Standard Communication with Network Elements
Internet-Draft
Intended status: Standards Track
Expires: 1 November 2026

M. Duke
Google

30 April 2026

In-Band SCONE Reporting over QUIC
draft-duke-scone-scone-echo-02

Abstract

The SCONE protocol relies on the receiver of SCONE packets to send bandwidth estimates back to the sender via unspecified application-layer messages. In some cases, a peer might have SCONE receive capability at the QUIC layer but not implement the necessary application level functionality. A new QUIC frame that directly reports the contents of received SCONE packets can address these use cases. There are no changes in the interaction with SCONE Network Elements.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://martinduke.github.io/scone-echo/draft-duke-scone-scone-echo.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-duke-scone-scone-echo/.

Discussion of this document takes place on the Standard Communication with Network Elements Working Group mailing list (mailto:scone@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/scone. Subscribe at https://www.ietf.org/mailman/listinfo/scone/.

Source for this draft and an issue tracker can be found at https://github.com/martinduke/scone-echo.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The SCONE protocol ([SCONE]) allows networks to provide bandwidth guidance to endpoints. Senders prepend a SCONE header to QUIC ([QUIC]) packets that include a 7-bit bandwidth field. Network elements can update this field. The receiver of SCONE packets reports the received value to the application. The application can use this information to adjust the bit rate, either by directly reporting the value back to the sender at the application layer, or by using it to make some other adjustment to the incoming traffic.

This architecture requires cooperation from the application layer: a receiver cannot usefully process a SCONE packet without application involvement to take action on the result. In principle, a QUIC implementation could _send_ SCONE packets solely based on the receiver's advertised ability to receive, but it might have difficulty determining the correct rate to send such packets. The receiver would need to effectuate any behavior changes without SCONE-aware cooperation from the sender. The authors are not aware of any deployments that send SCONE packets without explicit confirmations from the application layer.

There are some use cases where it would be useful to not require cooperation from the receiving application, instead returning feedback directly at the QUIC layer. There are fewer QUIC implementations than applications. A QUIC implementation might support SCONE, but the intervening layers do not provide SCONE APIs. For example, a browser could use a third-party QUIC implementation that supports SCONE, but not provide the JavaScript APIs to enable and process SCONE. If the QUIC implementation could directly return feedback to the sender, then only application support at the sender is required.

This document proposes an extension to the QUIC protocol defining a new QUIC frame that echoes received SCONE feedback directly to the sender at the QUIC layer.

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. Overview

The use of the SCONE_ECHO frame is negotiated by new transport parameters separately in each direction. This negotiation is an alternate means of enabling the use of SCONE packets, in addition to scone_supported from [SCONE]. For a given direction, sending SCONE is authorized by the new transport parameters or scone_supported, never both.

When an endpoint receives a valid SCONE packet and SCONE echo was negotiated in that direction, it sends a SCONE_ECHO QUIC frame.

Upon receipt of a valid SCONE_ECHO packet, the SCONE sender reports the bandwidth advice to its local application layer for further action.

There are no changes to SCONE Network Element behavior or the SCONE packet format from [SCONE]. SCONE packets are still only valid if another QUIC packet in the UDP datagram is successfully decrypted.

4. The SCONE_ECHO Frame

An endpoint uses the SCONE_ECHO frame to return the 7-bit value encoded in a SCONE packet. The conditions for sending it are described in Section 3.

```
SCONE_ECHO Frame {
  Type (i) = 0xff005345,
  Packet Number (i),
  Zero (1),
  Throughput Advice (7),
}
```

Packet Number: the full (62-bit) packet number of the first successfully decrypted QUIC packet in the UDP datagram that contained the SCONE header.

Zero: This bit MUST be zero and MUST be ignored on receipt.

Throughput Advice: The Rate Signal in the SCONE packet as encoded in Section 5 of [SCONE].

A SCONE sender SHOULD keep track of the Packet Numbers to which it prepended SCONE headers, and MUST ignore any SCONE_ECHO frames where it does not have a record of prepending SCONE to that packet number. It might not store such numbers when it hits storage limitations or receives duplicate SCONE_ECHO frames.

SCONE_ECHO frames are retransmittable and MUST only appear in 1-RTT packets, because a succesfully decrypted 1-RTT packet indicates all transport parameters have been verified. However, the Packet Number field can refer to a packet number in any packet number space.

The arrival of a SCONE packet triggers a new SCONE_ECHO frame and cancels the retransmission of any previous SCONE_ECHO frame. Implementations MAY store the most recent value if a SCONE_ECHO frame is already in flight and wait until it is acknowledged or lost before sending the latest value to naturally rate limit SCONE_ECHO to approximately once per round trip. As a result, SCONE senders cannot expect one SCONE_ECHO frame per SCONE packet sent.

5. Negotiating SCONE Echo

This document specifies two new transport parameters: scone_echo_send and scone_echo_receive.

