Workgroup: SFC Working Group
Internet-Draft:
draft-ietf-sfc-nsh-ecn-support-13
Published: 15 April 2024
Intended Status: Standards Track
Expires: 17 October 2024

Authors: D. Eastlake   B. Briscoe   Y. Li
Independent   Independent   Huawei Technologies
A. Malis   X. Wei
Malis Consulting   Huawei Technologies

Explicit Congestion Notification (ECN) and Congestion Feedback Using the Network Service Header (NSH) and IPFIX

Abstract

Explicit Congestion Notification (ECN) allows a forwarding element to notify downstream devices of the onset of congestion without having to drop packets. Coupled with a means to feed information about congestion back to upstream nodes, this can improve network efficiency through better congestion control, frequently without packet drops. This document specifies ECN and congestion feedback support within a Service Function Chaining (SFC) enabled domain through use of the Network Service Header (NSH, RFC 8300) and IP Flow Information Export (IPFIX, RFC 7011) protocol.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Explicit Congestion Notification (ECN [RFC3168]) allows a forwarding element to notify downstream nodes of the onset of congestion without having to drop packets. Coupled with a means to feed information about congestion back to upstream nodes, this can improve network efficiency through better congestion control, frequently without packet drops. This document specifies ECN and congestion feedback support within a Service Function Chaining (SFC [RFC7665]) enabled domain through use of the Network Service Header (NSH [RFC8300]) and IP Flow Information Export (IPFIX [RFC7011]) protocol.

This document requires that all ingress and egress nodes of the SFC domain, for the flows to which these techniques are applied, implement ECN. While congestion management will be the most effective if all interior nodes of the SFC enabled domain transited by those flows implement ECN, some benefit is obtained even if some of those nodes do not implement ECN. Congestion at any interior bottleneck where ECN marking is not implemented will be unmanaged.

The following subsections provide background information on NSH, ECN, congestion feedback through IPFIX, and terminology used in this document.

1.1. NSH Background

The Service Function Chaining (SFC [RFC7665]) architecture calls for the encapsulation of traffic within a service function chaining domain with a Network Service Header (NSH [RFC8300]) added by a "Classifier" (ingress node) on entry to the domain with the NSH being removed on exit from the domain at the egress node. The NSH is used to control the path of a packet in the SFC domain.

```
        |
        v
   +----------+
. .|Classifier|. . . . . . . . . . . . . .
.  +----------+                          .
.       |          +----+                .
.       |        --+ SF |     Service    .
.       |       /  +----+     Function   .
.       v    ---              Chaining   .
.    +-----+/       +----+    domain     .
.    | SFF |--------+ SF |               .
.    +-----+\       +----+               .
.       |    ---                         .
.       |       \  +----+                .
.       |        --+ SF |                .
.       v          +----+                .
.    +-----+                 +----+      .
.    | SFF |-----------------+ SF |      .
.    +-----+                 +----+      .
.       |          +----+                .
.       |        --+ SF |                .
.       |       /  +----+                .
.       v    ---                         .
.    +-----+/       +----+               .
.    | SFF |--------+ SF |               .
.    +-----+\       +----+               .
.       |    ---                         .
.       |       \  +----+                .
.       |        --+ SF |                .
.       v          +----+                .
.    +-------+                           .
. . .|Egress|. . . . . . . . . . . . . . .
     +-------+
        |
        v
```

Figure 1: Example SFC Forwarding Nodes Path

Figure 1 shows an SFC enabled domain for the purpose of illustrating the use of the NSH. Traffic passes through a sequence of Service Function Forwarders (SFFs) each of which sends the traffic to one or more Service Functions (SFs). Each SF performs some operation on the traffic, for example firewalling or Network Address Translation (NAT) or load balancing, and then returns the traffic to the SFF from which it was received.

Logically, during the transit of each SFF, the outer transport header that got the packet to the SFF is stripped (see Figure 3), the SFF decides on the next forwarding step, either adding a new outer transport header or, if the SFF is the exit/egress/end, removing the NSH header. The outer transport headers added may be different in different regions of the SFC enabled domain. For example, IP could be used for some SFF-to-SFF communication and MPLS used for other SFF- to-SFF communication.

1.2. ECN Background

Explicit Congestion Notification (ECN [RFC3168]) allows a forwarding element (such as a router or a Service Function Forwarder (SFF) or Service Function (SF)) to notify downstream nodes of the onset of congestion without having to drop packets. This can be used as an element in active queue management (AQM) [RFC7567] to improve network efficiency through better traffic control without packet drops. The forwarding element can explicitly mark some packets in an ECN field instead of dropping the packet. For example, a two-bit field is available for ECN marking in IP headers [RFC3168].

1.3. Tunnel Congestion Feedback Background

Tunnels are widely deployed in various networks including data center networks, enterprise networks, and the public Internet. A tunnel consists of ingress, egress, and a set of intermediate nodes including routers. Tunnel Congestion Feedback (Section 4) is a building block for congestion mitigation methods. It supports feedback of congestion information from an egress node to an ingress node. This document treats paths in the SFC enabled domain as tunnels with the initial Classifier node being the ingress; however, the tunnel congestion feedback facilities specified in this document MAY be used in contexts other than SFC.

Any action by a tunnel ingress to reduce congestion needs to allow sufficient time for the end-to-end congestion control loop to respond first, for instance by the ingress taking a smoothed average of the level of congestion signaled by feedback from the tunnel egress or delaying any action for at least the worst case end-to-end round-trip time (for example, 200 milliseconds). Otherwise, the system could become unstable.

