SCONE
Internet-Draft
Intended status: Informational
Expires: 23 April 2026

M. Thomson
Mozilla
C. Huitema
Private Octopus Inc.
奥 一穂 (K. Oku)
Fastly
M. Joras
Meta
M. Ihlar
Ericsson

20 October 2025

Standard Communication with Network Elements (SCONE) Protocol
draft-ietf-scone-protocol-03

Abstract

This document describes a protocol where on-path network elements can give endpoints their perspective on what the maximum achievable throughput might be for QUIC flows.

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://ietf-wg-scone.github.io/scone/draft-ietf-scone-protocol.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-scone-protocol/.

Discussion of this document takes place on the SCONE 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/ietf-wg-scone/scone.

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/.

This Internet-Draft will expire on 23 April 2026.

Copyright Notice

Table of Contents

1. Introduction 3
2. Overview 4
3. Applicability 5
  3.1. Independent of Congestion Signals 5
  3.2. Unspecified Scope 5
  3.3. Per-Flow Signal 6
  3.4. Undirectional Signal 6
  3.5. Advisory Signal 6
  3.6. Following Advice 7
4. Conventions and Definitions 7
5. SCONE Packet 8
  5.1. Rate Signals 9
  5.2. Monitoring Period 10
  5.3. Endpoint Processing of SCONE Packets 11
  5.4. Following Throughput Advice 12
6. Negotiating SCONE 13
  6.1. Indicating Support on New Flows 13
  6.2. Limitations of Indication 13
  6.3. Indications for Migrated Flows 14
7. Deployment 14
  7.1. Applying Throughput Advice Signals 14
  7.2. Monitoring Flows 15
  7.3. Flows That Exceed Throughput Advice 17
8. Version Interaction 17

8.1. Providing Opportunities to Apply Throughput Advice Signals 17
8.2. Feedback To Sender About Signals 18
8.3. Interactions with Congestion Control 19
9. Security Considerations 20
9.1. Flooding intermediaries with fake packets 20
9.2. Damage to Other Protocols 21
10. Privacy Considerations 22
10.1. Passive Attacks 22
10.2. Active Attacks 23
11. IANA Considerations 24
11.1. SCONE Versions 24
11.2. scone_supported Transport Parameter 24
12. References 24
12.1. Normative References 24
12.2. Informative References 25
Appendix A. Sliding Window for Rate Limit Monitoring 26
Acknowledgments 26
Authors' Addresses 26

1. Introduction

Many access networks apply rate limits to constrain the data rate of attached devices. This is often done without any indication to the applications running on devices. The result can be that application performance is degraded, as the manner in which rate limits are enforced can be incompatible with the rate estimation or congestion control algorithms used at endpoints.

Having the network indicate what its rate limiting policy is, in a way that is accessible to endpoints, allows applications to use this information when adapting their send rate.

Network elements are not limited to communicating information about rate limiting policies. Network elements in access networks could provide information to endpoints that can help account for changes in network capacity that are not suited to congestion control feedback. This might include reduced capacity due to overuse, equipment faults, or other transient issues; conversely, networks might choose to signal increased availability of capacity.

The Standard Communication with Network Elements (SCONE) protocol is negotiated by QUIC endpoints. This protocol provides a means for network elements to signal the maximum available sustained throughput, or rate limits, for flows of UDP datagrams that transit that network element to a QUIC endpoint.

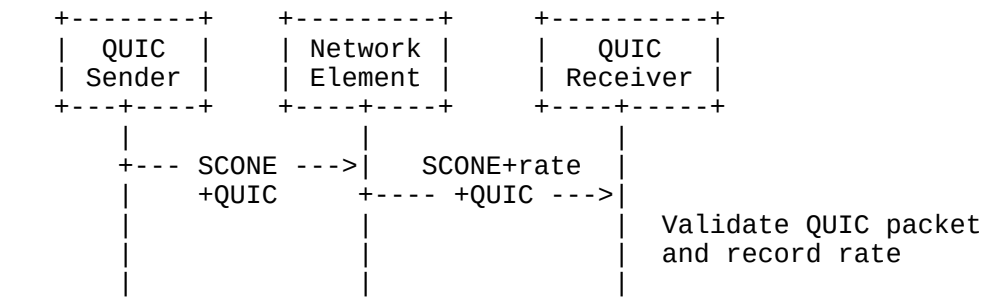
Networks with rate limiting policies use SCONE to send throughput advice to cooperating endpoints to limit overall network usage. Where congestion control signals -- such as ECN, delays and loss -- operate on a time scale of a round trip time, throughput advice operates over a much longer period. This has benefits in some networks as endpoints can fully consume network capacity in bursts, rather than extending network interaction at lower rates.

For endpoints, SCONE throughput advice makes network policies visible, which can reduce wasteful probing beyond those limits.

2. Overview

QUIC endpoints can negotiate the use of SCONE by including a transport parameter (Section 6) in the QUIC handshake. Endpoints then occasionally coalesce a SCONE packet with ordinary QUIC packets that they send.

