ECN Nonces for Stream Control Transmission Protocol (SCTP)
draft-stewart-tsvwg-sctp-nonce-00

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on August 20, 2009.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.
Abstract

This document describes the addition of the ECN-nonce RFC 3540 [RFC3540] to the Stream Control Transmission Protocol (SCTP) RFC 2960 [RFC2960]. The ECN-nonce reduces the vulnerability of ECN senders to misbehaving receivers that conceal congestion signals like ECN marks and packet losses. The ECN-nonce approach is different in SCTP because SCTP uses chunks for extensible protocol features and is selective acknowledge (SACK)-based; this document describes those differences. In particular this document describes (1) protocol extensions in the form of a single new parameter for the INIT/INIT-ACK chunks, and a single bit flag in the SACK chunk, and (2) rules governing the sender and receiver side implementation.

This document outlines a minimum response that an SCTP sender should apply after detecting a misbehaving receiver.
Table of Contents

1. Introduction .................................................. 4
   1.1. Overview of Protocol Extensions .......................... 4
2. Terminology .................................................... 5
3. Protocol Extension to SCTP ................................. 5
   3.1. Nonce-Supported Parameter Definition .................. 5
   3.2. Nonce Sum (NS) flag Definition ........................ 5
4. Nonce-Supported Parameter Negotiation .................... 6
5. ECN-nonce Operation ........................................ 7
   5.1. State Information ........................................ 7
   5.2. Sender Behavior (Transmit) ............................. 7
   5.3. Router Behavior .......................................... 8
   5.4. Receiver Behavior (Receive and Transmit) ............... 8
   5.5. Sender Side (Receive) ................................... 9
   5.6. Re-synchronization of the Nonce Sum .................... 10
6. Sender’s Response ............................................. 10
   6.1. Response to a peer that does not support the ECN-nonce . 10
   6.2. Response to a misbehaving receiver ..................... 11
7. Summary ......................................................... 11
8. Security Considerations ...................................... 11
9. IANA Considerations .......................................... 11
10. Acknowledgements ............................................ 11
11. References .................................................... 12
   11.1. Normative references .................................. 12
   11.2. Informational References ............................... 12
Authors’ Addresses .............................................. 12
1. Introduction

Like TCP, SCTP [RFC2960] supports Explicit Congestion Notification (ECN) [RFC3168], and therefore is exposed to misbehaving receivers that conceal congestion signals. The misbehavior includes the concealment of ECN Echo (ECNE) signals which may cause an SCTP sender to be aggressive and unfair to compliant flows. The ECN-nonce [RFC3540] adds robustness to ECN signaling for TCP. This document analogously applies the ECN-nonce to SCTP. The ECN-nonce can be used to protect against other misbehaviors as well, such as optimistic acknowledgements [paper.savage99], and false Duplicate TSN notifications for more information discussions in [RFC3708] may be helpful.

The reader should be familiar with the ECN-Nonce as described in [RFC3540]. This document describes only the SCTP-specific aspects of the ECN-nonce.

1.1. Overview of Protocol Extensions

This document specifies the following:

1. A single new Nonce-Supported parameter in the INIT/INIT-ACK chunks exchanged during the association establishment to indicate to the peer whether ECN-nonce is supported.

2. A single bit flag in the SACK chunk called the Nonce Sum (NS), and the rules governing the sending, calculating and checking of the nonce sum.

The ECN-nonce does not change the existing ECN protocol. The ECN-nonce uses two bits of the IP header called the ECN Capable Transport (ECT) bits. The sender randomly generates a single bit nonce and encodes it in the ECT codepoints, ECT(0) or ECT(1). To indicate congestion in the network, routers may overwrite the ECT codepoints with a Congestion Experienced (CE) marking. The nonce sum is a cumulative one bit addition of the nonces received so far. The receiver calculates the nonce sum and returns it in the NS flag of the SACK chunk. The sender verifies the value of the NS flag in the SACK chunk. Since all the nonces are needed to calculate the correct nonce sum, an incorrect nonce sum implies that one or more nonces are missing at the receiver. If an incorrect nonce sum is received by the sender without ECNE signals, the sender can infer that the receiver is misbehaving and concealing congestion notifications.

It is beyond the scope of this document to specify a sender’s complete response after detecting a misbehaving receiver. However this document outlines the minimum response that a sender SHOULD
apply to a receiver’s misbehavior.

2. Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, NOT RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in RFC 2119 [RFC2119].

3. Protocol Extension to SCTP

3.1. Nonce-Supported Parameter Definition

Endpoints supporting ECN-nonce SHOULD use the following OPTIONAL parameter to indicate that the ECN-nonce extension is supported.

<table>
<thead>
<tr>
<th>Fixed Parameter</th>
<th>Status</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonce-Supported</td>
<td>OPTIONAL</td>
<td>0x8001</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter Type = 0x8001</th>
<th>Parameter Length = 4</th>
</tr>
</thead>
</table>

Type: 16 bit u_int

0x8001, Indicating the Nonce-Supported parameter.

