SCONE
Internet-Draft
Intended status: Standards Track
Expires: 18 June 2026

W. Eddy
M. Joras
Meta

15 December 2025

# SCONE TCP Option
## draft-eddy-tcpm-scone-01

Abstract

This document describes a TCP option that permits network elements to provide TCP endpoints advice on rate limits within the network. The functionality for TCP is analogous to SCONE packets within the QUIC protocol.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 18 June 2026.

Table of Contents



## 1. SCONE Background and Introduction

Standard Communication with Network Elements (SCONE) [I-D.ietf-scone-protocol] is an extension to QUIC [RFC9000] that provides the capability for network elements to provide QUIC application endpoints with guidance on potential permitted throughput, e.g. in order to make explicit the presence of traffic limiting mechanisms within the network that can be problematic for video streaming [I-D.joras-scone-video-optimization-requirements] and other applications.

In QUIC, SCONE is negotiated between endpoints using a transport parameter, and the QUIC endpoints send SCONE packets periodically. The SCONE packets are visible to network elements that modify the throughput guidance within them.

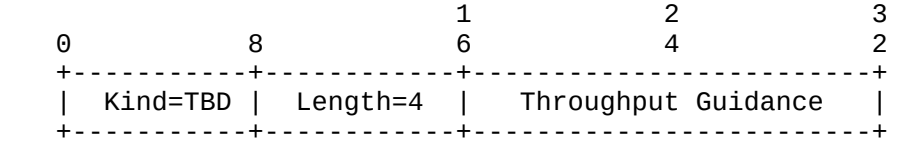
## 2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. TCP Option for SCONE

This document defines a TCP [RFC9293] option to provide analogous SCONE functionality for TCP. This could be viewed as similar to the way that TCP MSS clamping works traditionally, with the TCP MSS options included by endpoints being inspected and modified en-route by elements on the network path that can provide more pertinent guidance.

### 3.1. Option Format




The TCP option kind value (TBD) indicates the SCONE-TCP option. The length value of 4 is always used, along with a two byte throughput guidance.

The Throughput Guidance field is 16 bits and takes values based on the QUIC SCONE packet definitions of the "Rate Signal High Bits" [I-D.ietf-scone-protocol] (Section on "Rate Signals"). Only the lowest 7 bits of Throughput Guidance are currently used, and the highest 9 bits are zeroed. These may be used in a future extension, and should be ignored by implementations based on this current specification.

A rate limit is computed from the value of these 7 bits interpreted as an unsigned integer "n" ranging from 0 to 126, within the formula below.

rate limit = 100 kbps + 10^(n/20) kbps

### 3.2. Use During Handshake

Like other TCP options, SCONE-TCP is sent during connection establishment on SYN and SYN-ACK segments, and then only used subsequently if it has been successfully negotiated via use on the handshake.

Since it is used on the initial SYN, the SCONE-TCP option can serve as a "client hint" that informs the behavior of traffic limiting mechanisms within the network.

Since it is used on the SYN-ACK, the SCONE-TCP option can provide an immediate signal to the endpoint application about the advised bitrate that can help to inform the selection of media content to be requested subsequently within the application.

### 3.3. Use Post-Handshake

After TCP connection establishment with successful SCONE-TCP negotiation, the option can be used at any time. It does not need to be sent on every segment, and providing an update may be the sole reason for sending a segment. Since it takes up valuable header space, and will be inspected and operated on by network elements, it is not advisable to set on every segment that is transmitted. Instead, SCONE-TCP options may be included either periodically by an endpoint (e.g. every 10 seconds as a probe before requesting new media chunks) or in response to other events, such as at application request to help determine throughput guidance.

When an endpoint TCP desires to send the SCONE-TCP option, it can either include the option within the header of an outgoing segment carrying data (if there is user data to be sent), or may generate a pure ACK segment with the SCONE-TCP option.