Network elements that have rate limiting policies can detect flows that include SCONE packets. The network element can indicate a maximum sustained throughput by modifying the SCONE packet as it transits the network element.



QUIC endpoints that receive modified SCONE packets observe the indicated version, process the QUIC packet, and then record the indicated rate.

Throughput advice in each direction -- the client-to-server direction and the server-to-client direction -- are independent. It is expected that these rates will differ for multiple reasons, which include asymmetry of link capacity and path diversity. Applications may choose to enable SCONE signals in either or both directions as they see fit.

Indicated rate limits only apply to the path on which they are received. A connection that migrates or uses multipath [QUIC-MP] cannot assume that rate limit indications from one path apply to new paths.

3. Applicability

This protocol only works for flows that use the SCONE packet (Section 5).

The protocol requires that packets are modified as they transit a network element, which provides endpoints strong evidence that the network element has the power to apply rate limiting; though see Section 9 for potential limitations on this.

The throughput advice signal that this protocol carries is independent of congestion signals, limited to a single path and UDP packet flow, unidirectional, and strictly advisory.

3.1. Independent of Congestion Signals

Throughput advice signals are not a substitute for congestion feedback. Congestion signals, such as acknowledgments, provide information on loss, delay, or ECN markings [ECN] that indicate the real-time condition of a network path. Congestion signals might indicate a throughput that is different from the signaled rate limit.

Endpoints cannot assume that a signaled rate limit is achievable if congestion signals indicate otherwise. Congestion could be experienced at a different point on the network path than the network element that indicates a rate limit. Therefore, endpoints need to respect the send rate constraints that are set by a congestion controller.

3.2. Unspecified Scope

Modifying a packet does not prove that the throughput that is indicated would be achievable. A signal that is sent for a specific flow is likely enforced at a different scope. The extent of that scope is not carried in the signal.

For instance, limits might apply at a network subscription level, such that multiple flows receive the same signal.

Endpoints can therefore be more confident in the throughput signal as an indication of the maximum achievable throughput than as any indication of expected throughput. That throughput will only be achievable when there is no significant data flowing in the same scope. In the presence of other flows, congestion limits are likely to determine actual throughput.

This makes the application of signals most usefully applied to a downlink flow in access networks, close to an endpoint. In that case, capacity is less likely to be split between multiple active flows.

3.3. Per-Flow Signal

The same UDP address tuple might be used for multiple QUIC connections. A single signal might be lost or only reach a single application endpoint. Network elements that signal about a flow might choose to send additional signals, using connection IDs to indicate when new connections could be involved.

3.4. Undirectional Signal

The endpoint that receives a throughput advice signal is not the endpoint that might adapt its sending behavior as a result of receiving the signal. This ensures that the throughput advice signal is attached to the flow that it is mostly likely to apply to.

An endpoint might need to communicate the value it receives to its peer in order to ensure that the limit is respected. This document does not define how that signaling occurs as this is specific to the application in use.

3.5. Advisory Signal

A signal does not prove that a higher rate would not be successful. Endpoints that receive this signal therefore need to treat the information as advisory.

The fact that an endpoint requests bitrate signals does not necessarily mean that it will adhere to them; in some cases, the endpoint cannot. For example, a flow may initially be used to serve video chunks, with the client selecting appropriate chunks based on bitrate signals, but later switch to a bulk download for which bitrate adaptation is not applicable. Composite flows from multiple applications, such as tunneled flows, might only have a subset of the involved applications that are capable of handling SCONE signals. Therefore, when a network element detects a flow using more bandwidth than advertised via SCONE, it might switch to applying its policies for non-SCONE flows, using congestion control signals.

The signaled advice applies to the flow of packets on the same UDP address tuple for the duration of the current monitoring period, unless it is updated earlier or the flow ends; see Section 5.2 for details on the monitoring period. Rate limiting policies often apply on the level of a device or subscription, but endpoints cannot assume that this is the case. A separate signal can be sent for each flow.

Network conditions and rate-limit policies can change in ways that make previously signaled advice obsolete. There are no guarantees that updated advice will be sent at such events.

3.6. Following Advice

The SCONE throughput advice is advisory (see Section 3.5). Applications that chose to follow it will do so in the way that best suits their needs.

The most obvious way to keep within the limits set by throughput advice is to inform the sending peer of the limit so that the peer can do whatever rate limiting is necessary. Alternatively, a receiver can control the release of flow control credit (see Section 4 of [QUIC]) to indirectly limit the sending rate of a peer.

Some applications offer options for rate control that can offer superior outcomes. Real-time and streaming video applications are able to adjust video quality to fit within a target throughput. For instance, an HTTP Live Streaming client [HLS] can ask for lower bitrate, lower quality media segments.

4. 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 [BCP14] when, and only when, they appear in all capitals, as shown here.