Examples of actions that can be taken by an ingress node when it has knowledge of downstream congestion include those listed below. Details of implementing these traffic control methods, beyond those given here, are outside the scope of this document.

(1)

Traffic throttling (policing), where the downstream traffic flowing out of the ingress node is limited to reduce or eliminate congestion.

(2) Upstream congestion feedback, where the ingress node sends messages indicating congestion upstream to or towards the ultimate traffic source, a function that can throttle traffic generation/transmission.

(3) Traffic re-direction, where the ingress node configures the NSH of some future traffic so that it avoids congested paths. Great care must be taken with this option to avoid (a) significant re-ordering of traffic in flows that it is desirable to keep in order due to end-to-end requirements or due to a stateful SF and (b) oscillation/instability in traffic paths due to alternate congestion of previously idle paths and the idling of previously congested paths. For example, it is preferable to classify traffic into flows of a sufficiently coarse granularity that the flows are long lived and to use a stable path per flow, sending only newly appearing flows on apparently uncongested paths rather than changing the path for any already existing flow.

Figure 2 shows an example path from an original sender to a final receiver passing through a chain of service functions between the ingress and egress of an SFC enabled domain. The path is likely to pass through other network nodes outside the SFC enabled domain (not shown) before entering that domain and after leaving that domain.

Figure 2 shows typical congestion feedback that would be expected from the final receiver to the origin sender, which controls the load the origin sender directs to elements on the path. The figure also shows the congestion feedback from the egress to the ingress of the SFC enabled domain that is described in this document, to control or balance load within that domain.

```
   .:= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = :.
  _||_                 End-to-End Congestion Feedback              ||
  \  /                                                             ||
   \/                                                              ||
   __                Inner Transport Header and Payload            __
  |  | ->- - - - - - - - - - - - - - ->- - - - - -- - - - - - ->- |  |
  |  |                                                            |  |
  |  |       .:= = = = = = = = = = = = = = = = = = = = = =:.      |  |
  |  |      _||_         Tunnel Congestion Feedback       ||      |  |
  |  |      \  /                                          ||      |  |
  |  |       \/                                           ||      |  |
  |  |       __                    NSH                    __      |  |
  |  |      |  |-------------------------->--------------|  |     |  |
  |  |. . . |  |      ___         ___           ___      |  |. . .|  |
  |  |      |  | OT1 |   |  OT4  |   |  . . .  |   | OTn |  |     |  |
  |  |      |  |-->--|SFF|--->---|SFF|         |SFF|-->--|  |     |  |
  |__|      |__|     |___|       |___|         |___|     |__|     |__|
  origin    SFC       | ^         | ^                    SFC     final
  sender   domain  OT2| |OT3   OT6| |OT7                domain   rcvr
           ingress    v |         v |                   egress
                     +---+       +---+                   SFF
                     |SF |       |SF |
                     +---+       +---+
```

Figure 2: Congestion Feedback across an SFC enabled Domain

SFC enabled Domain congestion feedback in Figure 2 is shown within the context of an end-to-end congestion feedback loop. Also shown is the encapsulated layering of NSH headers within a series of outer transport headers (OT1, OT2, ... OTn).

Figure 2 is simplified as there might be multiple egress nodes and some of them may be final receivers for particular packets. (See Section 3.4.)

1.4. Conventions Used in This Document

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.

Acronyms:

AQM - Active Queue Management [RFC7567]

CE - Congestion Experienced [RFC3168]

DDoS -
Distributed Denial of Service

downstream - The direction from ingress to egress

ECN - Explicit Congestion Notification [RFC3168]

ECT - ECN Capable Transport [RFC3168]

IPFIX - IP Flow Information Export [RFC7011]

Not-ECT - Not ECN-Capable Transport [RFC3168]

NSH - Network Service Header [RFC8300]

SF - Service Function [RFC7665]

SFC - Service Function Chaining [RFC7665]

SFF - Service Function Forwarder [RFC7665] - A type of node that forwards based on the NSH.

SPI - Service Path Identifier

TLV - Type Length Value

upstream - The direction from egress to ingress

2. The NSH ECN Field

The NSH is used to encapsulate traffic and control its subsequent path (see Section 2 of [RFC8300]). The NSH also provides for optional metadata inclusion, as shown in Figure 3.

```
+-----------------------------------+
|   Outer Transport Header          |
+-----------------------------------+
|   Network Service Header (NSH)    |
| +------------------------------+  |
| | Base Header                  |  |
| +------------------------------+  |
| | Service Path Header          |  |
| +------------------------------+  |
| | Metadata (Context Header(s)) |  |
| +------------------------------+  |
+-----------------------------------+
| Original Packet / Frame / Payload |
+-----------------------------------+
```

