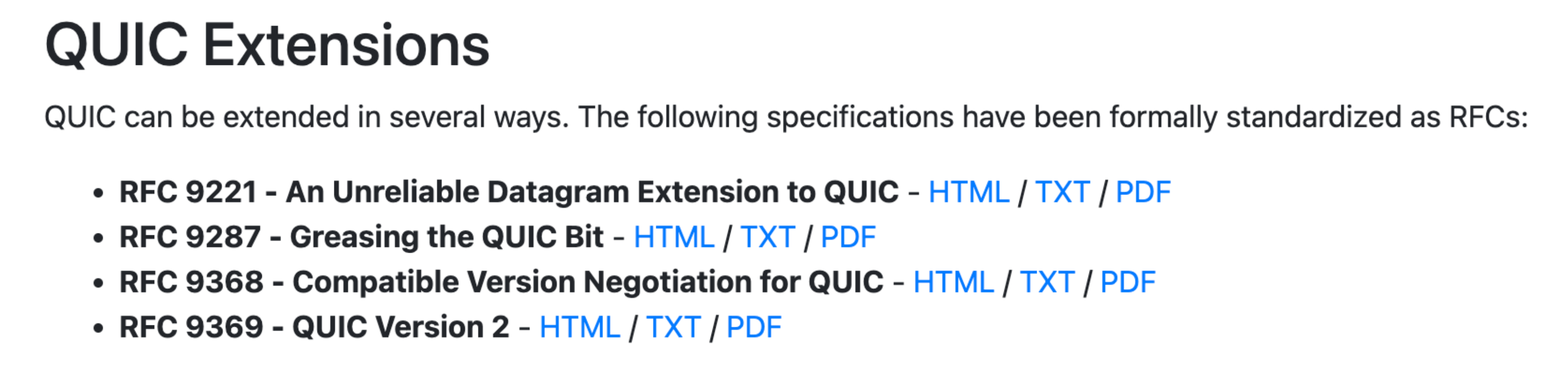
QUIC RFC



Core Specifications

The ‘core’ specifications comprising QUIC are:

- **RFC 8999 - Version-Independent Properties of QUIC** - HTML / TXT / PDF
- **RFC 9000 - QUIC: A UDP-Based Multiplexed and Secure Transport** - HTML / TXT / PDF
- **RFC 9001 - Using TLS to Secure QUIC** - HTML / TXT / PDF
- **RFC 9002 - QUIC Loss Detection and Congestion Control** - HTML / TXT / PDF

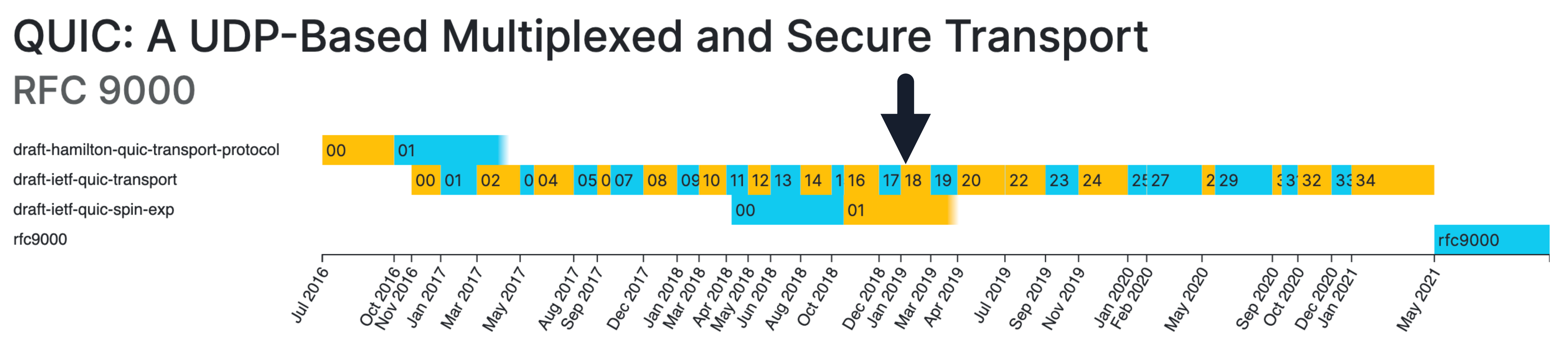


QUIC Extensions

QUIC can be extended in several ways. The following specifications have been formally standardized as RFCs:

- **RFC 9221 - An Unreliable Datagram Extension to QUIC** - HTML / TXT / PDF
- **RFC 9287 - Greasing the QUIC Bit** - HTML / TXT / PDF
- **RFC 9368 - Compatible Version Negotiation for QUIC** - HTML / TXT / PDF
- **RFC 9369 - QUIC Version 2** - HTML / TXT / PDF