5. SCONE Packet

A SCONE packet is a QUIC long header packet that follows the QUIC invariants; see Section 5.1 of [INVARIANTS].

Figure 1 shows the format of the SCONE packet using the conventions from Section 4 of [INVARIANTS].

```
SCONE Packet {
  Header Form (1) = 1,
  Reserved (1),
  Rate Signal High Bits (6),
  Version (32) = 0x6f7dc0fd or 0xef7dc0fd,
  Destination Connection ID Length (8),
  Destination Connection ID (0..2040),
  Source Connection ID Length (8),
  Source Connection ID (0..2040),
}
```

Figure 1: SCONE Packet Format

The most significant bit (0x80) of the packet indicates that this is a QUIC long header packet. The next bit (0x40) is reserved and can be set according to [QUIC-BIT].

The Rate Signal High Bits field consists of the low six bits (0x3f) of the first byte. Together with the most significant bit of the Version field, this forms the 7-bit Rate Signal. Values for the Rate Signal are described in Section 5.1.

The Version field contains either 0x6f7dc0fd or 0xef7dc0fd. The only difference between these two values is the most significant bit, which also contributes to the Rate Signal. All other bits are identical, which facilitates detection and modification of SCONE packets.

This packet includes a Destination Connection ID field that is set to the same value as other packets in the same datagram; see Section 12.2 of [QUIC].

The Source Connection ID field is set to match the Source Connection ID field of any packet that follows. If the next packet in the datagram does not have a Source Connection ID field, which is the case for packets with a short header (Section 5.2 of [INVARIANTS]), the Source Connection ID field is empty.

SCONE packets MUST be included as the first packet in a datagram. This is primarily to simplify the process of updating throughput advice in network elements. This is also necessary in many cases for QUIC versions 1 and 2 because packets with a short header cannot precede any other packets.

5.1. Rate Signals

A Rate Signal is a 7-bit unsigned integer (0-127). The high six bits are the Rate Signal High Bits, and the least significant bit is the most significant bit of the Version field.

When sent by a QUIC endpoint, the Rate Signal is set to 127. Receiving a value of 127 indicates that throughput advice is unknown, either because network elements on the path are not providing advice or they do not support SCONE. All other values (0 through 126) represent the ceiling of rates advised by the network element(s) on the path.

The rate limits follow a logarithmic scale defined as:

* Base rate (b_min) = 100 Kbps

* Bitrate at value n = b_min * 10^(n/20)

where n is an integer between 0 and 126 represented by the Rate Signal.

Table 1 lists some of the potential values for signals and the corresponding bitrate for each.

Bitrate	Rate Signal
100 Kbps	0
112 Kbps	1
126 Kbps	2
141 Kbps	3
1 Mbps	20
1.12 Mbps	21
10 Mbps	40
11.2 Mbps	41
100 Mbps	60
112 Mbps	61
1 Gbps	80
1.12 Gbps	81
10 Gbps	100
11.2 Gbps	101
100 Gbps	120
112 Gbps	121
199.5 Gbps	126
Unknown	127

Table 1: Examples of SCONE signals and corresponding rates

5.2. Monitoring Period

The time over which throughput advice applies is defined to be a period of 67 seconds.

Protocol participants can use a different period, depending on their role. Senders can limit their send rate over any time period up to 67 seconds. Network elements can monitor and apply limits to send rates using time period of at least 67 seconds.

The choice of 67 seconds is a compromise between competing interests. Longer periods allow applications more flexibility in terms of how to allocate bandwidth over time. Shorter periods allow networks to administer policies more tightly. Also, when circumstances change, applications are better able recover without exceeding limits for a significant time if they have exceeded limits.

The choice of 67 seconds, as a prime number, also helps avoid synchronization with other periodic effects that are commonly measured in whole seconds. This includes segment length or key frame intervals in video applications, but also includes timers for middleboxes; see Section 4.3 of [RFC4787]. Any repeating phenomenon at a 67 second interval is therefore unlikely to be due to other periodic effects.

A sample algorithm for ensuring adherance to throughput advice is included in Appendix A.

5.3. Endpoint Processing of SCONE Packets

Processing a SCONE packet involves reading the value from the Rate Signal field. However, this value MUST NOT be used unless another packet from the same datagram is successfully processed. Therefore, a SCONE packet always needs to be coalesced with other QUIC packets.

A SCONE packet is defined by the use of the long header bit (0x80 in the first byte) and the SCONE protocol version (0x6f7dc0fd or 0xef7dc0fd in the next four bytes). The 7-bit Rate Signal can be extracted by combining the low 6 bits of the first byte with the most significant bit of the version field. A SCONE packet MAY be discarded, along with any packets that come after it in the same datagram, if the Source Connection ID is not consistent with those coalesced packets, as specified in Section 5.

A SCONE packet is discarded if the rate signal is unknown (127).

A SCONE packet MUST be discarded if the Destination Connection ID does not match one recognized by the receiving endpoint.