Figure 3: Data Encapsulation with the NSH

This document assigns two currently unused bits (indicated by "U") in the NSH Base Header (Section 2.2 of [RFC8300]) for the purpose of ECN indication as shown in Figure 4.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver|O|U|    TTL    |   Length  |U|U|U|U|MD Type| Next Protocol |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                 ^ ^
                                 | |
                              +-------+
                              |NSH ECN|
                              | field |
                              +-------+
```

Figure 4: Updated NSH Base Header

RFC Editor NOTE: The above figure should be adjusted based on the bits actually assigned by IANA (see Section 6) and this note deleted.

Table 1 shows the meaning of the code points in the NSH ECN field. These have the same meaning as the ECN field code points in the IPv4 or IPv6 header as defined in Section 23.1 of [RFC3168].

Binary	Name	Meaning
00	Not-ECT	Not ECN-Capable Transport
01	ECT(1)	ECN-Capable Transport
10	ECT(0)	ECN-Capable Transport
11	CE	Congestion Experienced

Table 1: ECN Field Code Points

3. ECN Support in the NSH

This section describes the required behavior to support ECN using the NSH. There are two aspects to ECN support:

1. ECN propagation during ingress or egress;

2. ECN marking during congestion at bottlenecks.

While this section covers all combinations of ECN-aware and ECN-unaware, it is expected that in most cases the NSH domain will be uniform so that, if this document is applicable, all SFFs will support ECN; however, some SFs might not support ECN.

ECN Propagation:

The specification of ECN tunneling [RFC6040] explains that an ingress must not propagate ECN support into an encapsulating header unless the egress supports correct onward propagation of the ECN field during decapsulation. We define Compliant ECN Decapsulation here as decapsulation compliant with either [RFC6040] or an earlier compatible equivalent ([RFC4301], or the full functionality mode of [RFC3168]).

The procedures in Section 3.2.1 ensure that each ingress of the transport links within the SFC enabled domain does not propagate ECN support into the encapsulating outer transport header unless the corresponding egress of that link supports Compliant ECN Decapsulation.

Section 3.3 requires that all the egress nodes of the SFC enabled domain that continue to propagate a packet support Compliant ECN Decapsulation in conjunction with tunnel congestion feedback; otherwise the scheme in this document will not work. (An SFC domain may have nodes that terminate packets and thus are logically "egress" nodes but for which further propagation of ECN is meaningless.)

ECN Marking:

At transit nodes the marking behavior specified in Section 3.2.1 is recommended and if not implemented at such transit nodes, there may be unmanaged congestion.

Detection of congestion will be most effective if ECN marking is supported by all potential bottlenecks inside the domain in which NSH is being used to route traffic as well as at the ingress and egress. Nodes that do not support ECN marking, or that support AQM but not ECN, will naturally use drop to relieve congestion. The gap in the end-to-end packet sequence will be detected as congestion by the final receiving endpoint, but not by the NSH egress (see Figure 2).

3.1. At The Ingress

When the ingress/Classifier encapsulates an incoming packet with an NSH, it MUST set the NSH ECN field using the "Normal mode" specified in [RFC6040] (e.g., copied from the incoming IP header).

Then, if the resulting NSH ECN field is Not-ECT, the ingress SHOULD set it to ECT(0). This indicates that, even though the end-to-end transport is not ECN-capable, the egress and ingress of the SFC enabled domain are acting as an ECN-capable transport. This approach supports all known variants of ECN, including the experimental L4S capability [RFC8311] [ecnL4S].

Packets arriving at the ingress might not use IP. If the protocol of arriving packets supports an ECN field similar to IP, for example MPLS [RFC5129], the procedures for IP packets can be used. If arriving packets do not support an ECN field similar to IP, they MUST be treated as if they are Not-ECT IP packets.

Then, as the NSH encapsulated packet is further encapsulated with a transport header, if ECN marking is available for that transport (as it is for IP [RFC3168] and MPLS [RFC5129]), the ECN field of the transport header MUST be set using the "Normal mode" specified in [RFC6040] (i.e., copied from the NSH ECN field).

A summary of these normative steps is given in Table 2.

Incoming Header (also equal to departing Inner Header	Departing NSH and Outer Headers
Not-ECT	ECT(0)
ECT(0)	ECT(0)
ECT(1)	ECT(1)
CE	CE

Table 2: Setting of ECN fields by an Ingress/Classifier

The requirements in this section apply to all ingress nodes for the domain in which an NSH is being used to steer traffic.

3.2. At Transit Nodes

This section describes the behavior at nodes that forward based on the NSH such as SFF and other forwarding nodes such as IP routers. Figure 5 shows a packet on the wire between forwarding nodes.

```
+-----------------+
|   Outer Header  |
+-----------------+
|       NSH       |
+-----------------+
|   Inner Header  |
+-----------------+
|     Payload     |
+-----------------+
```

Figure 5: Packet in Transit

There can be nodes implementing firewall, DDoS, or similar functions that conditionally discard packets. When they do discard a packet, they are an egress node (see Section 3.3), not a transit node.

3.2.1. At NSH Transit Nodes

When a packet is received at an NSH based forwarding node such as an SFF, say N1, the outer transport encapsulation is removed and its ECN marking SHOULD be combined into the NSH ECN marking as specified in [RFC6040]. If this is not done, any congestion encountered at non-NSH transit nodes between N1 and the previous upstream NSH based forwarding node will be lost and not transmitted downstream.

The NSH forwarding node SHOULD use a recognized AQM algorithm [RFC7567] to detect congestion. If the NSH ECN field indicates ECT, it will probabilistically set the NSH ECN field to the Congestion Experienced (CE) value or, in cases of extreme congestion, drop the packet.

When the NSH encapsulated packet is further encapsulated for transmission to the next SFF or SF, ECN marking behavior depends on whether or not the node that will decapsulate the outer header supports Compliant ECN Decapsulation (see Section 3). If it does, then the encapsulating node propagates the NSH ECN field to this outer encapsulation using the "Normal Mode" of ECN encapsulation [RFC6040] (the ECN field is copied). If it does not, then the encapsulating node MUST clear ECN in the outer encapsulation to non-ECT (the "Compatibility Mode" of [RFC6040]).

3.2.2. At an SF/Proxy

If the SF is NSH and ECN-aware, the processing is essentially the same at the SF as at an SFF as discussed in Section 3.2.1 (except in the case where the SF terminates the packets path).

If the SF is NSH-aware but ECN-unaware, then the SFF transmitting the packet to the SF will use Compatibility Mode. Congestion encountered in the SFF to SF and SF to SFF paths or internal to the SF will be unmanaged.

If the SF is not NSH-aware, then an NSH proxy will be between the SFF and the SF to avoid exposure of the NSH-ignorant SF to NSHs as shown in Figure 6. This is described in Section 4.6 of [RFC7665]. The SF and proxy together look to the SFF like an NSH-aware SF. The behavior at the proxy and SF in this case is as below:

If such a proxy is not ECN-aware, then congestion in the entire path from SFF to proxy to SF back to proxy to SFF will be unmanaged.