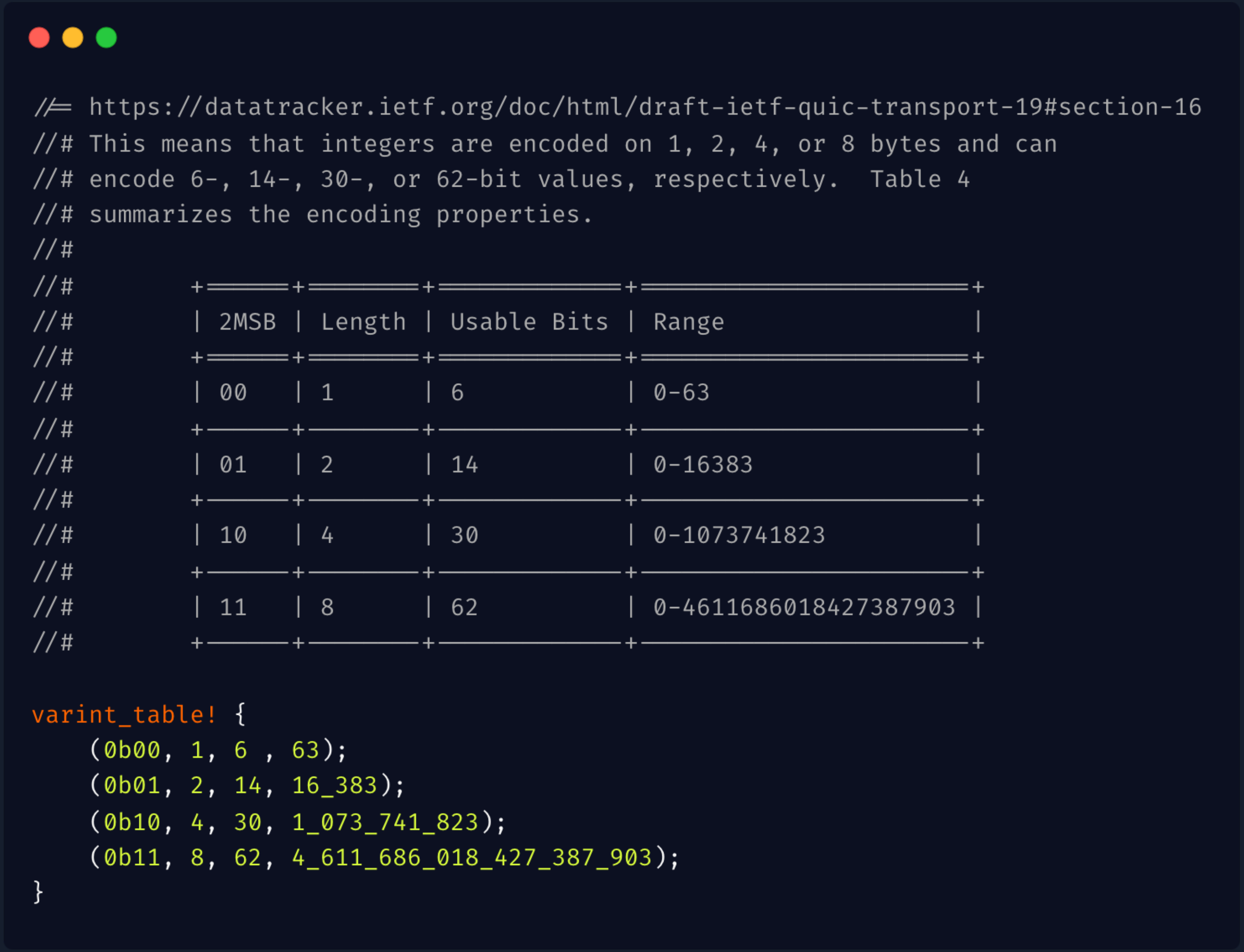
QUIC RFC



QUIC: A UDP-Based Multiplexed and Secure Transport

RFC 9000

RFC Quotes

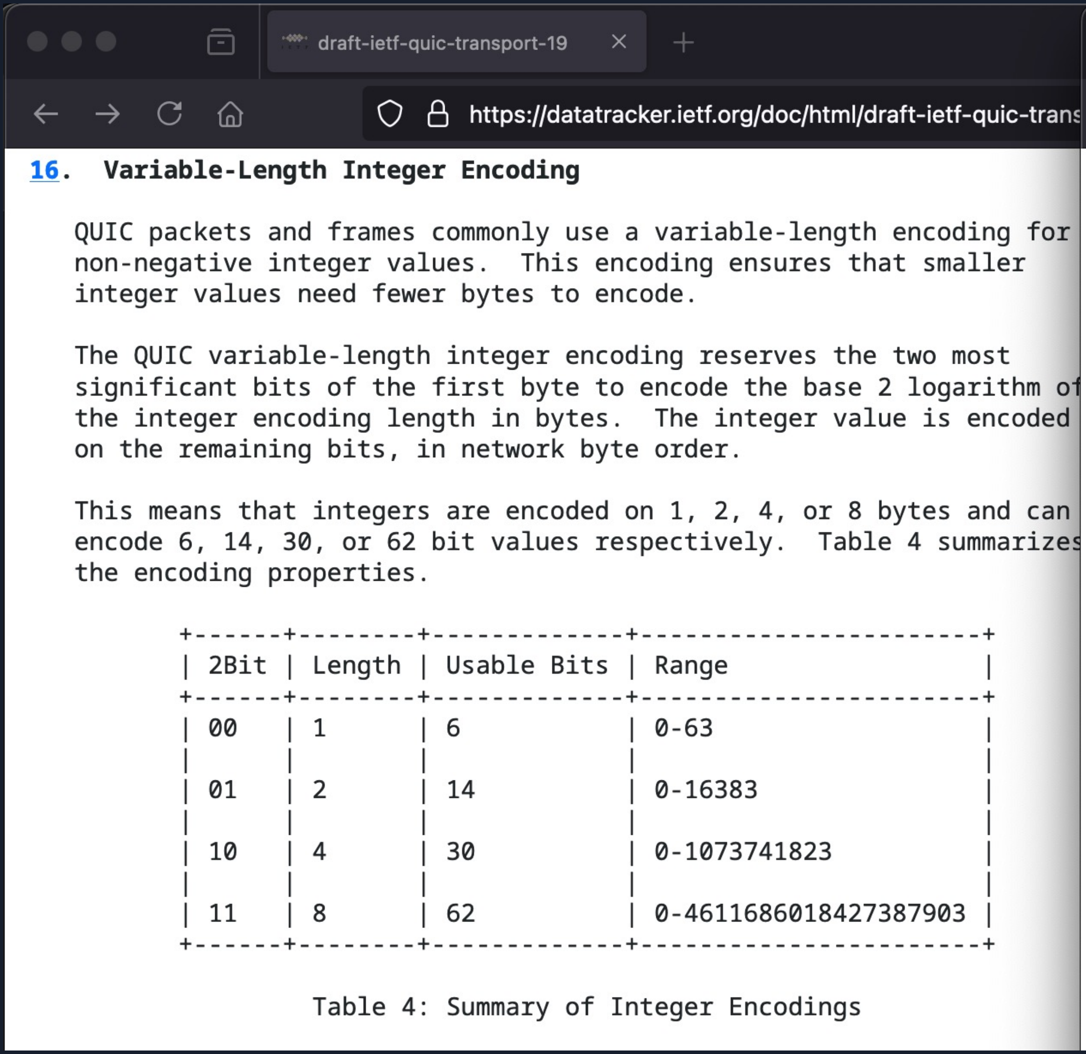


```
//= https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-19#section-16
//# This means that integers are encoded on 1, 2, 4, or 8 bytes and can
//# encode 6-, 14-, 30-, or 62-bit values, respectively.  Table 4
//# summarizes the encoding properties.
//#
//#        +======+========+=============+=======================+
//#        | 2MSB | Length | Usable Bits | Range                 |
//#        +======+========+=============+=======================+
//#        | 00   | 1      | 6           | 0-63                  |
//#        +------+--------+-------------+-----------------------+
//#        | 01   | 2      | 14          | 0-16383               |
//#        +------+--------+-------------+-----------------------+
//#        | 10   | 4      | 30          | 0-1073741823          |
//#        +------+--------+-------------+-----------------------+
//#        | 11   | 8      | 62          | 0-4611686018427387903 |
//#        +------+--------+-------------+-----------------------+

varint_table! {
    (0b00, 1, 6 , 63);
    (0b01, 2, 14, 16_383);
    (0b10, 4, 30, 1_073_741_823);
    (0b11, 8, 62, 4_611_686_018_427_387_903);
}
```

What do we get from this?

- Implementing the code is clearer
- Give context to pull request reviewers
- Leave a paper trail for posterity

What happens when there’s a new draft?


draft-ietf-quic-transport-19
https://datatracker.ietf.org/doc/html/draft-ietf-quic-trans

16. Variable-Length Integer Encoding

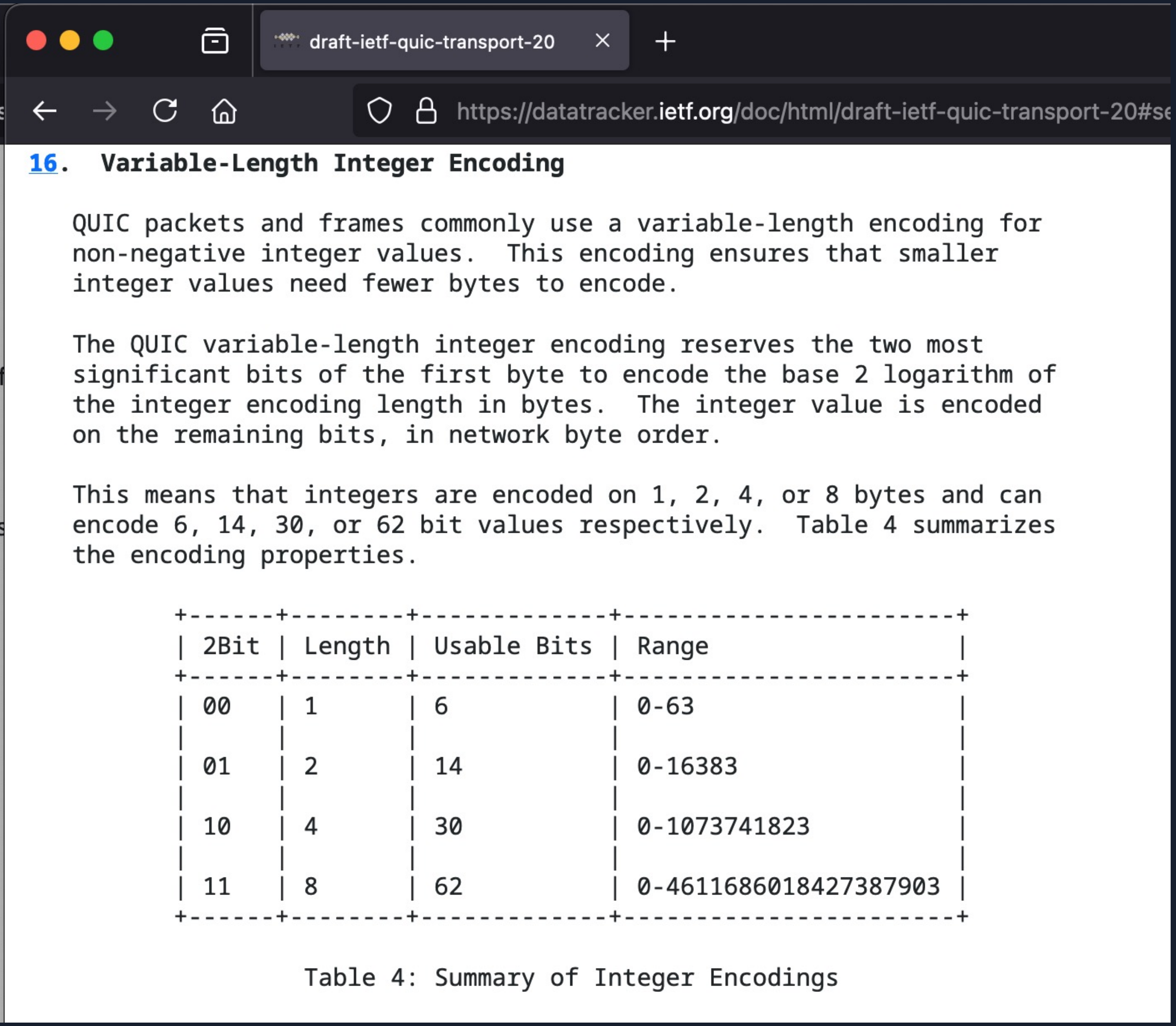
QUIC packets and frames commonly use a variable-length encoding for non-negative integer values. This encoding ensures that smaller integer values need fewer bytes to encode.