If a connection uses multiple DSCP markings [RFC2474], the throughput advice that is received on datagrams with one marking might not apply to datagrams that have different markings.

5.4. Following Throughput Advice

Endpoints that receive throughput advice can advise their peer of the limit so that the peer might limit the amount of data it sends over any monitoring period (Section 5.2). Alternatively, the endpoint might change its own behavior to effect a similar outcome indirectly, which might use flow control or changes to request patterns.

An endpoint that receives throughput advice might receive multiple different rate limits. If advice is applied by applications, applications MUST apply the lowest throughput advice received during any monitoring period; see Section 5.2.

After a monitoring period (Section 5.2) without receiving throughput advice, any previous advice expires. Endpoints can remove any constraints placed on throughput based on receiving throughput advice. This does not mean that there are no limits, either in policy or due to network conditions, only that these limits are now unknown. Other constraints on usage will still apply, which necessarily includes congestion control and might include other, application-specific constraints.

Applications MAY discard throughput usage state when they receive throughput advice that indicates a reduced rate. Otherwise, data that was sent at a higher rate might force the available rate to zero for the remainder of the monitoring period; see also Section 7.2.

The relatively long duration of the monitoring period means that this is preferable to disabling sending completely. The cost is that loss of information about recent send rate might result in temporarily exceeding the rates indicated by throughput advice. In comparison, applications retain throughput usage state when throughput advice increases.

This approach ensures that network elements are able to reduce the frequency with which they send updated signals to as low as once per monitoring period. However, applying signals at a low frequency risks throughput advice being reset if no SCONE packet is available for applying signals (Section 7.1), or the rewritten packets are lost. Sending the signal multiple times increases the likelihood that the signal is received.

6. Negotiating SCONE

A QUIC endpoint indicates that it is willing to receive SCONE packets by including the scone_supported transport parameter (0x219e). The scone_supported transport parameter MUST be empty. Receiving a non-zero length scone_supported transport parameter MUST be treated as a connection error of type TRANSPORT_PARAMETER_ERROR; see Section 20.1 of [QUIC].

This transport parameter is valid for QUIC versions 1 [QUIC] and 2 [QUICv2] and any other version that recognizes the versions, transport parameters, and frame types registries established in Sections 22.2, 22.3, and 22.4 of [QUIC].

6.1. Indicating Support on New Flows

All new flows that are initiated by a client that supports SCONE MUST include bytes with values 0xc8 and 0x13 as the last two bytes of datagrams that commence a new flow if the protocol permits it. This indication MUST be sent in every datagram until the client receives any datagram from the server, at which point the client can be confident that the indication was received.

A client that uses a QUIC version that includes length-delimited packets, which includes QUIC versions 1 [QUIC] and 2 [QUICv2], can include an indicator of SCONE support at the end of datagrams that start a flow. The handshakes of these protocols ensures that the indication can be included in every datagram the client sends until it receives a response -- of any kind -- from the server.

6.2. Limitations of Indication

This indication does not mean that SCONE signals will be respected, only that the client is able to negotiate SCONE. A server might not support SCONE or might choose not to send SCONE packets. Finally, applications might be unable to apply throughput advice or choose to ignore it.

This indication being just two bytes means that there is a non-negligible risk of collision with other protocols or even QUIC usage without SCONE indications. This means that the indication alone is not sufficient to indicate that a flow is QUIC with the potential for SCONE support.

Despite these limitations, having an indication might allow network elements to change their starting posture with respect to their enforcement of their rate limit policies.

6.3. Indications for Migrated Flows

Applications MAY decide to indicate support for SCONE on new flows, including when migrating to a new path (see Section 9 of [QUIC]). In QUIC version 1 and 2, the two byte indicator cannot be used.

Sending a SCONE packet for the first few packets on a new path gives network elements on that path the ability to recognize the flow as being able to receive throughput advice and also gives the network element an opportunity to provide that throughput advice.

To enable this indication, even if an endpoint would not otherwise send SCONE packets, endpoints can send a SCONE packet any time they send a QUIC PATH_CHALLENGE or PATH_RESPONSE frame. This applies to both client and server endpoints, but only if the peer has sent the transport parameter; see Section 6.

7. Deployment

QUIC endpoints can enable the use of the SCONE protocol by sending SCONE packets Section 5. Network elements then apply or replace the Rate Signal field (Section 7.1) according to their policies.

7.1. Applying Throughput Advice Signals

A network element detects a SCONE packet by observing that a packet has a QUIC long header and one of the SCONE protocol versions (0x6f7dc0fd or 0xef7dc0fd).

A network element then conditionally replaces the most significant bit of the Version field and the Rate Signal High Bits field with values of its choosing.

A network element might receive a packet that already includes a rate signal. The network element replaces the rate signal if it wishes to signal a lower rate limit; otherwise, the original values are retained, preserving the signal from the network element with the lower policy.