```
                  |
                  v
             +----------+                  +---------+
             |          |     +--------+     |   NSH   |
             |   SFF    +---->|  NSH   +---->|un-aware |
             |(Service  |     | aware  |     |   SF    |
             | Function |<----+ proxy  |<----+(Service |
             |Forwarder)|     +--------+     |Function)|
             +----------+                  +---------+
                  |
                  v
```

Figure 6: Proxy for NSH Un-aware SFF

If the proxy is ECN-aware, the proxy uses an AQM to indicate congestion within the proxy in the NSH that it returns to the SFF. The outer header used for the proxy-to-SF path uses Normal Mode. The outer header used for the proxy-to-SFF path uses Normal Mode based copying of the NSH ECN field to the outer header. Thus congestion in the proxy will be managed.

Congestion in the SF will be managed only if the SF is ECN-aware and implements an AQM.

3.2.3. At Other Forwarding Nodes

Other forwarding nodes, that is non-NSH forwarding nodes between NSH forwarding nodes, such as IP or label switched routers, bridges, or other devices, might also contain potential bottlenecks. If so, they SHOULD implement an AQM algorithm to update the ECN marking in the outer transport header as specified in [RFC3168].

3.3. At Exit/Egress/End

At an SFC enabled domain egress node, first any actions are taken based on Congestion Experienced or other values of ECN marking, such as accumulating statistics to send back to the ingress (see Section 4) or for other uses.

There can be nodes implementing firewall, DDoS, or similar functions that then discard the packet. If the packet is so discarded, no further actions are needed.

If the packet is to be propagated and is carried inside the NSH as encapsulated IP, then when the NSH is removed the NSH ECN field MUST be combined with the IP ECN field as specified in Table 3 that was extracted from Section 3.2 of [RFC6040]. This requirement applies to all egress nodes for the domain in which an NSH is being used to route traffic.

Arriving Inner Header	Arriving Outer Header			
	Not-ECT	ECT(0)	ECT(1)	CE
Not-ECT	Not-ECT	Not-ECT	Not-ECT	<drop>
ECT(0)	ECT(0)	ECT(0)	ECT(0)	CE
ECT(0)	ECT(0)	ECT(0)	ECT(0)	CE
CE	CE	CE	CE	CE

Table 3: Exit ECN Fields Merger (Source [RFC6040])

All the egress nodes of the SFC enabled domain that can propagate NSH encapsulated packets MUST support Compliant ECN Decapsulation as specified in this section. If this is not the case, the scheme described in this document will not work.

3.4. Congestion Statistics and More Complex Cases

The SFC specification permits an SF to absorb packets and to generate new packets as well as simply processing and returning the packets it receives to an SFF. Such actions might appear to be packet loss due to congestion or might mask the loss of packets by generating additional packets.

The closer a particular application of SFC is to a simple tunnel with a single ingress and egress, the simpler it is to accurately use the techniques in this document. Where there is a single ingress but multiple egress nodes (where a node that discards a packet counts as an egress) these techniques can still work well if all egress nodes feedback congestion information to that ingress. Multiple ingress nodes are a substantial complication, but similar techniques may still work in some cases if multiple physical ingress nodes can coordinate to act as one logical ingress node; methods for such coordination are beyond the scope of this document. Use of the techniques in this document for a flow with multiple egress and uncoordinated ingress nodes is NOT RECOMMENDED, although there might be some cases where these techniques could be elements in some sort of beneficial scheme; such schemes are beyond the scope of this document.

The tunnel congestion feedback approach (Section 4) can detect congestions in several ways. One way detects traffic loss by counting payload packets and bytes in at the ingress and counting them out at the egress. This does not work unless nodes conserve the number of payload packets and/or bytes. Therefore, it will not be possible to accurately detect packet loss using this technique if traffic volume, as measured by the metric in use (packets or bytes), is not conserved by the service function chain processing that traffic.

Nonetheless, if a bottleneck supports ECN marking, it will be possible to detect the high level of CE markings that are associated with congestion at that bottleneck by looking at the ratio of CE-marked to non-CE-marked packets. However, it will not be possible to detect any congestion based on ECN marking, whether slight or severe, if it occurs at a bottleneck that does not support ECN marking.

4. Tunnel Congestion Feedback Support

The collection and storage of congestion information at an egress can be useful for later analysis and MAY be used without the feedback mechanisms specified in this Section. However, if congestion information is not fed back to a point which can act to reduce congestion, it will not be useful in real time. Such congestion feedback to the ingress enables the ingress to take actions such as those listed in Section 1.3.

IP Flow Information Export (IPFIX [RFC7011]) provides a standard for communicating traffic flow statistics. As extended by this document, IPFIX messages from the egress to the ingress are used to communicate the extent of congestion between an ingress and egress based on ECN marking in the NSH and traffic statistics. Each egress

MUST be able to identify the relevant ingress for a packet based on information in the packed such as the SPI or the Ingress Network Node Information Context Header [RFC9263].

4.1. Congestion Level Measurements

The congestion level measurements are based on ECN marking in the NSH and packet drop. In particular, congestion information includes at least one of the following:

- cumulative byte counts of packets with each type of outer/inner header ECN marking combination,
- the ratio of CE-marked packets to all packets, and
- the ratio of dropped packets to all packets.

All IPFIX messages are time stamped [RFC7011]. So, for example, it is possible to compute rates of packets or packets with various ECN labeling from two IPFIX messages that have cumulative counts and time stamps. An earlier count and time can be deducted from a later count and time to give the time interval and count during that interval.

If the congestion level is low enough, the packets are marked as CE instead of being dropped, and then the congestion level can be calculated according to the ratio of CE-marked packets. If the congestion level is so high that ECT packets will be dropped, then the packet loss ratio can be calculated by comparing total packets entering ingress and total packets arriving at egress over the same span of packets. Note that a node that discards packets for firewall, DDoS, or similar reasons counts as an egress. If packet loss, other than such deliberate discard, is detected, then it can be assumed that severe congestion has occurred.

Faked ECN-Capable Transport (ECT) is used at the ingress to defer packet loss to the egress. The basic idea of faked ECT is that, when encapsulating packets, the ingress first marks the tunnel outer header according to [RFC6040], and then remarks the outer header of Not-ECT packets as ECT. (ECT(0) and ECT(1) are treated as the same.) In this case, the NSH is treated as the tunnel outer header because it will be present for the entire SFC enabled domain transit while transport headers may change. Thus, as transmitted by the ingress node, there will be one of three combinations of outer header ECN field and inner header ECN field as follows: CE|CE, ECT|N-ECT, and ECT|ECT (in the format of outer-ECN|inner-ECN); when decapsulating packets at the egress, [RFC6040] defined decapsulation behavior is used, and according to [RFC6040], the packets marked as CE|N-ECT will be dropped. Faked-ECT is used to shift some drops to the egress in order to allow the egress to calculate the CE-marked packet counts and ratio more precisely.

The ingress encapsulates packets and marks their outer header according to faked ECT as described above. The ingress cumulatively counts packet bytes for three types of ECN combination (CE|CE, ECT|N- ECT, and ECT|ECT) and then the ingress regularly sends cumulative byte counts message of each type of ECN combination to the egress.

When each message arrives at the egress, the following two steps occur: (1) the egress calculates the ratio of CE-marked packets; (2) the egress cumulatively counts packet bytes coming from the ingress and adds its own bytes counts of each type of ECN combination (CE|CE, ECT|N-ECT, CE|N-ECT, CE|ECT, and ECT|ECT) to the message for the ingress to calculate packet loss. The egress feeds back the CE-marked packet ratio, packet loss ratio, byte counts information, and the like to the ingress as requested for evaluating congestion level in the tunnel.

The egress calculates the CE-marked packet ratio by counting packets with different ECN markings. The CE-marked packet ratio can be used as an indication of tunnel load level. For example, the tunnelEcnCEMarkedRatio field (specified below) indicates the fraction of traffic that has been marked in the ECN field of the NSH as Congestion Experienced (CE). It is assumed that nodes between the ingress and egress will not drop packets biased towards certain ECN codepoints, so calculating of CE-marked packet ratio is not affected by packet drop.

The calculation of the fraction of packets dropped is by comparing the traffic volumes between ingress and egress.

In the case of multiple egresses, the ingress can combine their reports. Statistics of number of packets or bytes can simply be added. Statistics of percentage or ratio of particular ECN marking can be averaged with reports from different egresses weighted by the number of packets processed by that egress.

The statistics can be at the granularity of all traffic from the ingress to the egress to learn about the overall congestion status of the path between the ingress and the egress or at the granularity of individual customer's traffic or a specific set of flows to learn about their congestion contribution.

4.2. Congestion Information Delivery

As described above, the tunnel ingress sends a message containing cumulative byte counts of packets of each type of ECN marking to the tunnel egress, and the tunnel egress feeds back messages to the ingress with at least one of the following: cumulative byte counts of packets of each type of ECN combination, the ratio of CE-marked packets to all packets, and/or the ratio of dropped packets to all packets. It is possible for these messages to contribute to congestion. This section specifies how the messages are conveyed.

IPFIX recommends, but does not require, use of SCTP [RFC9260] in partial reliability mode [RFC3758] for the transport of its messages. This mode allows loss of some packets, which is tolerable because IPFIX communicates cumulative statistics. IPFIX over SCTP over IP SHOULD be used directly where there is IP connectivity between the ingress and egress; however, there might be different transport protocols or address spaces used in different regions of an SFC enabled domain that block such direct IP connectivity. The NSH provides the general method of routing traffic within an SFC enabled domain so the encapsulation of the required IPFIX traffic in NSH MUST be implemented and, when IP connectivity is not available, IPFIX over NSH, as specified in Section 4.4, SHOULD be used along with configuration of appropriate SFC paths for the IPFIX over NSH traffic. Other methods MAY be used in particular SFC domains which support them, such as IPFIX over MPLS.