The QUIC variable-length integer encoding reserves the two most significant bits of the first byte to encode the base 2 logarithm of the integer encoding length in bytes. The integer value is encoded on the remaining bits, in network byte order.

This means that integers are encoded on 1, 2, 4, or 8 bytes and can encode 6, 14, 30, or 62 bit values respectively. Table 4 summarizes the encoding properties.

2Bit	Length	Usable Bits	Range
00	1	6	0-63
01	2	14	0-16383
10	4	30	0-1073741823
11	8	62	0-4611686018427387903

Table 4: Summary of Integer Encodings


draft-ietf-quic-transport-20
https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-20#se

16. Variable-Length Integer Encoding

QUIC packets and frames commonly use a variable-length encoding for non-negative integer values. This encoding ensures that smaller integer values need fewer bytes to encode.

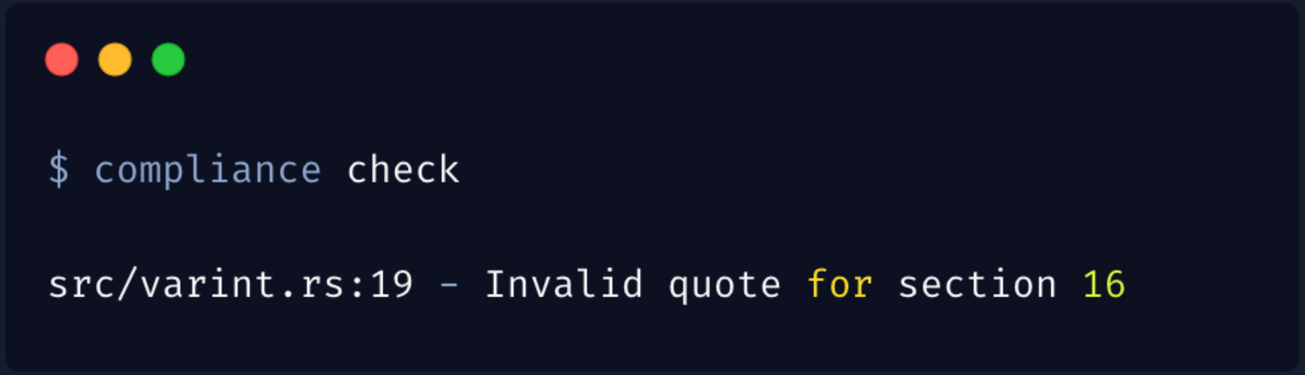
The QUIC variable-length integer encoding reserves the two most significant bits of the first byte to encode the base 2 logarithm of the integer encoding length in bytes. The integer value is encoded on the remaining bits, in network byte order.

This means that integers are encoded on 1, 2, 4, or 8 bytes and can encode 6, 14, 30, or 62 bit values respectively. Table 4 summarizes the encoding properties.

2Bit	Length	Usable Bits	Range
00	1	6	0-63
01	2	14	0-16383
10	4	30	0-1073741823
11	8	62	0-4611686018427387903

Table 4: Summary of Integer Encodings

In need of a tool

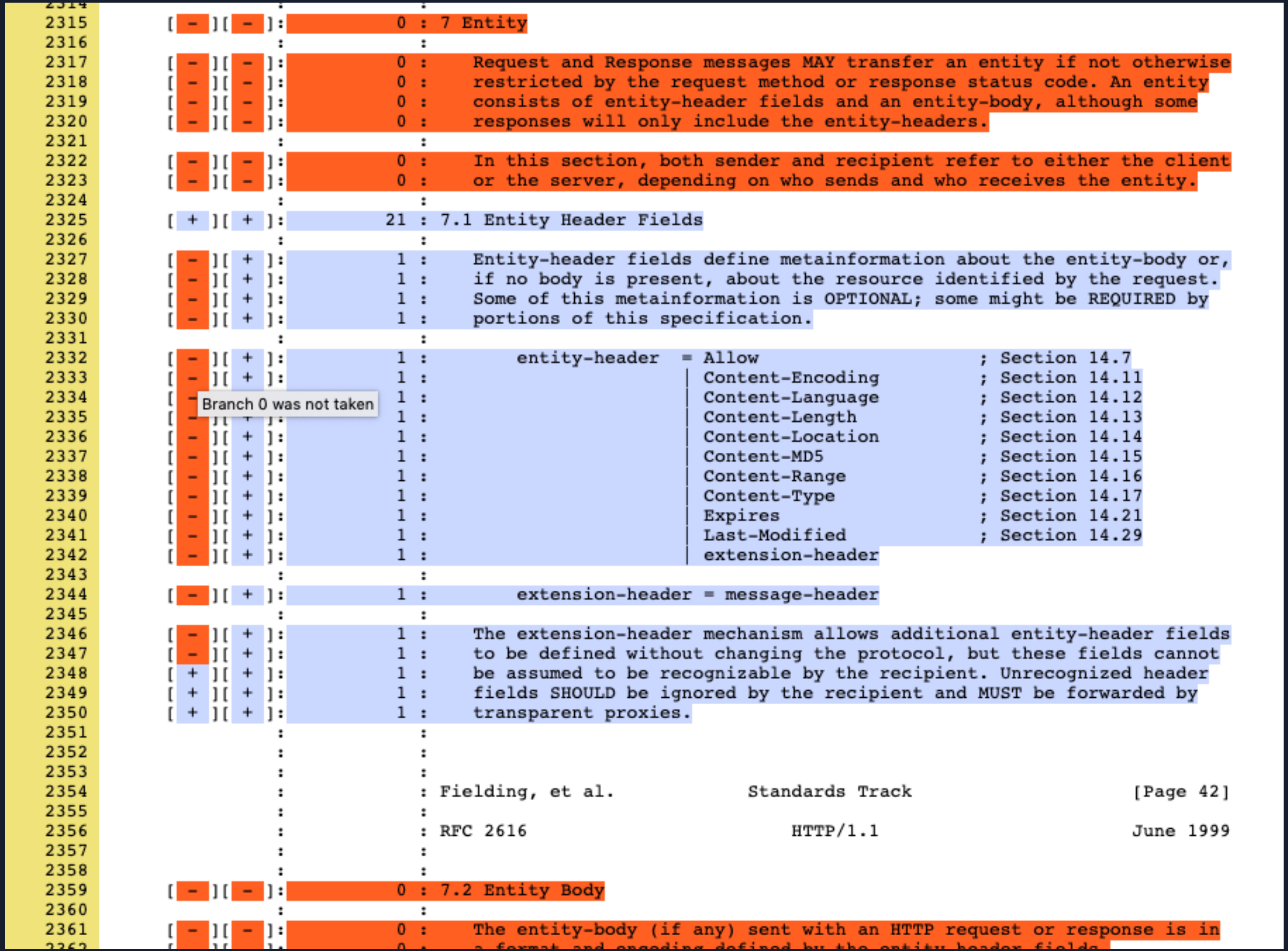


```
$ compliance check

src/varint.rs:19 - Invalid quote for section 16
```

What do we get from this?