The following pseudocode indicates how a network element might detect a SCONE packet and replace the existing rate signal (packet_signal) with a new rate signal (target_signal) that encodes the throughput advice of this network element.

```
is_long = packet[0] & 0x80 == 0x80
packet_version = ntohl(packet[1..5])
if is_long and (packet_version & 0x7fffffff) == SCONE_VERSION_BITS:
  packet_signal = ((packet[0] & 0x3f) << 1) | (packet_version >> 31)
  if target_signal < packet_signal:
    packet[0] = (packet[0] & 0xc0) | (target_signal >> 1)
    packet[1] = (packet[1] & 0x7f) | (target_signal << 7)
```

Once the throughput advice signal is updated, the network element updates the UDP checksum for the datagram.

A network element needs to ensure that it sends updated rate signals with no more than a monitoring period (Section 5.2) between each update. Because this depends on the availability of SCONE packets and packet loss can cause signals to be missed, network elements might need to update more often.

At the start of a flow, network elements are encouraged to update the rate signal of the first few SCONE packets it observes so that endpoints can obtain throughput advice early.

Senders that send a SCONE packet or network elements that update SCONE packets every 2030 seconds is likely sufficient to ensure that throughput advice is not lost. To reduce the risk of synchronization across multiple senders, which may cause network elements to miss updates, senders can include a small random delay.

7.2. Monitoring Flows

Network elements can monitor flows to determine which flows are from applications that respect throughput advice. This section outlines an exemplary algorithm for network elements.

This monitoring algorithm is guidance only. Network elements are advised that monitoring any more strictly than the following likely invalidates their advice. Network elements could choose not to monitor in this way or use a looser monitoring approach. For instance, monitoring over a longer time window than the monitoring period (67s) or using a higher rate than is signaled is possible.

To operate this algorithm, the minimal state a network element maintains is:

* the current rate limit,

* any state necessary to monitor throughput (that is, throughput usage state, such as the state in Appendix A),

* a timer used for rate increases, and

* the time at which that rate limit was last updated.

When advice is set or updated, the network element waits until it receives the next SCONE packet on affected flows. It then updates the throughput advice in that packet (Section 7.1) and updates its monitoring state as follows:

* If the signaled rate limit is the same as the current rate limit, set the last update time to the current time.

* If the signaled rate is higher than the current value, then:

  - If the time since the last SCONE packet was updated is greater than the monitoring period, the current rate limit is increased to the received value and the last update time is set to the current time.

  - Otherwise, the throughput advice is deferred. The implementation sets the rate increase timer to one monitoring period relative to the time that the last SCONE packet was updated. When the timer lapses, change the rate monitoring to the higher rate.

* If the signaled rate is lower than the current value, set the current rate limit to match the advice, cancel any rate increase timer, discard all rate monitoring state, and set the last update time to the current time.

A network element that reduces the throughput advice it provides MUST also discard any state it maintains about the use of the network under previous, higher rates. This is necessary to allow senders to discard state when advice is reduced (see Section 5.4). This avoids cases where an application that has planned usage might otherwise be completely unable to send due to a lower limits.

A network element that reduces its throughput advice might also need delay monitoring or enforcement to allow applications time to receive and act on the updated advice.

A network element MUST NOT alter datagrams to add SCONE packets or synthesize datagrams that contain SCONE packets. The latter will not be accepted and the former, even if they do not exceed the path MTU as a result, can be detected by applications and could be ignored. This document does not define a mechanism to support detection, but one might be added in future.

7.3. Flows That Exceed Throughput Advice

Network elements that provide throughput advice can monitor flows -- or sets of flows that are subject to the same throughput limit -- for adherance to that advice.

In the event that a flow exceeds these limits, a network element could immediately start enforcing adherence to the advice as though it were a hard limit. However, this risks creating a situation where communication ceases entirely for a significant period of time; that is, up to the period defined in Section 5.2.

A better approach is to exclude any data transmitted before sending reduced throughput advice from any monitoring that uses the reduced rate. This ensures that the enforcement of limits is minimally disruptive.

8. Version Interaction

The SCONE protocol defines two versions (0x6f7dc0fd and 0xef7dc0fd) that cover different ranges of bitrates. This design allows for:

* Support for both very low bitrates (down to 100 Kbps) and very high bitrates (up to 199.5 Gbps)

* Graceful handling of network elements that might only recognize one version.

8.1. Providing Opportunities to Apply Throughput Advice Signals

Endpoints that wish to offer network elements the option to add throughput advice signals can send SCONE packets at any time. This is a decision that a sender makes when constructing datagrams.

When sending SCONE packets, enpoints MUST include the SCONE packet as the first packet in a datagram, coalesced with additional packets.

Upon confirmation that the peer is willing to receive SCONE packets, an endpoint SHOULD include SCONE packets in the first few UDP datagrams that it sends. Doing so increases the likelihood of eliciting early throughput advice from network elements, allowing applications to apply that advice from the early stages of the data transfer.

