

TCP RST Diagnostic Payload

[draft-boucadair-tcpm-rst-diagnostic-payload](#)

IETF#125, March 2026, Shenzhen

M. Boucadair , T. Reddy, & **J. Xing**

Pending Issues from IETF 124

- More INTERNET experiments are needed before standardization
- CBOR format is not appropriate for TCP

Link: <https://datatracker.ietf.org/doc/minutes-124-tcpm-202511051700/00/>

Current Status

- Replace CBOR with a more concise and compact plain text
- Complete common socket API extension
- Support FreeBSD kernel
- Support Wireshark
- Change Intended status to Experimental

Payload Format (2)

- An alternative/optional format to help users get more useful information

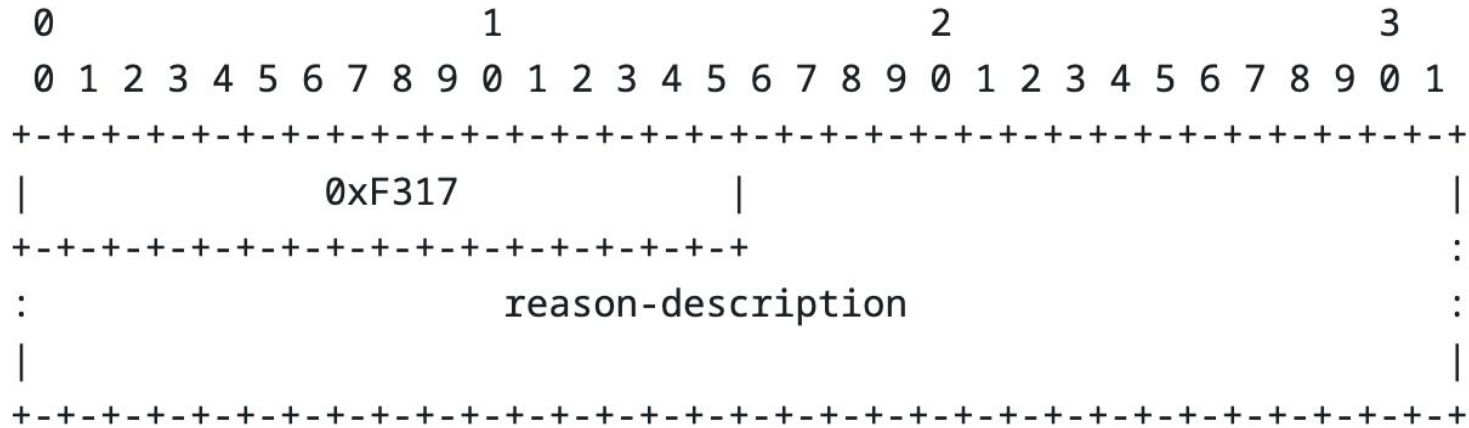


Figure 2: Structure of the RST Diagnostic Payload with Reason Description

Socket API

- Michael Tuexen wrote a clean API documentation

- Detailed discussion:

<https://github.com/boucadair/draft-boucadair-tcpm-rst-diagnostic-payload/pull/14>

Option Name	Data Type	Set	Get
TCP_RST_REASON_ENABLE	uint32_t	X	
TCP_RST_REASON_CODE	struct tcp_rst_reason	X	X
TCP_RST_REASON_DESC	char[]	X	X

Table 1: Socket Options

Kernel Implementations

- Michael Tuexen has done the complete patch targetted at FreeBSD kernel
 - <https://reviews.freebsd.org/D55338>
- Jason Xing implemented a simple function in Linux kernel to do exhaustive experiments in production. Prior to this, part of this draft has been merged in mainline in 2024.
 - <https://mailarchive.ietf.org/arch/msg/tcpm/BISlIldzPWSUuggluCF4o-5QNOVI/>
 - <https://git.kernel.org/pub/scm/linux/kernel/git/netdev/net-next.git/commit/?id=d5115a55ffb52>

Wireshark Support

- Gavin Bunney updates RST diagnostic payload for draft-15
 - commit:
<https://gitlab.com/wireshark/wireshark/-/commit/89359ff540c26eecab2e3457062b541b203030eb>
 - pcap:
<https://gitlab.com/wireshark/wireshark/-/blob/89359ff540c26eecab2e3457062b541b203030eb/test/captures/tcp-rst-diagnostic.pcap>
- Note that tcpdump reserved this feature around 2001

Length	Info
48	12345 → 80 [RST, ACK] Seq=1000 Ack=2000 Win=0 Len=8
48	12346 → 443 [RST, ACK] Seq=1000 Ack=2000 Win=0 Len=8
57	54321 → 8080 [RST, ACK] Seq=1000 Ack=2000 Win=0 Len=17
58	11111 → 22 [RST, ACK] Seq=1000 Ack=2000 Win=0 Len=18
46	22222 → 443 [RST, ACK] Seq=1000 Ack=2000 Win=0 Len=6

```
Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2
  Transmission Control Protocol, Src Port: 22222, Dst Port: 443, Seq: 1000, Ack: 2000, Len: 6
    Source Port: 22222
    Destination Port: 443
    [Stream index: 4]
    [Conversation completeness: Incomplete (40)]
    [TCP Segment Len: 6]
    Sequence Number: 1000
    [Next Sequence Number: 1006]
    Acknowledgment Number: 2000
    0101 ... = Header Length: 20 bytes (5)
  > Flags: 0x014 (RST, ACK)
    Window: 0
    [Calculated window size: 0]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x03c8 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
  > Reset cause: 30
0000  45 00 00 2e 12 34 40 00  40 06 14 88 0a 00 00 07  E...4@.....
0010  0a 00 00 08 56 ce 01 bb  00 00 03 e8 00 00 07 d0  ....V.....
0020  50 14 00 00 03 c8 00 00  33 aa 00 09 00 00      P.....3.....
```

Next Steps

- Request WG Adoption