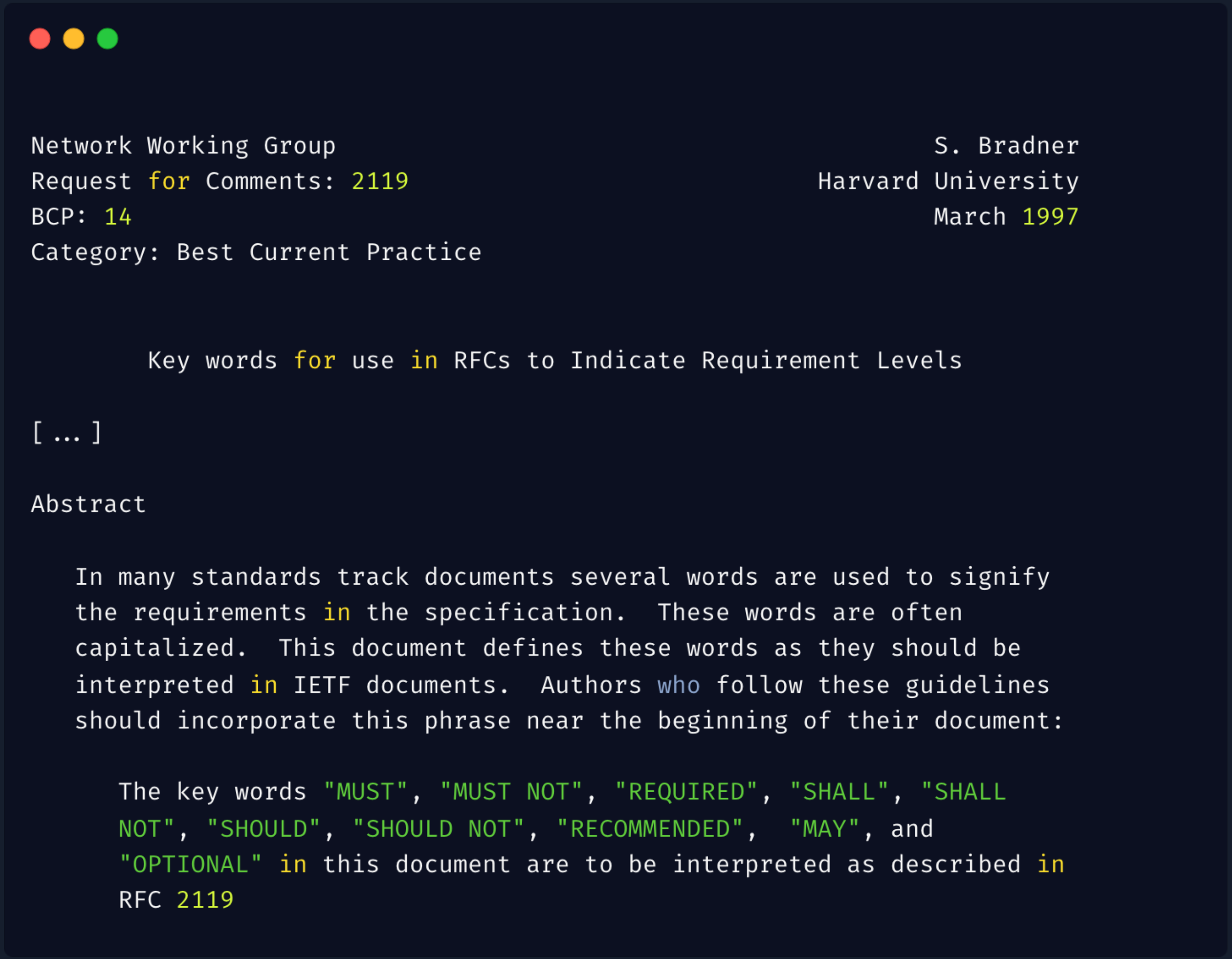
- Greatly reduced the time manually checking for changes
- Reduced the amount of human error
- Some enforcement in CI

How do we track progress/coverage?



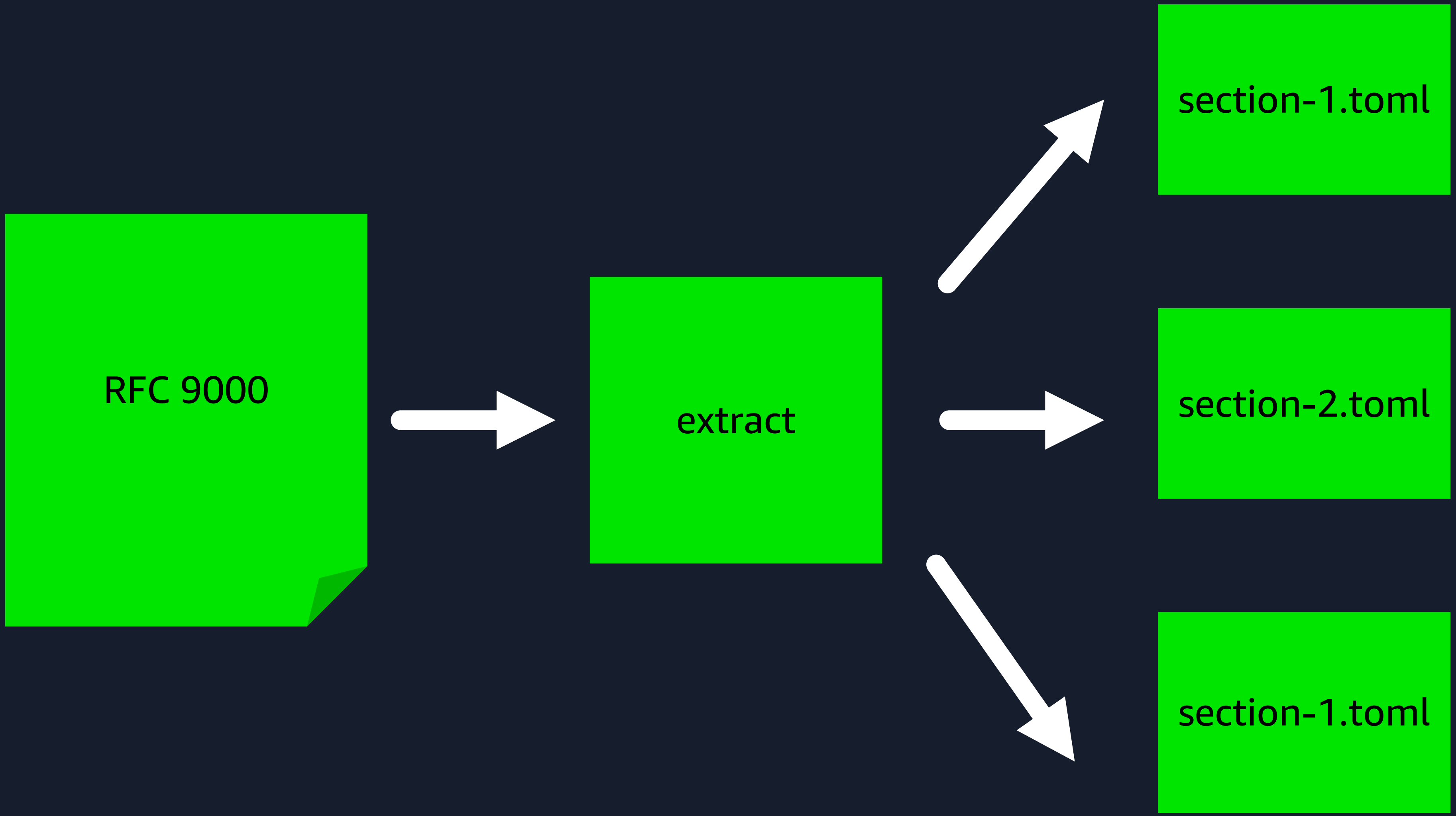
```
                 :      :
[ - ][ - ]:      0 : 7 Entity
          :        :
[ - ][ - ]:      0 :    Request and Response messages MAY transfer an entity if not otherwise
[ - ][ - ]:      0 :    restricted by the request method or response status code. An entity
[ - ][ - ]:      0 :    consists of entity-header fields and an entity-body, although some
[ - ][ - ]:      0 :    responses will only include the entity-headers.
          :        :
[ - ][ - ]:      0 :    In this section, both sender and recipient refer to either the client
[ - ][ - ]:      0 :    or the server, depending on who sends and who receives the entity.
          :        :
[ + ][ + ]:     21 : 7.1 Entity Header Fields
          :        :
[ - ][ + ]:      1 :    Entity-header fields define metainformation about the entity-body or,
[ - ][ + ]:      1 :    if no body is present, about the resource identified by the request.
[ - ][ + ]:      1 :    Some of this metainformation is OPTIONAL; some might be REQUIRED by
[ - ][ + ]:      1 :    portions of this specification.
          :        :
[ - ][ + ]:      1 :        entity-header  = Allow                    ; Section 14.7
[ - ][ + ]:      1 :                       | Content-Encoding         ; Section 14.11
[ -  Branch 0 was not taken  1 :          | Content-Language         ; Section 14.12
[ -       ]:      1 :                       | Content-Length           ; Section 14.13
[ - ][ + ]:      1 :                       | Content-Location         ; Section 14.14
[ - ][ + ]:      1 :                       | Content-MD5              ; Section 14.15
[ - ][ + ]:      1 :                       | Content-Range            ; Section 14.16
[ - ][ + ]:      1 :                       | Content-Type             ; Section 14.17
[ - ][ + ]:      1 :                       | Expires                  ; Section 14.21
[ - ][ + ]:      1 :                       | Last-Modified            ; Section 14.29
[ - ][ + ]:      1 :                       | extension-header
          :        :
[ - ][ + ]:      1 :        extension-header = message-header
          :        :
[ - ][ + ]:      1 :    The extension-header mechanism allows additional entity-header fields
[ - ][ + ]:      1 :    to be defined without changing the protocol, but these fields cannot
[ + ][ + ]:      1 :    be assumed to be recognizable by the recipient. Unrecognized header
[ + ][ + ]:      1 :    fields SHOULD be ignored by the recipient and MUST be forwarded by
[ + ][ + ]:      1 :    transparent proxies.
          :        :
          :        :
          :        :
          :        : Fielding, et al.            Standards Track                  [Page 42]
          :        :
          :        : RFC 2616                       HTTP/1.1                       June 1999
          :        :
          :        :
[ - ][ - ]:      0 : 7.2 Entity Body
          :        :
[ - ][ - ]:      0 :    The entity-body (if any) sent with an HTTP request or response is in
[ - ][ - ]:      0 :    a format and encoding defined by the entity-header fields.
```