Endpoints receiving segments with the SCONE-TCP option MUST NOT treat any pure ACKs that have SCONE-TCP as potential indicators of loss (i.e. these are not duplicate acknowledgements caused by gaps in the received data, and should not count towards triggering fast retransmission, for instance)..

ACKs that carry SACK information MAY include the SCONE-TCP option. Endpoints receiving these MAY use the SACK information to determine reordering, loss inference, and retransmission behavior.

## 4. API for TCP Applications

SCONE provides a signal to applications that can be used, for instance, to select proper media from manifests listing different available bitrates (e.g. at different resolutions, etc.) for video data. To that extent, it is important for the SCONE signal information to be made available to TCP applications. Relevant application programming interface (API) details are left to TCP implementations, though this section provides the outline of expected capabilities.

Since not all applications would be interested in SCONE throughput advice, the option might only be enabled for negotiation by specific application request. In that case, a TCP implementation supporting the typical "socket" API might define arguments for the "setsockopt" call to request SCONE use.

Similarly, the "getsockopt" call might be used in order to supply any received SCONE thoughput guidance back to the application. In some use cases, this may only need to happen once, early in the connection (e.g. after receiving a video manifest), while in other cases, an application may need to periodically poll the advice using getsockopt calls to sense if the advice may have changed over time.

## 5. Security Considerations

The security considerations for SCONE-TCP are similar to those for SCONE as present in QUIC, however, some differences arise because TCP security lacks the same cryptographic methods that are present in QUIC.

Middleboxes making changes to TCP headers (and options such as SCONE-TCP) might be considered as an attack, or used as part of an attack, although in general this is already common due to NAT, MSS clamping, and other network features.

TCP headers can be protected by TCP-MD5 [RFC2385], which is a legacy obsolete option, that does not cover the TCP options, so is compatible with the use of SCONE-TCP and the modification of SCONE-TCP options by middleboxes.

TCP-AO [RFC5925] replaces TCP-MD5 and can be configured to protect TCP options, or to leave TCP options uncovered by its MAC. If TCP-AO is used and configured to protect TCP options, then SCONE-TCP SHOULD NOT be used, as any modifications of it would cause segments to be rejected.

## 6. IANA Considerations

If this document is approved for Standards Track, a TCP option kind value should be allocated.

In early use, a TCP experimental option kind value can be used, with suggested ExID value 0x6f7d (to be registered with IANA). This matches 15 bits of both of the QUIC version numbers used for SCONE.

## 7. References

### 7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC9293] Eddy, W., Ed., "Transmission Control Protocol (TCP)", STD 7, RFC 9293, DOI 10.17487/RFC9293, August 2022, <https://www.rfc-editor.org/rfc/rfc9293>.

## 7.2. Informative References

[I-D.ietf-scone-protocol] Thomson, M., Huitema, C., Oku, K., Joras, M., and L. M. Ihlar, "Standard Communication with Network Elements (SCONE) Protocol", Work in Progress, Internet-Draft, draft-ietf-scone-protocol-04, 14 December 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-scone-protocol-04>.

[I-D.joras-scone-video-optimization-requirements] Joras, M., Tomar, A., Tiwari, A., and A. Frindell, "SCONE Video Optimization Requirements", Work in Progress, Internet-Draft, draft-joras-scone-video-optimization-requirements-01, 12 May 2025, <https://datatracker.ietf.org/doc/html/draft-joras-scone-video-optimization-requirements-01>.

[RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, DOI 10.17487/RFC2385, August 1998, <https://www.rfc-editor.org/rfc/rfc2385>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <https://www.rfc-editor.org/rfc/rfc5925>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/rfc/rfc9000>.

## Acknowledgments

This document represents collaboration and inputs from others, including:

* Alan Frindell

* Bryan Tan

* Anoop Tomar

## Authors' Addresses

Wesley Eddy
Meta
Email: wesleyeddy@meta.com

Matt Joras
Meta
Email: matt.joras@gmail.com