After that, endpoints that seek to receive throughput advice on a flow MUST send a SCONE packet at least twice each monitoring period; see Section 5.2.

Sending SCONE packets more often might be necessary to:

Avoid missing advice: If SCONE packets are not sent, updated, and received for an entire monitoring period, an application might incorrectly assume that no advice is being provided.

Reduce latency: The time between SCONE packets determines the maximum delay between changes in throughput advice and when that advice can be received and acted upon.

A sender can track the receipt of the coalesced QUIC packet and send another SCONE packet when loss is detected. However, it is likely simpler to send SCONE packets more often.

Sending a SCONE packet every 2030 seconds is likely sufficient to ensure that throughput advice is not lost, though endpoints might send a packet every few seconds to improve responsiveness. This period could be determined by how quickly an application is able to respond to a change in throughput advice.

For example, a streaming application that fetches video segments that are 5 seconds in length might send SCONE packets on a similar cadence. A real-time conferencing application might send more often. In either case, the length of the monitoring period (Section 5.2) limits how fast any application can react.

Though sending SCONE packets more than once each round trip time might help reduce exposure to packet loss, it is better to spread updates over time rather than to send multiple SCONE packets in less frequent bursts.

The main cost associated with sending SCONE packets is the reduction in available space in datagrams for application data.

A network element that wishes to signal an updated rate limit waits for the next SCONE packet in the desired direction; see Section 7.1.

8.2. Feedback To Sender About Signals

Information about throughout advice is intended for the sending application. Any signal from network elements can be propagated to the receiving application using an implementation-defined mechanism.

This document does not define a means for indicating what was received. That is, the expectation is that any signal is propagated to the application for handling, not handled automatically by the transport layer. How a receiving application communicates the throughput advice signal to a sending application will depend on the application in use.

Different applications can choose different approaches. For example, in an application where a receiver drives rate adaptation, it might not be necessary to define additional signaling.

A sender can use any acknowledgment mechanism provided by the QUIC version in use to learn whether datagrams containing SCONE packets were likely received. This might help inform whether to send additional SCONE packets in the event that a datagram is lost. However, rather than relying on transport signals, an application might be better able to indicate what has been received and processed.

SCONE packets could be stripped from datagrams in the network, which cannot be reliably detected. This could result in a sender falsely believing that no network element applied a throughput advice signal.

8.3. Interactions with Congestion Control

SCONE and congestion control both provide the application with estimates of a path capacity. They are complementary. Congestion control algorithms are typically designed to quickly detect and react to congestion, i.e., to the "minimum" capacity of a path. SCONE informs the endpoint of the maximum capacity of a path based on network rate limit policy, network conditions, or a combination of the two.

Consider for example a path in which the bottleneck router implements Early Congestion Notification as specified in the L4S architecture [RFC9330]. If the path capacity diminishes, queues will build up and the router will immediately start increasing the rate at which packets are marked as "Congestion Experienced". The receiving endpoint will notice these marks, and inform its peer. The incoming congestion will be detected within 1 round trip time (RTT). This scenario will play out whatever the reason for the change in capacity, whether due to increased competition between multiple applications or, for example, to a change in capacity of a wireless channel.

9. Security Considerations

The modification of packets provides endpoints proof that a network element is in a position to drop datagrams and could apply a rate limit policy. Section 8.1 states that endpoints only accept signals if the datagram contains a packet that it accepts to prevent an off-path attacker from inserting spurious throughput advice signals.

Some off-path attackers may be able to both observe traffic and inject packets. Attackers with such capabilities could observe packets sent by an endpoint, create datagrams coalescing an arbitrary SCONE packet and the observed packet, and send these datagrams such that they arrive at the peer endpoint before the original packet. Spoofed packets that seek to advertise a higher limit than might otherwise be permitted also need to bypass any rate limiters. The attacker will thus get arbitrary SCONE packets accepted by the peer, with the result being that the endpoint receives a false or misleading rate limit.

The recipient of a throughput advice signal therefore cannot guarantee that the signal was generated by an on-path network element. However, the capabilities required of an off-path attacker are substantially similar to those of on path elements.

The actual value of the throughput advice signal is not authenticated. Any signal might be incorrectly set in order to encourage endpoints to behave in ways that are not in their interests. Endpoints are free to ignore limits that they think are incorrect. The congestion controller employed by a sender provides real-time information about the rate at which the network path is delivering data.

Similarly, if there is a strong need to ensure that a rate limit is respected, network elements cannot assume that the signaled limit will be respected by endpoints.

9.1. Flooding intermediaries with fake packets

Attackers that can inject packets may compose arbitrary "SCONE-like" packets by selecting a pair of IP addresses and ports, an arbitrary rate signal, a valid SCONE version number, an arbitrary "destination connection ID", and an arbitrary "source connection ID". The SCONE packet will carry these information. A coalesced "1RTT" packet will start with a plausible first octet, and continue with the selected destination connection ID followed by a sufficiently long series of random bytes, mimicking the content of an encrypted packets.