RFC 2119



```
Network Working Group                                         S. Bradner
Request for Comments: 2119                            Harvard University
BCP: 14                                                       March 1997
Category: Best Current Practice


        Key words for use in RFCs to Indicate Requirement Levels

[ … ]

Abstract

   In many standards track documents several words are used to signify
   the requirements in the specification.  These words are often
   capitalized.  This document defines these words as they should be
   interpreted in IETF documents.  Authors who follow these guidelines
   should incorporate this phrase near the beginning of their document:

      The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL
      NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED",  "MAY", and
      "OPTIONAL" in this document are to be interpreted as described in
      RFC 2119
```

Extracting requirements



RFC 9000

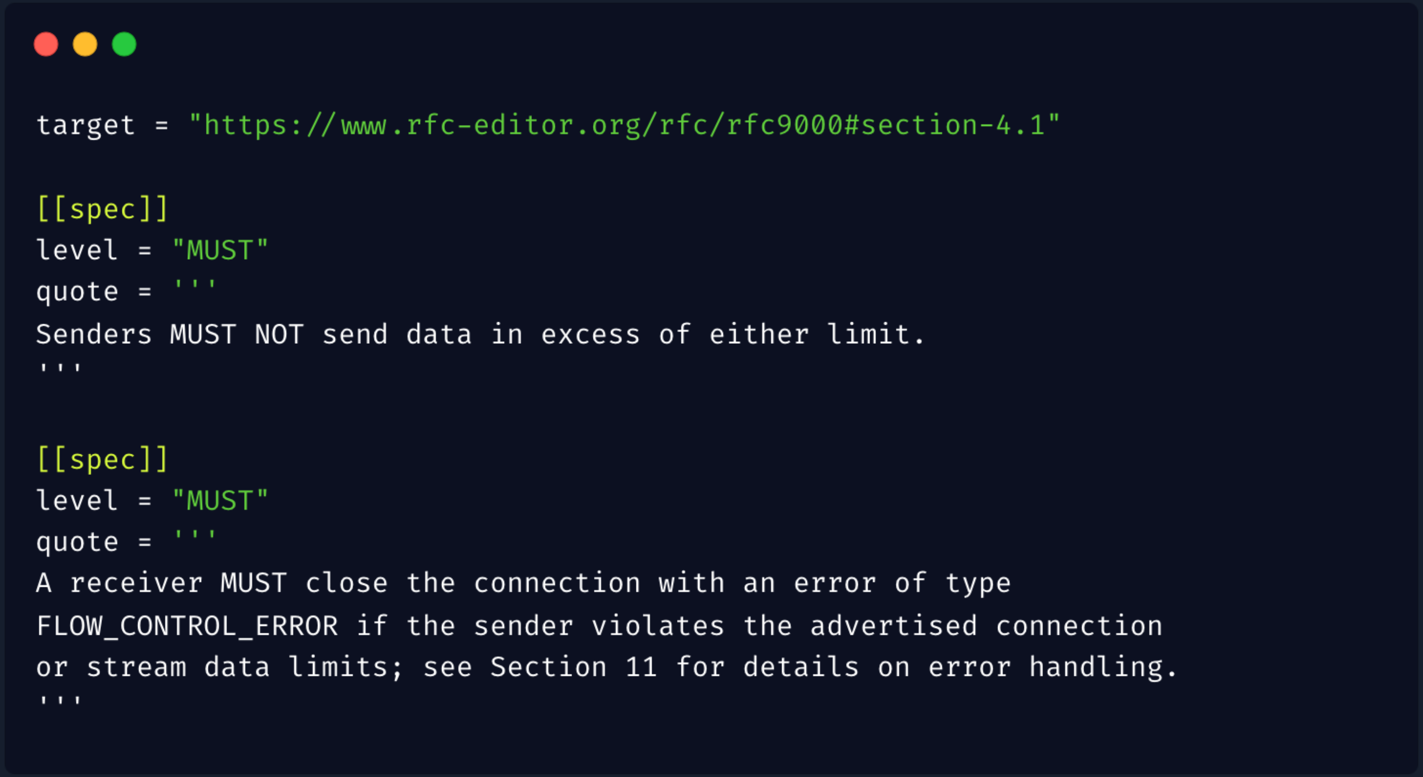
extract

section-1.toml

section-2.toml

section-1.toml

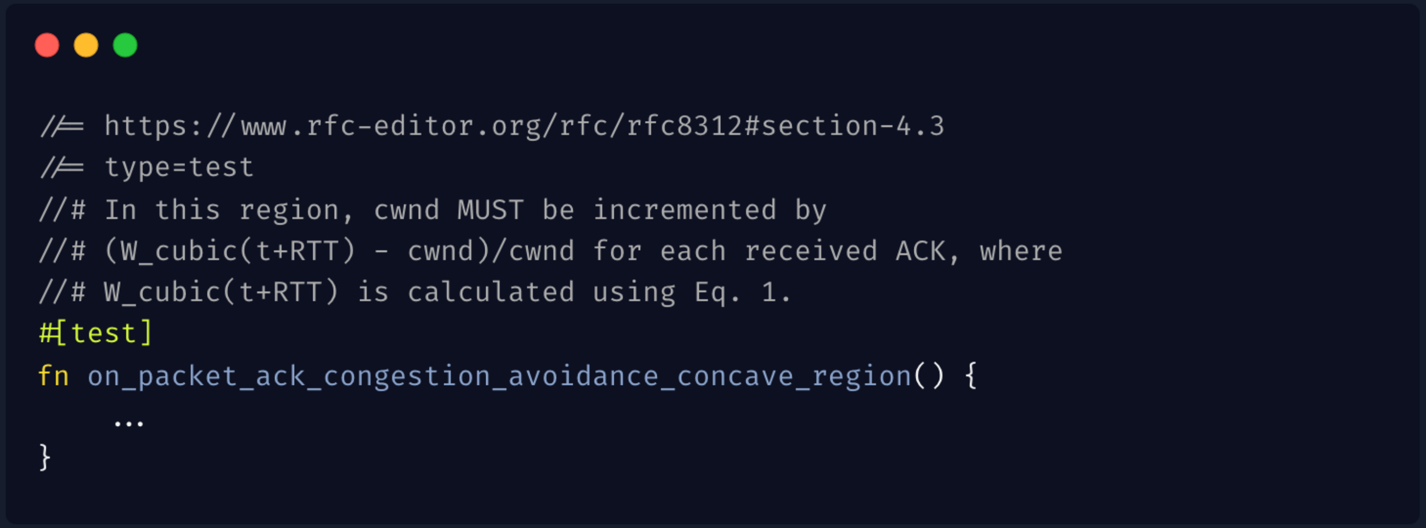
Extracting requirements



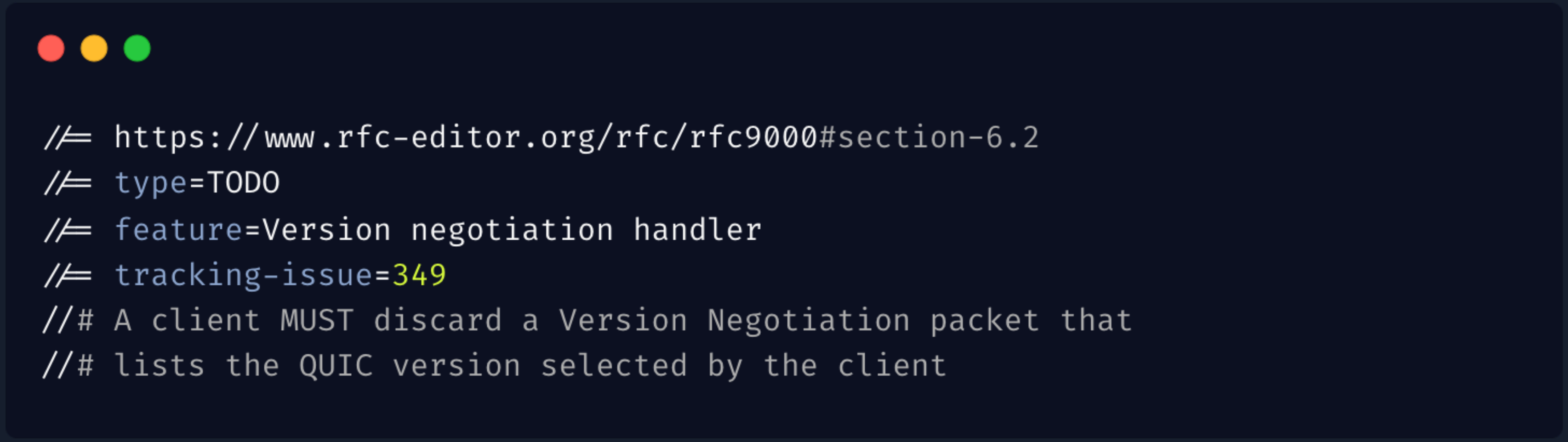
```
target = "https://www.rfc-editor.org/rfc/rfc9000#section-4.1"

[[spec]]
level = "MUST"
quote = '''
Senders MUST NOT send data in excess of either limit.
'''

[[spec]]
level = "MUST"
quote = '''
A receiver MUST close the connection with an error of type
FLOW_CONTROL_ERROR if the sender violates the advertised connection
or stream data limits; see Section 11 for details on error handling.
'''
```