Length: 16 bit u_int

4, Indicating the size of the parameter.

3.2. Nonce Sum (NS) flag Definition

A single bit flag (NS) in the SACK chunk is defined as follows:
Chunk Flags:

Reserved: 7 bits

Set to 0 on transmit and ignored on receipt.

NS: 1 bit

The Nonce Sum (NS) flag is set by the sender of the SACK chunk with the appropriate calculated value (1 bit cumulative nonce sum). If either peer does not support the ECN-nonce, then the NS flag is always set to 0.

4. Nonce-Supported Parameter Negotiation

An SCTP endpoint supporting the ECN-nonce SHOULD include the Nonce-Supported Parameter in the INIT chunk when initializing the
association to indicate this to the peer.

When a receiver supports the ECN-nonce and detects a Nonce-Supported parameter in the INIT chunk, the receiver SHOULD respond with a Nonce-supported parameter in the INIT-ACK chunk.

When an SCTP endpoint that supports the ECN-nonce receives an INIT chunk without the Nonce-Supported parameter, that SCTP endpoint:

- MAY include the Nonce-Supported parameter in the INIT-ACK chunk.
- SHOULD NOT set the NS flag in the SACK chunk during the lifetime of the association.

An SCTP endpoint that supports the ECN-nonce SHOULD disable ECN (i.e., stop setting the ECT codepoints for the association) if its peer does not support the ECN-nonce. An SCTP endpoint that does not support the ECN-nonce may be denied preferential treatment or otherwise penalized by a peer that supports the ECN-nonce based on that peer’s policy.

5. ECN-nonce Operation

The following sub-sections describe in detail the ECN-nonce operation. This operation includes (i) the local SCTP state to be maintained at each endpoint, (ii) the sender procedures for transmitting nonces and checking of the nonce sum, (iii) intermediate router(s) behavior, and (iv) the receiver procedures for calculating the nonce sum from the received nonces, and returning the nonce sum to the sender.

5.1. State Information

SCTP endpoints should maintain a single bit local state called "current nonce sum". The current nonce sum is a one bit cumulative addition of the nonces. In the beginning of a new SCTP association the current nonce sum SHOULD be initialised to one.

SCTP senders should store a mapping from Transmission Sequence Numbers (TSNs) to nonce values associated with packets.

The following sections describe in detail how these values are maintained and updated.

5.2. Sender Behavior (Transmit)

The sender’s transmit behavior follows Section 3 of RFC 3540 [RFC3540] with the following changes to nonce handling.
The one-bit nonce is placed or encoded only for SCTP packets which carry one or more "new" DATA chunks. Only SCTP packets containing one or more "new" DATA chunks can be marked as ECN-capable (with either ECT(0) or ECT(1) codepoints set), so only these packets can use the one-bit nonce. This is because control chunks are not subject to congestion control, and thus a router's ECN-marking of control chunks would not illicit a congestion response.

The sender maintains a mapping of the TSNs of transmitted DATA chunks with the nonce values placed in the respective packets. When multiple DATA chunks are bundled in the same packet, only one of the TSNs transmitted in that packet is mapped to the nonce placed in the packet. All other TSNs transmitted in the same packet are mapped to a nonce value of zero.

(Implementation Note: An implementation may maintain the mapping of only the single TSN sent in each packet with the corresponding nonce. Said implementation will assume that all TSNs lying between two consecutive TSNs in the data structure are mapped to a nonce value of zero.)

5.3. Router Behavior

For the ECN-nonce to function correctly, routers should behave as specified in RFC 3168 [RFC3168].

5.4. Receiver Behavior (Receive and Transmit)

The receiver updates the value of its current nonce sum on receiving a packet carrying one or more new DATA chunks. Recall that the nonce sum is the cumulative one bit addition of the nonces received so far. Thus on receipt of a packet with one or more new DATA chunks a single bit addition of the current nonce sum and the received nonce is performed. The result is the new value of current nonce sum.

In contrast to RFC 3540 [RFC3540], the current nonce sum is also updated immediately when receiving out of order packets. SCTP is inherently a SACK based protocol, hence an SCTP sender can know exactly which packets had reached the receiver out of order.

When a packet is marked with a Congestion Experienced (CE) signal, the original nonce is unknown to the receiver. The missing nonce value is ignored (or equivalently a value of zero is assumed) when calculating the current nonce sum.

When the receiver sends a SACK chunk, the current nonce sum (the updated value as described above) is placed in the NS flag (defined in Section 3.2) of the SACK chunk.
(Implementation Note: When sending an ECNE chunk and a SACK chunk in response to a CE marked packet that contained one or more DATA chunks, an SCTP endpoint should try to bundle the ECNE chunk and the SACK chunk in the same packet.)

5.5. Sender Side (Receive)

This section describes the sender’s procedure for verifying the nonce sum returned by the receiver.