The injected packets will travel towards the destination. The final destination will reject such packets because the destination ID is invalid or because decryption fail, but network elements cannot do these checks, and will have to process the packets. All the network elements between the injection point and the destination will have to process these packets.

Attackers could send a high volume of these "fake" SCONE packets in a denial of service (DOS) attempt against network elements. The attack will force the intermediaries to process the fake packets. If network elements are keeping state for ongoing SCONE flows, the attack can cause the excessive allocation of memory resource. The mitigation there will be the same as mitigation of other distributed DOS attacks: limit the rate of SCONE packets that a network element is willing to process; possibly, implement logic to distinguish valid SCONE packets from fake packets; or, use generic protection against Distributed DOS attacks.

Attackers could also try to craft the fake SCONE packets in ways that trigger a processing error at network elements. For example, they might pick connection identifiers of arbitrary length. Network elements can mitigate these attacks with implementations that fully conform to the specification of Section 5.

9.2. Damage to Other Protocols

Network elements that update SCONE packet fields might do that for datagrams exchanged in other protocols. This could result in damage to those protocols.

The most serious damage occurs when every datagram is modified, because that could mean that the protocol is effectively unable to operate end-to-end.

To that end, network elements MUST only update the content of datagrams on a given address tuple a few times each monitoring period. Network elements MAY update more often immediately after a change in their throughput advice, to reduce the reaction time from senders.

In addition, some heuristics might be used to detect SCONE-compatible QUIC flows. This includes identification of a QUIC handshake on the flow, the presence of indications (Section 6.1), or other heuristics. If these heuristics indicate a non-QUIC flow, the safest option is for network elements to disable updating of datagrams.

10. Privacy Considerations

The focus of this analysis is the extent to which observing SCONE packets could be used to gain information about endpoints. This might be leaking details of how applications using QUIC operate or leaks of endpoint identity when using additional privacy protection, such as a VPN.

Any network element that can observe the content of that packet can read the rate limit that was applied. Any signal is visible on the path, from the point at which it is applied to the point at which it is consumed at an endpoint. On path elements can also alter the SCONE signal to try trigger specific reactions and gain further knowledge.

In the general case of a client connected to a server through the Internet, we believe that SCONE does not provide much advantage to attackers. The identities of the clients and servers are already visible through their IP addresses. Traffic analysis tools already provide more information than the data rate limits set by SCONE.

There are two avenues of attack that require more analysis:

* that the passive observation of SCONE packets might help identify or distinguish endpoints; and

* that active manipulation of SCONE signals might help reveal the identity of endpoints that are otherwise hidden behind VPNs or proxies.

10.1. Passive Attacks

If only few clients and server pairs negotiate the usage of SCONE, the occasional observation of SCONE packets will "stick out". That observation, could be combined with observation of timing and volume of traffic to help identify the endpoint or categorize the application that they are using.

A variation of this issue occurs if SCONE is widely implemented, but only used in some specific circumstances. In that case, observation of SCONE packets reveals information about the state of the endpoint.

If multiple servers are accessed through the same front facing server, Encrypted Client Hello (ECH) may be used to prevent outside parties to identify which specific server a client is using. However, if only a few of these servers use SCONE, any SCONE packets will help identify which specific server a client is using.

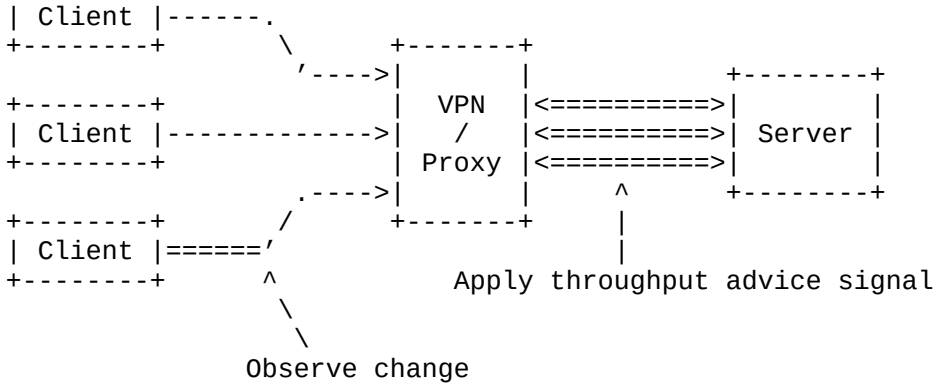
This issue will be mitigated if SCONE becomes widely implemented, and if the usage of SCONE is not limited to the type of applications that make active use of the signal.

QUIC implementations are therefore encouraged to make the feature available unconditionally. Endpoints might send SCONE packets whenever a peer can accept them.