IPFIX messages could travel along the same path as network data traffic. In any case, an IPFIX message packet may get lost in case of network congestion. Even though the missing information could be recovered because of the use of cumulative counts, IPFIX messages SHOULD be transmitted at a higher priority than users' traffic flows to improve the promptness of congestion information feedback.

The ingress node can do congestion management at different granularity which means both the overall aggregate congestion level and congestion level contributed by certain traffic flows could be measured for different congestion management purposes. For example, if the ingress only wants to limit congestion volume caused by certain traffic flows, such as UDP-based traffic, then congestion volume for that traffic can be fed back; or if the ingress is doing overall congestion management, the aggregated congestion volume can be fed back.

When sending IPFIX messages from ingress to egress, the ingress acts as IPFIX exporter and the egress acts as IPFIX collector. When feeding back congestion level information from egress to ingress, the egress acts as IPFIX exporter and ingress acts as IPFIX collector.

The combination of congestion level measurement and congestion information delivery procedures are as following:

- The ingress node determines the IPFIX template record to be used. The template record can be pre-configured or determined at runtime, the content of the template record will be determined according to the granularity of congestion management; if the ingress wants to limit congestion volume contributed by specific traffic flows then the elements such as source IP address, destination IP address, flow ID, and CE-marked packet volume of the flows, etc., will be included in the template record.

- Metering at the ingress measures traffic volume according to the template record chosen and then the measurement records are sent to the egress.

- Metering on the egress measures congestion level information according to template record which, in simple cases, SHOULD be the same as the template record sent by the ingress (see Section 3.4).

- The egress sends its measurement records together with the measurement records of the ingress back to the ingress.

4.3. IPFIX Extensions

This section specifies the new IPFIX Information Elements needed. It conforms to [RFC7013].

4.3.1. nshServicePathID

In order to identify SFC flows, so that congestion can be measured and reported at that granularity, it is necessary for IPFIX to be able to classify traffic based on the Service Path Identifier (SPI) field of the NSH [RFC8300]. Thus, an NSH Service Path Identifier (nshServicePathID) IPFIX Information Element [RFC7012] is specified.

Name: nshServicePathID

Description: Network Service Header [RFC8300] Service Path Identifier. This is a 24-bit value which is left justified in the Information Element. The low order byte MUST be sent as zero and ignored on receipt.

Abstract Data Type: unsigned32

Data Type Semantics: identifier

ElementId: TBD0

Status: current

4.3.2. tunnelEcnCeCeByteTotalCount

Description: The total number of bytes of incoming packets with the CE|CE ECN marking combination at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Abstract Data Type: unsigned64

Data Type Semantics: totalCounter

ElementId: TBD1

Statues: current

Units: bytes

4.3.3. tunnelEcnEctNectBytetTotalCount

Description: The total number of bytes of incoming packets with the ECT|N-ECT ECN marking combination (ECT(0) and ECT(1) are treated the same as each other) at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Abstract Data Type: unsigned64

Data Type Semantics: totalCounter

ElementId: TBD2

Statues: current

Units: bytes

4.3.4. tunnelEcnCeNectByteTotalCount

Description: The total number of bytes of incoming packets with the CE|N-ECT ECN marking combination at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Abstract Data Type: unsigned64

Data Type Semantics: totalCounter

ElementId: TBD3

Statues: current

Units: bytes

4.3.5. tunnelEcnCeEctByteTotalCount

Description: The total number of bytes of incoming packets with the CE|ECT ECN marking combination (ECT(0) and ECT(1) are treated the same as each other) at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Abstract Data Type: unsigned64

Data Type Semantics: totalCounter

ElementId: TBD4

Statues: current

Units: bytes

4.3.6. tunnelEcnEctEctByteTotalCount

Description: The total number of bytes of incoming packets with the ECT|ECT ECN marking combination (ECT(0) and ECT(1) are treated the same as each other) at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Abstract Data Type: unsigned64

Data Type Semantics: totalCounter

ElementId: TBD5

Statues: current

Units: bytes

4.3.7. tunnelEcnCEMarkedRatio

Description: The ratio of packets that are CE-marked to packets that are not CE-marked at the Observation Point.

Abstract Data Type: float32

ElementId: TBD6

Statues: current

4.4. IPFIX over NSH

Encapsulating IPFIX messages with an NSH can be an effective method for transporting such messages within an SFC enabled domain. This is particularly the case if different outer transport protocols are used in different parts of such a domain, for example IP in one part and MPLS in another part.

This is accomplished by setting the Next Protocol field in the NSH Base Header [RFC8300] to the value TBD7 and placing the IPFIX message immediately after the NSH (including after any NSH Metadata). See Figure 7.

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |Ver|O|U|    TTL    |   Length  |ECN|U|U|MD Type|Next Proto=TBD7|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |          Service Path Identifier (SPI)        | Service Index |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 /   Optional MD (Metadata)                                      /
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   IPFIX Message                                               |
```