Not all references are equal

- Implementation
- Test
- TODO
- Exception
- Implication



```
//= https://www.rfc-editor.org/rfc/rfc8312#section-4.3
//= type=test
//# In this region, cwnd MUST be incremented by
//# (W_cubic(t+RTT) - cwnd)/cwnd for each received ACK, where
//# W_cubic(t+RTT) is calculated using Eq. 1.
#[test]
fn on_packet_ack_congestion_avoidance_concave_region() {
    ...
}
```

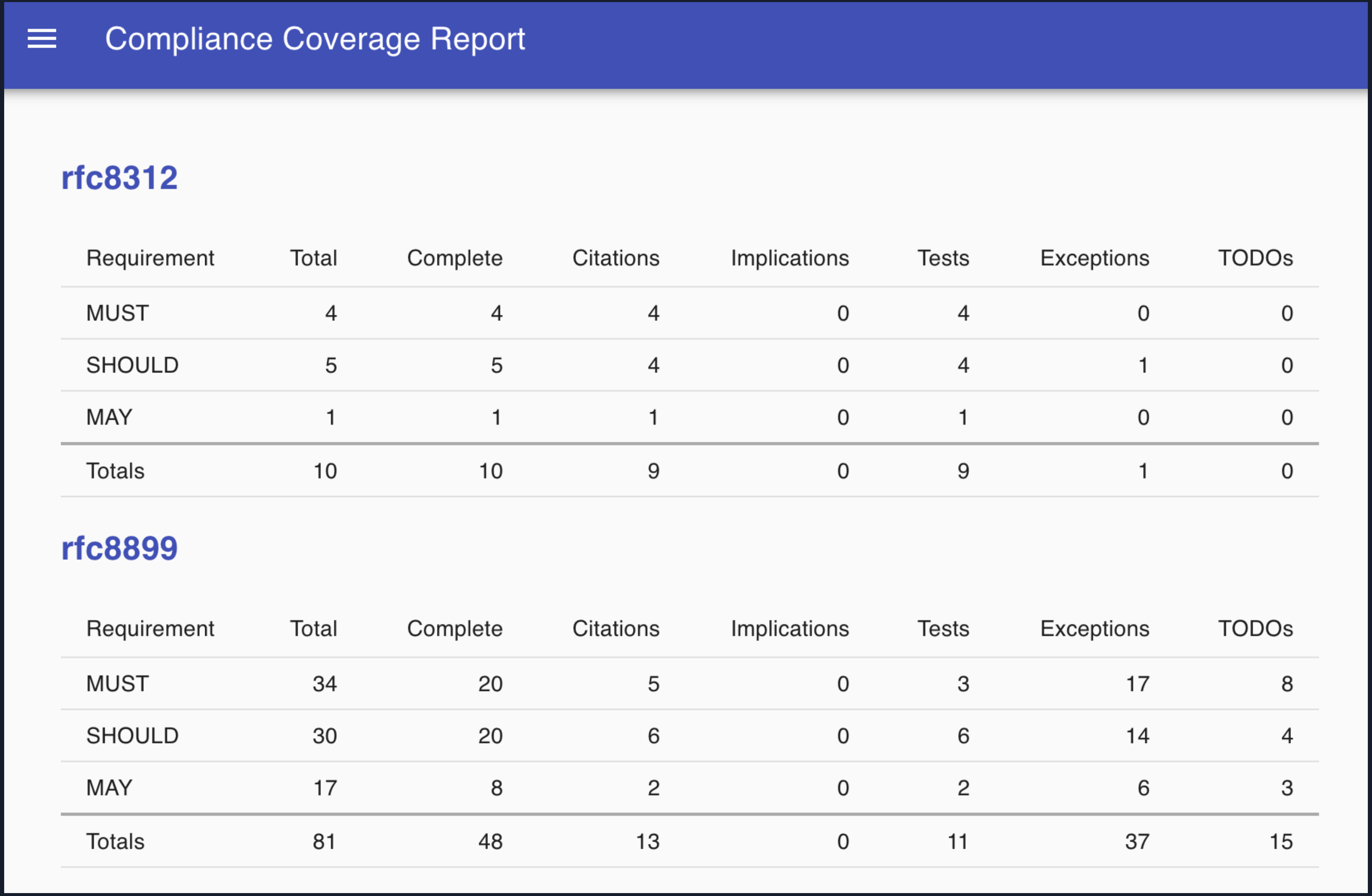


```
//= https://www.rfc-editor.org/rfc/rfc9000#section-6.2
//= type=TODO
//= feature=Version negotiation handler
//= tracking-issue=349
//# A client MUST discard a Version Negotiation packet that
//# lists the QUIC version selected by the client
```

What do we get from this?

- Automatic extraction of requirements
- Assigned priorities to each requirement
- References better reflect current status
- Percent completion based on requirements, not text

A better report



Compliance Coverage Report

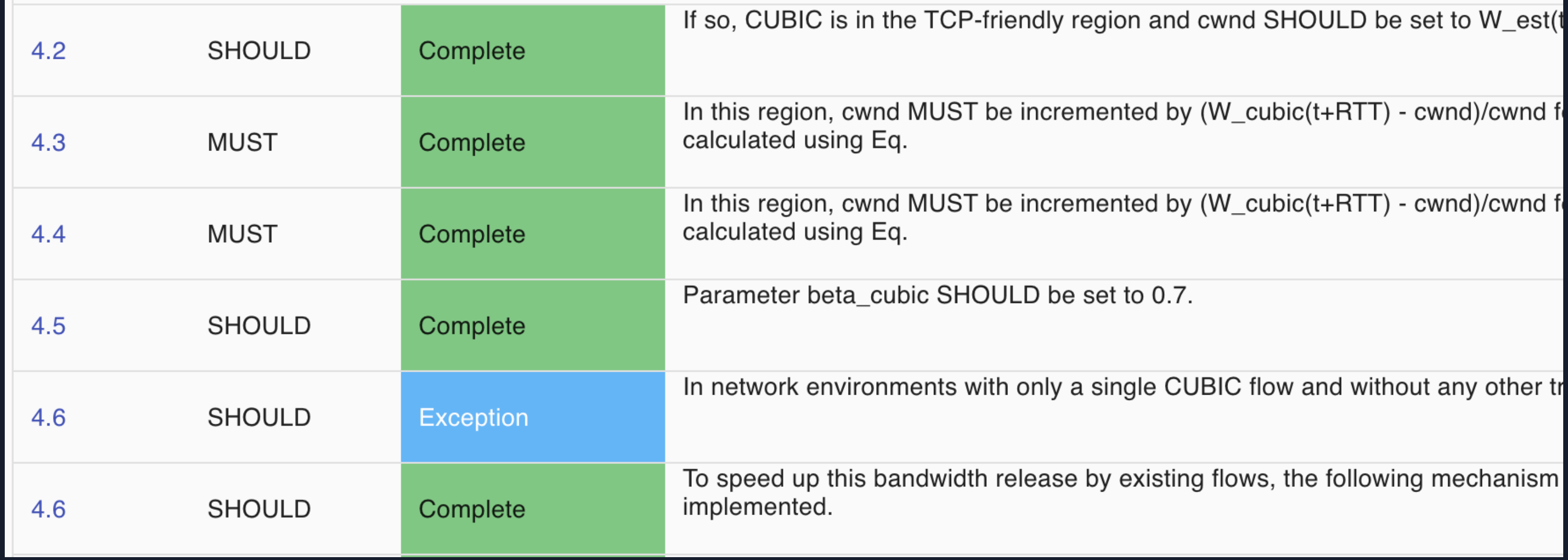
rfc8312

Requirement	Total	Complete	Citations	Implications	Tests	Exceptions	TODOs
MUST	4	4	4	0	4	0	0
SHOULD	5	5	4	0	4	1	0
MAY	1	1	1	0	1	0	0
Totals	10	10	9	0	9	1	0