The nonce sum is verified on the receipt of a SACK chunk which acknowledges new data, either via an advanced Cumulative TSN Ack field or by new Gap Ack Blocks. To verify the nonce sum, the sender SHOULD:

- Calculate the expected nonce sum. This calculation is a single bit addition of the current nonce sum plus all the nonce values corresponding to the new data acknowledged. The nonce values are looked up from the local mapping of TSNs and nonce values maintained at the sender.
- Set the current nonce sum to the value of expected nonce sum.
- Compare the current nonce sum with the NS flag returned in the SACK chunk.

The sender SHOULD disable checking of the nonce sum in the following three scenarios (checking of the nonce sum SHOULD be enabled after re-synchronization as described in Section 5.6):

- On detecting that a packet has been lost (i.e., after receiving four missing reports or after the expiration of the retransmission timer). Nonce checking is suspended because the receiver has admitted to missing packets and an ordinary congestion response is in effect.
- On receiving an ECNE chunk from the receiver. Nonce checking is suspended because the receiver has admitted to a congestion experienced mark and an ordinary congestion response is in effect.
- After sending a FORWARD TSN chunk defined in RFC 3758[RFC3758]. The FORWARD TSN chunk is used by SCTP senders that support the partially reliable extension to move the receiver’s cumulative ack point forward.

If nonce sum checking is enabled, and a SACK chunk is received with an incorrect nonce sum, then several scenarios are possible: (i) the ECNE and SACK chunks were sent in separate packets and the ECNE chunk was dropped by the network, (ii) the ECNE chunk was sent in a separate packet after the SACK chunk (i.e., the ECNE chunk is currently in the network), or (iii) the receiver is misbehaving and concealing Congestion Experienced (CE) signals.
To detect unambiguously that a receiver is misbehaving, the sender waits until new data, sent after having received the incorrect nonce sum, is acknowledged. The sender also suspends checking of the nonce sum during this period. If no ECNE chunk is received till the acknowledgement for the new data arrives, the sender can be certain that the receiver is concealing CE signals and thus misbehaving.

5.6. Re-synchronization of the Nonce Sum

To enable proper checking of the nonce sums, re-synchronization of the sender and receiver current nonce sums is required in three situations.

- Loss of one or more packets in a window of data: Re-synchronization of the sender and receiver current nonce sum occurs when new data sent after the congestion window was reduced is acknowledged via the Cumulative TSN Ack field.
- After having received an ECNE chunk: Re-synchronization of the sender and receiver current nonce sum occurs when new data sent after the Congestion Window Reduced (CWR) chunk is acknowledged via the Cumulative TSN Ack field.
- After sending a FORWARD TSN chunk: Re-synchronization of the sender and receiver current nonce sum occurs when new data sent after the FORWARD TSN chunk is acknowledged via the Cumulative TSN Ack field.

To re-synchronize, the sender simply sets its current nonce sum to the value of NS flag received in the SACK chunk.

6. Sender’s Response

The following discussion describes the response that SHOULD be applied (i) to a peer that does not support the ECN-nonce, or (ii) after having detected a misbehaving receiver.

6.1. Response to a peer that does not support the ECN-nonce

- An SCTP endpoint that supports the ECN-nonce SHOULD disable ECN (i.e., stop setting the ECT codepoints for the association) if its peer does not support the ECN-nonce.
- Optionally, an SCTP sender can further respond by rate limiting the association. Several such responses are possible; this document leaves them as a matter of policy that can be exercised by an endpoint.
6.2. Response to a misbehaving receiver

- After having detected a misbehaving receiver, the SCTP sender SHOULD disable ECN for the rest of the association.
- The minimum penalty to a receiver's misbehavior SHOULD be the equivalent of the response to an ECNE chunk.
- Optionally, an SCTP sender can further respond to a misbehaving receiver by setting the congestion window (cwnd) to one, thus re-probing the available bandwidth. This conservative action eliminates the incentive for a receiver to misbehave. Several such responses are possible; this document leaves them as a matter of policy that can be exercised by an endpoint.

7. Summary

This document applies the ECN-nonce proposal RFC 3540 [RFC3540] to SCTP.

A single new parameter called the Nonce-Supported parameter and a single bit flag in the SACK chunk called the Nonce Sum (NS) have been described along with rules governing the sender and receiver side implementation. This document outlines the minimum response that a sender should apply to a receiver's misbehavior.

8. Security Considerations

This document shares the same security concerns as RFC 3540 [RFC3540].

9. IANA Considerations

A single bit flag in the SACK chunk called the Nonce Sum (NS) is used by this proposal, and must be allocated.

One new parameter type code is defined by this document to be added to SCTP RFC 2960 ('0x8001').

10. Acknowledgements

Paul Amer provided valuable feedback on an early version of this draft.

11. References
11.1. Normative references


11.2. Informational References


Authors’ Addresses

Randall R. Stewart
Researcher
Chapi, SC 29036
USA

Email: randall@lakerest.net