Figure 7: IPFIX over NSH

5. Example of Use

This section provides an example of the solution described in this document.

First, IPFIX template records are exchanged between ingress and egress to negotiate the format of the data records to be exchanged. The example here is to measure the congestion level for the overall tunnel caused by all the traffic. After the negotiation is finished, the ingress sends in-band messages to the egress containing the number of each kind of ECN-marked packets (i.e., CE|CE, ECT|N-ECT and ECT|ECT) received before it sent the IPFIX message.

After the egress receives the IPFIX message, the egress calculates the CE-marked packet ratio and counts the number of different kinds of ECN-marking packets received before it received that message. Then the egress sends a feedback IPFIX message containing the counts together with the information in the ingress's message back to the ingress.

Figure 8 to Figure 11 below illustrate the procedure between ingress and egress.

```
+---------------------------------+----------------------+
|Set ID=2                               Length=40        |
|---------------------------------|----------------------|
|Template ID=256                        Field Count=8    |
|---------------------------------|----------------------|
|tunnelEcnCeCeByteTotalCount            Field Length=8   |
|---------------------------------|----------------------|
|tunnelEcnEctNectByteTotalCount         Field Length=8   |
|---------------------------------|----------------------|
|tunnelEcnEctEctByteTotalCount          Field Length=8   |
|---------------------------------|----------------------|
|tunnelEcnCeNectByteTotalCount          Field Length=8   |
|---------------------------------|----------------------|
|tunnelEcnCeEctByteTotalCount           Field Length=8   |
+---------------------------------|----------------------+
|tunnelEcnCEMarkedRatio                 Field Length=4   |
+---------------------------------+----------------------+
```

Figure 8: Template Record Sent from Egress to Ingress

```
+---------------------------------+----------------------+
|Set ID=2                               Length=28        |
|---------------------------------|----------------------|
|Template ID=257                        Field Count=3    |
|---------------------------------|----------------------|
|tunnelEcnCeCeByteTotalCount            Field Length=8   |
|---------------------------------|----------------------|
|tunnelEcnEctNectByteTotalCount         Field Length=8   |
|---------------------------------|----------------------|
|tunnelEcnEctEctByteTotalCount          Field Length=8   |
|---------------------------------+----------------------|
```

Figure 9: Template Record Sent from Ingress to Egress

```
+-------+                                        +-------+
|       |  +-+ +-+ +-+  +-+ +-+ +-+  +-+         |       |
|       |  |P| |P| |M|  |P| |P| |P|  |M|         |       |
|       |  +-+ +-+ +-+  +-+ +-+ +-+  +-+         |       |
|       |--------------------------------------->|       |
|       |                                        |       |
|ingress|                                        |egress |
|       |            +-+             +-+         |       |
|       |            |M|             |M|         |       |
|       |            +-+             +-+         |       |
|       |<---------------------------------------|       |
|       |                                        |       |
+-------+                                        +-------+

            +-+
            |M| : IPFIX Message Packet
            +-+

            +-+
            |P| : User Data Packet
            +-+
```

Figure 10: Traffic Flow Between Ingress and Egress

```
                   SetID=257, Length=28
  +-------+             A1                    +-------+
  |       |             B1                    |       |
  |       |             C1                    |       |
  |       |  ------------------------------>  |       |
  |       |                                   |       |
  |       |                                   |       |
  |       |        SetID=256, Length=72       |       |
  |       |             A1                    |       |
  |       |             B1                    |       |
  |ingress|             C1                    |egress |
  |       |             A2                    |       |
  |       |             B2                    |       |
  |       |             C2                    |       |
  |       |             D                     |       |
  |       |             E                     |       |
  |       |             R                     |       |
  |       |    <----------------------------  |       |
  |       |                                   |       |
   +-------+                                   +-------+
```

Figure 11: Traffic Flow Between Ingress and Egress

The following provides an example of how the tunnel congestion level can be calculated (see Figure 11):

The congestion Level could be divided into two categories: (1) slight congestion (no packets dropped); (2) serious congestion (packets are being dropped).

For slight congestion, the congestion level is indicated by the ratio of CE-marked packets:

```
R = ce_marked_ratio = ce-marked / total_egress ;
```

For serious congestion, the congestion level is indicated as the volume of traffic loss:

```
total_ingress = (A1 + B1 + C1)

total_egress = (A2 + B2 + C2 + D + E)

volume_loss = (total_ingress - total_egress)
```

6. IANA Considerations

The following subsections provide IANA assignment considerations.

6.1. SFC NSH Header ECN Bits

IANA is requested to assign two contiguous bits in the NSH Base Header Bits registry for ECN (bits 16 and 17 suggested) and note this assignment as follows:

Bit	Description	Reference
tbd(16-17)	NSH ECN	[this document]

Table 4

6.2. SFC NSH Next Protocol Value

IANA is requested to assign a next protocol value in the NSH Next Protocol Registry, as follows:

Next Protocol	Description	Reference
TBD7	IPFIX	[this document]

Table 5

6.3. IPFIX Information Element IDs

IANA is requested to assign seven IPFIX Information Element IDs as follows:

ElementID: TBD0

Name: nshServicePathID

Data Type: unsigned32

Data Type Semantics: identifier

Status: current

Description: The Network Service Header [RFC8300] Service Path Identifier.