rfc8899

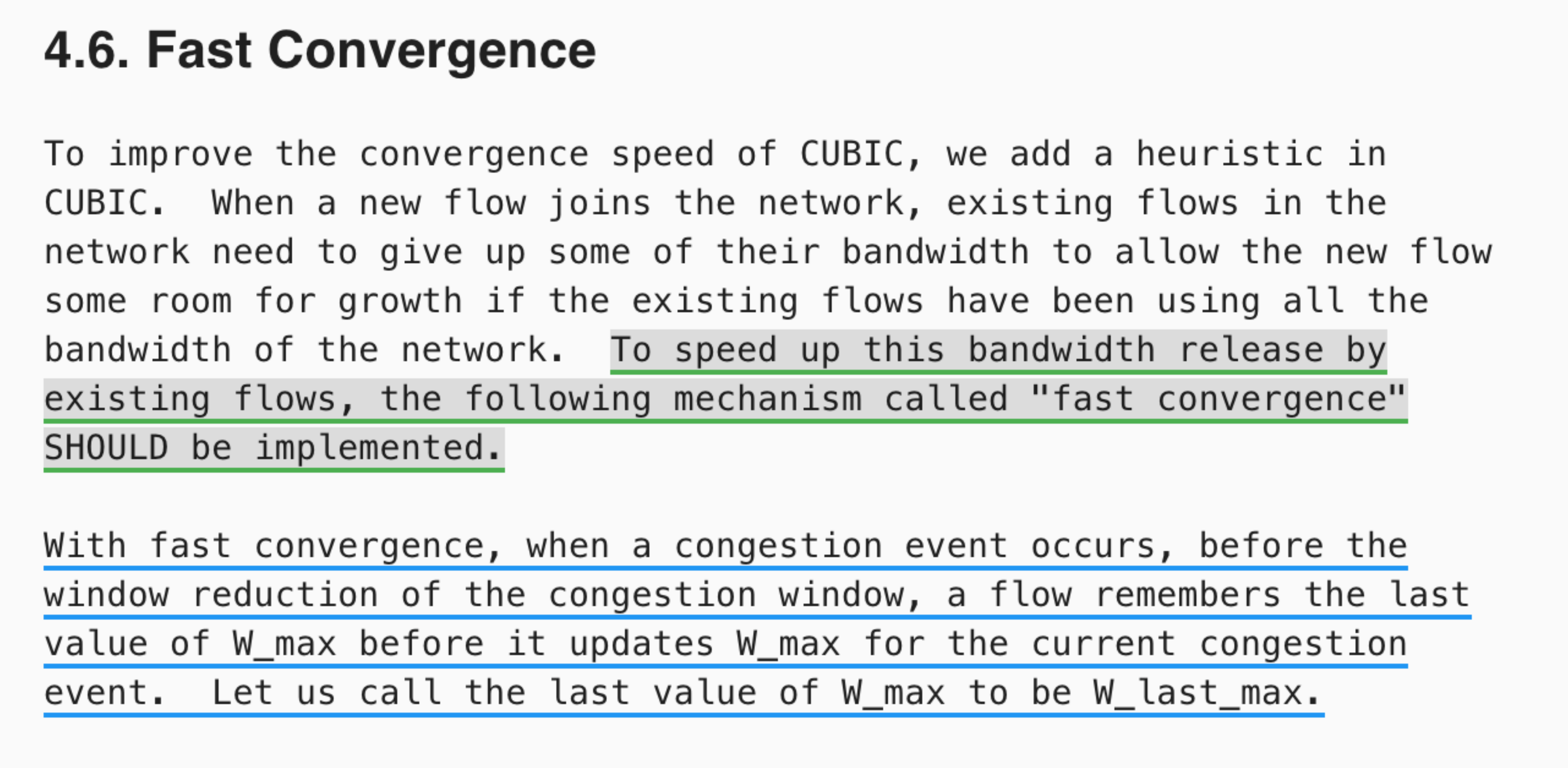
Requirement	Total	Complete	Citations	Implications	Tests	Exceptions	TODOs
MUST	34	20	5	0	3	17	8
SHOULD	30	20	6	0	6	14	4
MAY	17	8	2	0	2	6	3
Totals	81	48	13	0	11	37	15

A better report



4.2	SHOULD	Complete	If so, CUBIC is in the TCP-friendly region and cwnd SHOULD be set to W_est(t
4.3	MUST	Complete	In this region, cwnd MUST be incremented by (W_cubic(t+RTT) - cwnd)/cwnd f calculated using Eq.
4.4	MUST	Complete	In this region, cwnd MUST be incremented by (W_cubic(t+RTT) - cwnd)/cwnd f calculated using Eq.
4.5	SHOULD	Complete	Parameter beta_cubic SHOULD be set to 0.7.
4.6	SHOULD	Exception	In network environments with only a single CUBIC flow and without any other tr
4.6	SHOULD	Complete	To speed up this bandwidth release by existing flows, the following mechanism implemented.

A better report

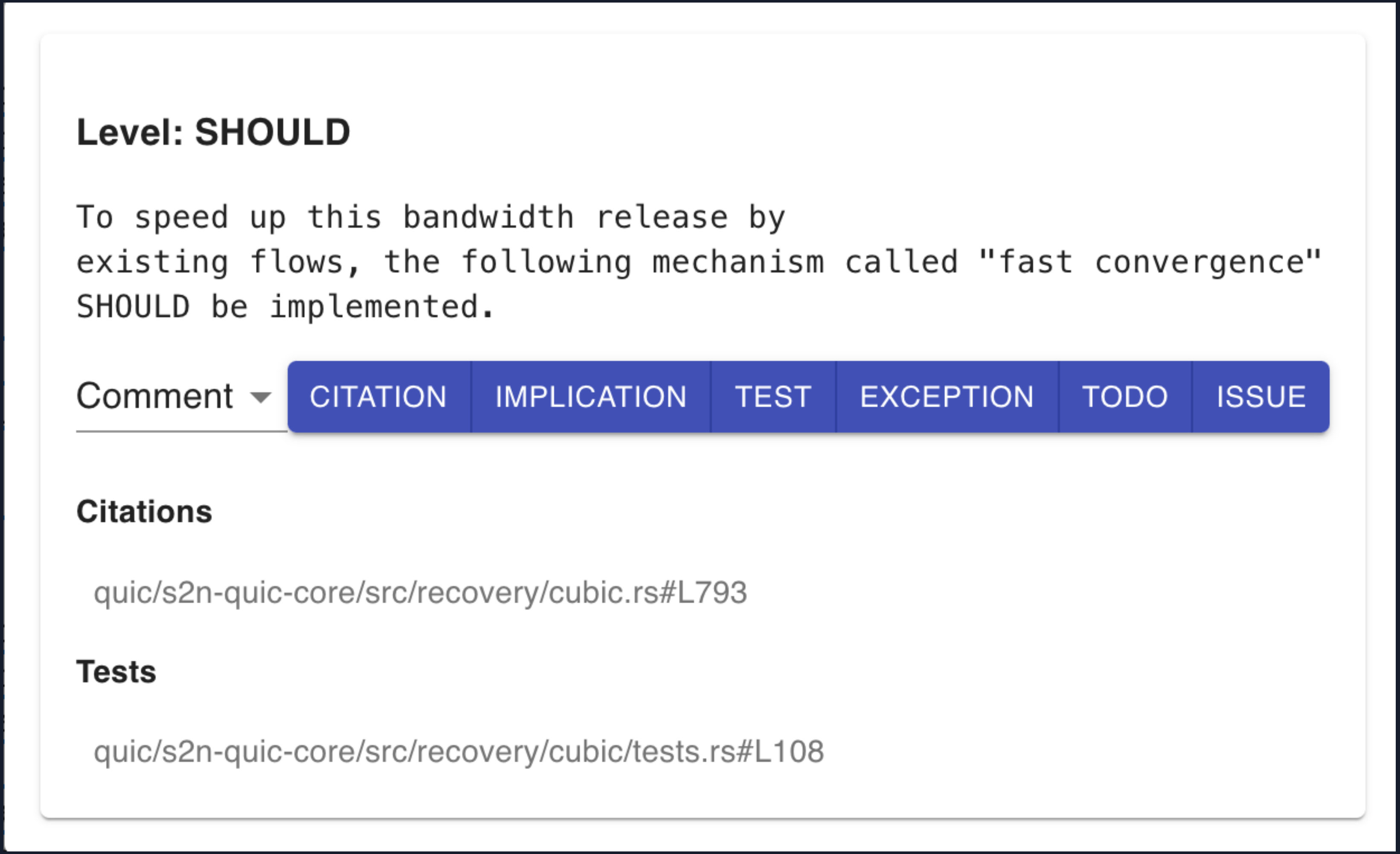


4.6. Fast Convergence

To improve the convergence speed of CUBIC, we add a heuristic in CUBIC. When a new flow joins the network, existing flows in the network need to give up some of their bandwidth to allow the new flow some room for growth if the existing flows have been using all the bandwidth of the network. To speed up this bandwidth release by existing flows, the following mechanism called "fast convergence" SHOULD be implemented.