Endpoints send scone_echo_send to indicate they will send SCONE_ECHO frames in response to valid SCONE packets. An endpoint MUST NOT send both scone_echo_send and scone_supported; doing so is a TRANSPORT_PARAMETER_ERROR.

Endpoints send scone_echo_receive to indicate the ability to process SCONE_ECHO frames.

Endpoints MUST NOT send SCONE packets unless the peer has sent either scone_supported or scone_echo_send. If the peer sent scone_echo_send, the endpoint MUST also have sent scone_echo_receive.

Endpoints MUST NOT send SCONE_ECHO frames unless it has sent scone_echo_send and the peer has sent scone_echo_receive.

scone_echo_send and scone_echo_receive MUST be empty. If not empty, it MUST be treated as a connection error of type TRANSPORT_PARAMETER_ERROR.

These transport parameters are valid for QUIC Version 1 [RFC9000], QUIC Version 2 [RFC9369], and any other version that supports SCONE as outlined in Section 6 of [SCONE].

These transport parameters MUST NOT be stored for 0-RTT purposes.

5.1. The SCONE indicator

A client that sends the scone_echo_send or scone_echo_receive transport parameter MUST send the SCONE Indicator as described in Section 6.1 of [SCONE], whether or not it also sends scone_supported. Its semantic meaning remains unchanged.

6. Applicability

In general, the scone_supported transport parameter from [SCONE] indicates that the sender has a local application that is willing to accept bandwidth advice, potentially including sending that information to the SCONE sender via application-layer messaging.

If this is the case, a QUIC endpoint SHOULD NOT send scone_echo_send, as application-layer approaches can incorporate various receiver-side actions as well as more bandwidth-efficient signals to the sender.

A QUIC implementation that does not have application-layer cooperation can send scone_echo_send instead to enable a purely sender-side approach.

QUIC implementations will generally not send SCONE packets without a request from the local application. An endpoint that wishes to send SCONE packets and supports this specification SHOULD send scone_echo_receive in case the peer is unable to support an application-layer response.

[Note: It is possible to revise this specification to allow the SCONE receiver to send both SCONE_ECHO and report to the application, thought this risks duplicate signaling and complicates reasoning about application response. Similarly, it is possible to allow a SCONE sender to signal preference for either SCONE_ECHO or application response, although this would further complicate negotiation. Nevertheless, both are viable options if the Working Group desires it.]

7. Security Considerations

The security considerations in Section 9 of [SCONE] apply.

8. Privacy Considerations

Section 10 of [SCONE] describes the potential privacy exposure of using SCONE. Requiring application-layer engagement provides an additional layer of consent to this exposure, although such engagement may not extend to the actual user.

This document envisions SCONE Echo being enabled by default in some QUIC implementations. This might actually obscure application fingerprinting, but it also further distances consent from the user.

SCONE Echo envisions a widely deployed network of endpoints willing to send network bandwidth advice to the sender. This makes it much easier for a observer to obtain a map of bandwidth advice from its location.

9. IANA Considerations

9.1. scone_echo_send Transport Parameter

The document registers the scone_echo_send transport parameter in the "QUIC Transport Parameters" registry maintained at https://www.iana.org/assignments/quic, following the guidance from Section 22.3 of [QUIC].

Value: 0xff002200

Parameter Name: scone_echo_send

Status: Provisional

Specification: This document

Date: This date

Change Controller: IETF (iesg@ietf.org)

Contact: QUIC Working Group (quic@ietf.org)

Notes: (none)

9.2. scone_echo_receive Transport Parameter

The document registers the scone_echo_receive transport parameter in the "QUIC Transport Parameters" registry maintained at https://www.iana.org/assignments/quic, following the guidance from Section 22.3 of [QUIC].

Value: 0xff002201

Parameter Name: scone_echo_receive

Status: Provisional

Specification: This document

Date: This date

Change Controller: IETF (iesg@ietf.org)

Contact: QUIC Working Group (quic@ietf.org)

Notes: (none)

9.3. SCONE_ECHO frame

This document registers the SCONE_ECHO frame in the "QUIC Frame Types" registry.

value: 0xff005345

name: SCONE_ECHO

Status: Provisional

Specification: Section 4

Date: This date

Change Controller: IETF (iesg@ietf.org)

Contact: QUIC Working Group (quic@ietf.org)

Pkts: 1-RTT

10. References

10.1. Normative References

[QUIC] Thomson, M., "Version-Independent Properties of QUIC", RFC 8999, DOI 10.17487/RFC8999, May 2021, <https://www.rfc-editor.org/rfc/rfc8999>.

[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>.

[SCONE] 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>.

10.2. Informative References

[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>.

[RFC9369] Duke, M., "QUIC Version 2", RFC 9369, DOI 10.17487/RFC9369, May 2023, <https://www.rfc-editor.org/rfc/rfc9369>.

Acknowledgments

TODO acknowledge.

Author’s Address

Martin Duke
Google
Email: martin.h.duke@gmail.com