10.2. Active Attacks

Suppose a configuration in which multiple clients use a VPN or proxy service to access the same server. The attacker sees the IP addresses in the packets behind VPN and proxy and also between the users and the VPN, but it does not know which VPN address corresponds to what user address.

Suppose now that the attacker selects a flow on the link between the VPN/proxy and server. The attacker applies throughput advice signals to SCONE packets in that flow. The attacker chooses a bandwidth that is lower than the "natural" bandwidth of the connection. A reduction in the rate of flows between client and VPN/proxy might allow the attacker to link the altered flow to the client.



An attacker that can manipulate SCONE headers can also simulate congestion signals by dropping packets or by setting the ECN CE bit. That will also likely result in changes in the congestion response by the affected client.

A VPN or proxy could defend against this style of attack by removing SCONE (and ECN) signals. There are few reasons to provide per-flow throughput advice signals in that situation. Endpoints might also either disable this feature or ignore any signals when they are aware of the use of a VPN or proxy.

11. IANA Considerations

This document registers a new QUIC version (Section 11.1) and a QUIC transport parameter (Section 11.2).

11.1. SCONE Versions

This document registers the following entries to the "QUIC Versions" registry maintained at https://www.iana.org/assignments/quic (https://www.iana.org/assignments/quic), following the guidance from Section 22.2 of [QUIC].

Value: 0x6f7dc0fd
Status: permanent
Specification: This document
Change Controller: IETF (iesg@ietf.org)
Contact: QUIC Working Group (quic@ietf.org)
Notes: SCONE Protocol - Low Range

Value: 0xef7dc0fd
Status: permanent
Specification: This document
Change Controller: IETF (iesg@ietf.org)
Contact: QUIC Working Group (quic@ietf.org)
Notes: SCONE Protocol - High Range

11.2. scone_supported Transport Parameter

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

Value: 0x219e
Parameter Name: scone_supported
Status: Permanent
Specification: This document
Date: This date
Change Controller: IETF (iesg@ietf.org)
Contact: QUIC Working Group (quic@ietf.org)
Notes: (none)

12. References

12.1. Normative References

[BCP14] Best Current Practice 14,
<https://www.rfc-editor.org/info/bcp14>.
At the time of writing, this BCP comprises the following:

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/info/rfc2119>.

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/info/rfc8174>.

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

[QUIC] 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>.

[QUIC-BIT] Thomson, M., "Greasing the QUIC Bit", RFC 9287, DOI 10.17487/RFC9287, August 2022, <https://www.rfc-editor.org/rfc/rfc9287>.

[QUICv2] Duke, M., "QUIC Version 2", RFC 9369, DOI 10.17487/RFC9369, May 2023, <https://www.rfc-editor.org/rfc/rfc9369>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/rfc/rfc2474>.

12.2. Informative References

[ECN] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/rfc/rfc3168>.

[HLS] Pantos, R., Ed. and W. May, "HTTP Live Streaming", RFC 8216, DOI 10.17487/RFC8216, August 2017, <https://www.rfc-editor.org/rfc/rfc8216>.

[QUIC-MP] Liu, Y., Ma, Y., De Coninck, Q., Bonaventure, O., Huitema, C., and M. Khlewind, "Managing multiple paths for a QUIC connection", Work in Progress, Internet-Draft, draft-ietf-quic-multipath-17, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-quic-multipath-17>.

[RFC4787] Audet, F., Ed. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, DOI 10.17487/RFC4787, January 2007, <https://www.rfc-editor.org/rfc/rfc4787>.

[RFC9330] Briscoe, B., Ed., De Schepper, K., Bagnulo, M., and G. White, "Low Latency, Low Loss, and Scalable Throughput (L4S) Internet Service: Architecture", RFC 9330, DOI 10.17487/RFC9330, January 2023, <https://www.rfc-editor.org/rfc/rfc9330>.

Appendix A. Sliding Window for Rate Limit Monitoring

One way to account for usage is to divide time into multiple smaller spans. For instance, 67 consecutive one second intervals or 134 half second intervals.

The amount of data transmitted in each interval is recorded in a circular buffer, as well as the total amount over 67 seconds. As time passes and new intervals are added, the overall total is reduced by the amount attributed to the oldest interval. The oldest interval is reset to zero and becomes the newest interval.

Sample code for this algorithm is included in Figure 2.

(Artwork only available as PSEUDOCODE: see https://www.ietf.org/archive/id/draft-ietf-scone-protocol-03.html)

Figure 2: Sample code for managing rate limit

Acknowledgments

Jana Iyengar has made significant contributions to the original TRAIN specification that forms the basis for a large part of this document.

Authors' Addresses

Martin Thomson
Mozilla
Email: mt@lowentropy.net

Christian Huitema
Private Octopus Inc.
Email: huitema@huitema.net

Kazuho Oku
Fastly
Email: kazuhooku@gmail.com

Additional contact information:

奥 一穂
Fastly

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

Marcus Ihlar
Ericsson
Email: marcus.ihlar@ericsson.com