ElementID: TBD1

Name: tunnelEcnCeCePacketTotalCount

Data Type: unsigned64

Data Type Semantics: totalCounter

Status: current

Description: The total number of bytes of incoming packets with the CE|CE ECN marking combination at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Units: octets

ElementID: TBD2

Name: tunnelEcnEctNectPacketTotalCount

Data Type: unsigned64

Data Type Semantics: totalCounter

Status: current

Description: The total number of bytes of incoming packets with the ECT|N-ECT ECN marking combination at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Units: octets

ElementID: TBD3

Name: tunnelEcnCeNectPacketTotalCount

Data Type: unsigned64

Data Type Semantics: totalCounter

Status: current

Description: The total number of bytes of incoming packets with the CE|N-ECT ECN marking combination at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Units: octets

ElementID: TBD4

Name: tunnelEcnCeEctPacketTotalCount

Data Type: unsigned64

Data Type Semantics: totalCounter

Status: current

Description: The total number of bytes of incoming packets with the CE|ECT ECN marking combination at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Units: octets

ElementID: TBD5

Name: tunnelEcnEctEctPacketTotalCount

Data Type: unsigned64

Data Type Semantics: totalCounter

Status: current

Description: The total number of bytes of incoming packets with the CE|ECT(0) ECN marking combination at the Observation Point since the Metering Process (re-)initialization for this Observation Point.

Units: octets

ElementID: TBD6

Name: tunnelEcnCEMarkedRatio

Data Type: float32

Status: current

Description: The ratio of CE-marked Packet at the Observation Point.

6.4. Security Considerations

For general NSH security considerations, see [RFC8300].

For security considerations concerning ECN signaling tampering, see [RFC3168]. For security considerations concerning ECN and encapsulation, see [RFC6040].

For general IPFIX security considerations, see [RFC7011]. If deployed in an untrusted environment, the signaling traffic between ingress and egress can be protected utilizing the security mechanisms provided by IPFIX (see Section 11 in [RFC7011]). The tunnel endpoints (the ingress and egress for an SFC enabled domain) are assumed to be in the same administrative domain, so they will trust each other.

The solution in this document does not introduce any greater potential to invade privacy than would have been available without the solution.

7. 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/info/rfc2119>.

[RFC3168] 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/info/rfc3168>.

[RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, DOI 10.17487/RFC3758, May 2004, <https://www.rfc-editor.org/info/rfc3758>.

[RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", RFC 5129, DOI 10.17487/RFC5129, January 2008, <https://www.rfc-editor.org/info/rfc5129>.

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, DOI 10.17487/RFC6040, November 2010, <https://www.rfc-editor.org/info/rfc6040>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/info/rfc7011>.

[RFC7013] Trammell, B. and B. Claise, "Guidelines for Authors and Reviewers of IP Flow Information Export (IPFIX) Information Elements", BCP 184, RFC 7013, DOI 10.17487/RFC7013, September 2013, <https://www.rfc-editor.org/info/rfc7013>.

[RFC7567] Baker, F., Ed. and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <https://www.rfc-editor.org/info/rfc7567>.

[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/info/rfc8174>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

8. Informative References

[ecnL4S] De Schepper, K. and B. Briscoe, "Identifying Modified Explicit Congestion Notification (ECN) Semantics for Ultra-Low Queuing Delay (L4S)", work in Progress, <draft-ietf-tsvwg-ecn-l4s-id>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC7012] Claise, B., Ed. and B. Trammell, Ed., "Information Model for IP Flow Information Export (IPFIX)", RFC 7012, DOI 10.17487/RFC7012, September 2013, <https://www.rfc-editor.org/info/rfc7012>.

[RFC7665]
Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC8311] Black, D., "Relaxing Restrictions on Explicit Congestion Notification (ECN) Experimentation", RFC 8311, DOI 10.17487/RFC8311, January 2018, <https://www.rfc-editor.org/info/rfc8311>.

[RFC9260] Stewart, R., Tüxen, M., and K. Nielsen, "Stream Control Transmission Protocol", RFC 9260, DOI 10.17487/RFC9260, June 2022, <https://www.rfc-editor.org/info/rfc9260>.

[RFC9263] Wei, Y., Ed., Elzur, U., Majee, S., Pignataro, C., and D. Eastlake 3rd, "Network Service Header (NSH) Metadata Type 2 Variable-Length Context Headers", RFC 9263, DOI 10.17487/RFC9263, August 2022, <https://www.rfc-editor.org/info/rfc9263>.

Acknowledgements

Most of the material on Tunnel Congestion Feedback was originally in draft-ietf-tsvwg-tunnel-congestion-feedback. After discussion with the authors of that draft, the authors of this draft, and the Chairs of the TSVWG and SFC Working Groups, the Tunnel Congestion Feedback draft was merged into this draft.

The authors wish to thank the following for their comments, suggestions, and reviews:

David Black, Mohamed Boucadair, Sami Boutros, Anthony Chan, Lingli Deng, Liang Geng, Joel Halpern, Jake Holland, John Kaippallimalil, Tal Mizrahi, Vincent Roca, Lei Zhu.

Authors' Addresses

Donald E. Eastlake, 3rd
Independent
2386 Panoramic Circle
Apopka, Florida 32703
United States of America

Phone: +1-508-333-2270
Email: d3e3e3@gmail.com

Bob Briscoe
Independent
United Kingdom

Email: ietf@bobbriscoe.net
URI: http://bobbriscoe.net/

Yizhou Li
Huawei Technologies
101 Software Avenue
Nanjing
Jiangsu, 210012
China

Phone: +86-25-56624584
Email: zhuangshunwan@huawei.com

Andrew G. Malis
Malis Consulting
United States of America

Email: agmalis@gmail.com

Xinpeng Wei
Huawei Technologies
Beiqing Rd. Z-park No.156, Haidian District
Beijing
100095
China

Email: weixinpeng@huawei.com