With fast convergence, when a congestion event occurs, before the window reduction of the congestion window, a flow remembers the last value of W_max before it updates W_max for the current congestion event. Let us call the last value of W_max to be W_last_max.

A better report



Level: SHOULD

```
To speed up this bandwidth release by
existing flows, the following mechanism called "fast convergence"
SHOULD be implemented.
```

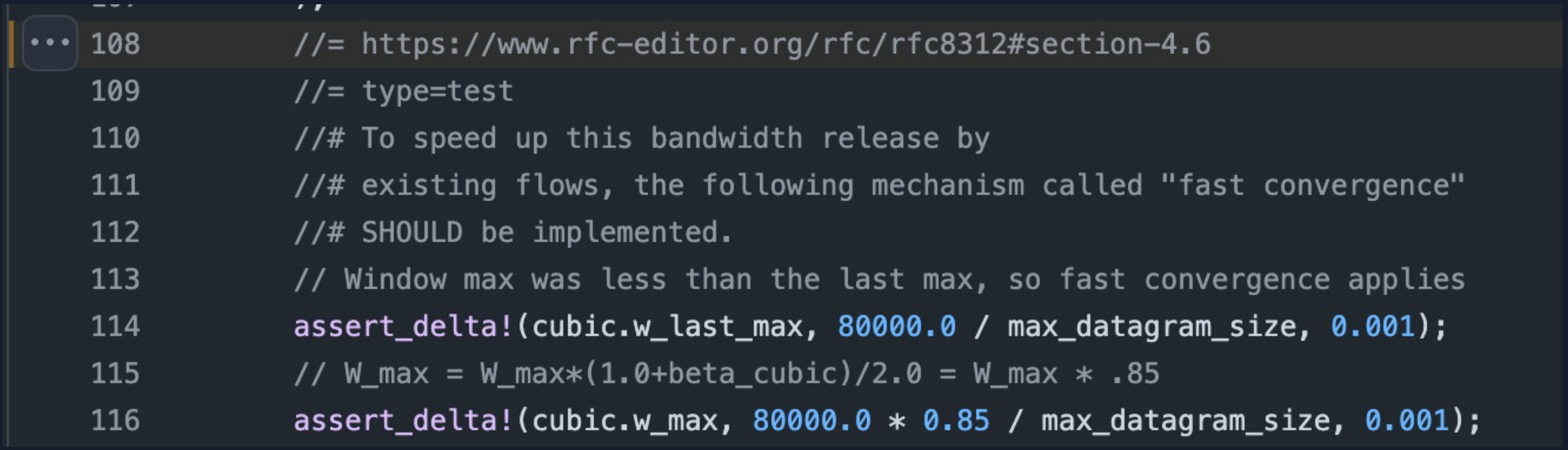
Citations

quic/s2n-quic-core/src/recovery/cubic.rs#L793

Tests

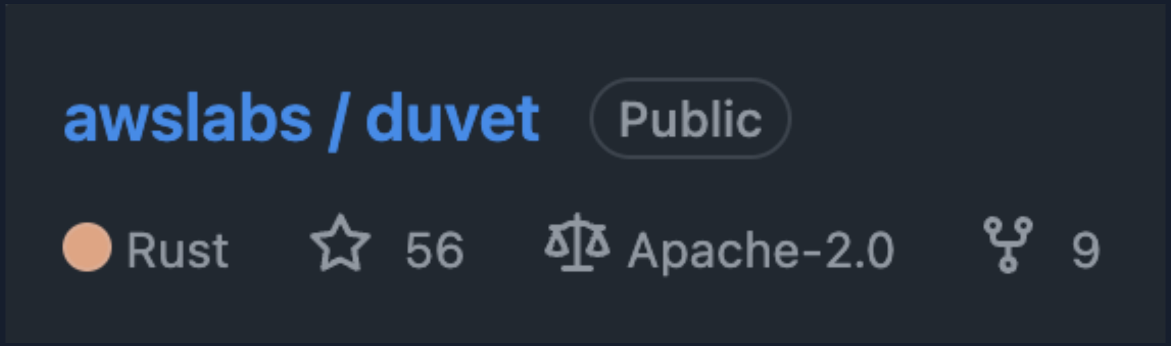
quic/s2n-quic-core/src/recovery/cubic/tests.rs#L108

A better report



```
//= https://www.rfc-editor.org/rfc/rfc8312#section-4.6
//= type=test
//# To speed up this bandwidth release by
//# existing flows, the following mechanism called "fast convergence"
//# SHOULD be implemented.
// Window max was less than the last max, so fast convergence applies
assert_delta!(cubic.w_last_max, 80000.0 / max_datagram_size, 0.001);
// W_max = W_max*(1.0+beta_cubic)/2.0 = W_max * .85
assert_delta!(cubic.w_max, 80000.0 * 0.85 / max_datagram_size, 0.001);
```

Duvet



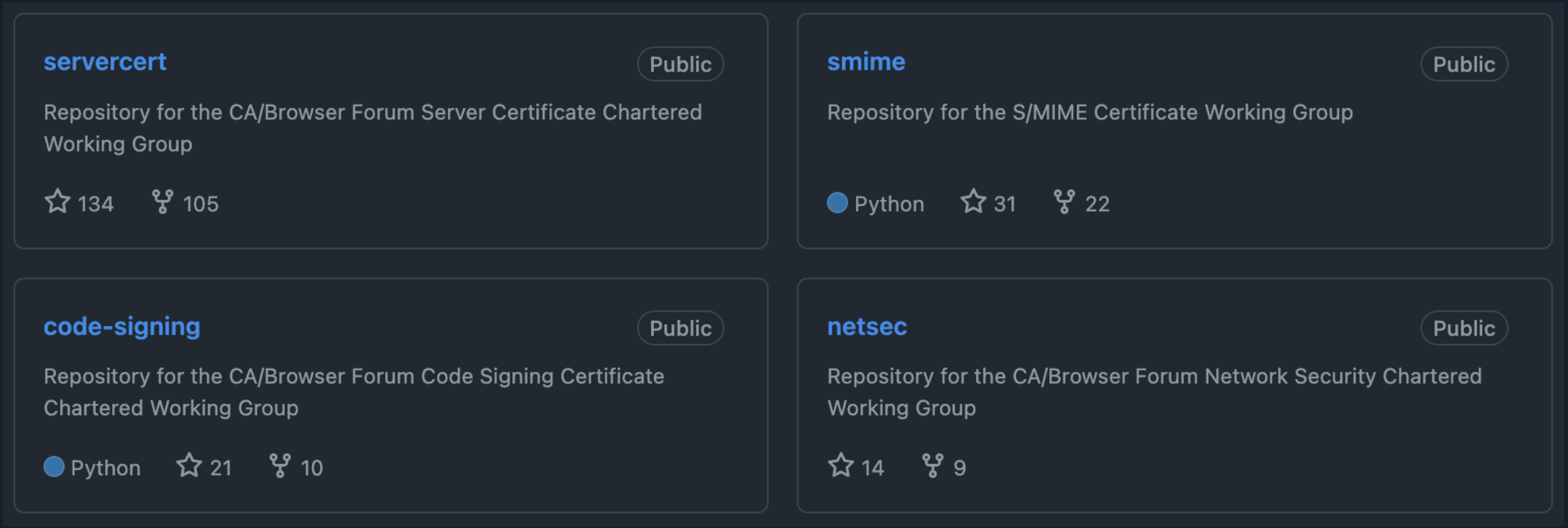
awslabs / duvet Public

● Rust ☆ 56 Apache-2.0 9

What is Traceability?

"the ability to describe and follow the life of a requirement in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases)"

So what?



servercert Public

Repository for the CA/Browser Forum Server Certificate Chartered Working Group

☆ 134 ⑂ 105

smime Public

Repository for the S/MIME Certificate Working Group

● Python ☆ 31 ⑂ 22

code-signing Public

Repository for the CA/Browser Forum Code Signing Certificate Chartered Working Group

● Python ☆ 21 ⑂ 10

netsec Public

Repository for the CA/Browser Forum Network Security Chartered Working Group

☆ 14 ⑂ 9

Thank you!