OPSAWG
Internet-Draft
Intended status: Informational
Expires: 22 March 2026

J. Evans
O. Pylypenko
K. Cheaito
Amazon

18 September 2025

Information Element for Flow Discard Classification
draft-evans-opsawg-ipfix-discard-class-ie-02

Abstract

This document defines an IPFIX Information Element for classifying flow-level discards that aligns with the information model defined in [I-D.ietf-opsawg-discardmodel]. The flowDiscardClass Information Element provides consistent classification of packet discards across IPFIX implementations, enabling correlation between device, interface and control-plane discards and the impacted flows.

Table of Contents

1. Introduction

For network operators, understanding both where and why packet loss occurs within a network is essential for effective operation. While certain types of packet loss, such as policy-based discards, are intentional and part of normal network operation, unintended packet loss can impact customer services. To automate network operations, operators must be able to detect customer-impacting packet loss, determine its root cause, and apply appropriate mitigation actions.

[I-D.ietf-opsawg-discardmodel] addresses this need by defining an information model that provides precise classification of packet loss, enabling accurate automated mitigation. While its YANG data model implementation provides device, interface and control-plane discards, effective automated triage often requires understanding which specific flows are impacted. For example, when mitigating congestion, operators may need to identify and trace the sources of elephant flows. This requires the ability to correlate device and interface-level discard classes with the specific flows being dropped.

Currently, [RFC7270] defines the forwardingStatus Information Element for reporting packet forwarding outcomes in IPFIX, including various reasons for packet drops. The defined drop reason codes lack the granularity and clarity needed for automated root cause analysis and impact mitigation, however. For instance, the "For us" reason code provides insufficient context to determine appropriate mitigation actions.

This document addresses these limitations by introducing a new Information Element, flowDiscardClass, to provide a consistent classification scheme for packet discards across IPFIX implementations. This new element aligns with the classification scheme defined in [I-D.ietf-opsawg-discardmodel] and enables:

1. Precise detection of unintended packet loss through clear distinction between intended and unintended discards

2. Accurate root cause analysis through detailed classification of discard reasons

3. Automated selection of mitigation actions based on discard type, rate, and duration

4. Consistent reporting across vendor implementations in both YANG and IPFIX data models

By providing this mapping between YANG and IPFIX implementations, this document enables operators to correlate device-level statistics with flow-level impacts, facilitating more effective automated network operations.

2. Terminology

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.

This document makes use of the following terms:

Packet discard: It accounts for any instance where a packet is dropped by a device, regardless of whether the discard was intentional or unintentional.

Intended packet discards (Intended discards, for short): Are packets dropped due to deliberate network policies or configurations designed to enforce security or Quality of Service (QoS). For example, packets dropped because they match an Access Control List (ACL) denying certain traffic types.

Unintended packet discards (Unintended discards, for short): Are packets that were dropped, which the network operator otherwise intended to deliver, i.e. which indicates an error state. There are many possible reasons for unintended packet loss, including: erroring links may corrupt packets in transit; incorrect routing tables may result in packets being dropped because they do not match a valid route; configuration errors may result in a valid packet incorrectly matching an ACL and being dropped.

Device discard counters do not by themselves establish operator intent. Discards reported under policy (e.g., ACL/policer) indicate only that traffic matched a configured rule; such discards may still be unintended if the configuration is in error. Determining intent for policy discards requires external context (e.g., configuration validation and change history) which is out of scope for this specification.

3. Information Element

This Information Element has been specified in accordance with the guidelines in [RFC7013].

3.1. Design Rationale

The mapping between [I-D.ietf-opsawg-discardmodel] and the IPFIX flowDiscardClass Information Element follows these principles, maintaining consistency with the YANG model while allowing self-contained decoding from a single IE:

1. Scope. The flowDiscardClass Information Element is specifically for reporting flow-level discard reasons, and therefore only represents the flow subtree from [I-D.ietf-opsawg-discardmodel]. The component is implicitly "flow" and the type is implicitly "discards"; interface, device, and control-plane components are out of scope for this IE.

2. Hierarchy preserved. The enumeration mirrors the model: both leaves (specific reasons) and structural aggregates are assigned values so collectors can perform coarse or fine roll-ups. For L3, structural aggregates include address-family and cast (v4/v6, unicast/multicast/broadcast).

3. Self-contained decoding. The value alone carries the discard class. Exporters and collectors can still use other IEs (e.g., flowDirection, ipVersion, addresses, ipDiffServCodePoint) for correlation, but they are not required to decode the class.

4. Specificity preference. The scheme encourages reporting the most-specific known class when available; aggregate values provide a fallback when only a broader category is known.

5. Implementation-friendly ordering. Codes are assigned in preorder traversal (where each parent is numbered before its children) to reflect the model's hierarchy and simplify range/roll-up handling in implementations.

3.2. flowDiscardClass Definition

Name: flowDiscardClass

Description: Classifies the reason a packet was discarded in a flow, using the hierarchical classification scheme defined in [I-D.ietf-opsawg-discardmodel].

Abstract Data Type: unsigned8

Data Type Semantics: identifier

Units: none

Range: 0..38 (values from Table 1; other values are unassigned and MUST be treated as unknown)

Reversibility: reversible (value does not change under flow reversal as per [RFC5103])

Status: current

ElementId: TBD

References: [I-D.ietf-opsawg-discardmodel]

3.3. flowDiscardClass Values

Table 1 defines the values for the flowDiscardClass Information Element mapped from the corresponding [I-D.ietf-opsawg-discardmodel] Discard Class. Codes are assigned in preorder traversal to reflect the hierarchy.

The code points for flowDiscardClass are maintained by IANA in the "flowDiscardClass Values" subregistry of the IPFIX registry. Future additions or changes are managed via Expert Review as described in Section 5.

Discard Class	flowDiscardClass Value
l2	0
l3	1
l3/v4	2
l3/v4/unicast	3
l3/v4/multicast	4
l3/v4/broadcast	5
l3/v6	6
l3/v6/unicast	7
l3/v6/multicast	8
errors	9
errors/l2	10
errors/l2/rx	11
errors/l2/rx/crc-error	12
errors/l2/rx/invalid-mac	13
errors/l2/rx/invalid-vlan	14

errors/l2/rx/invalid-frame	15
errors/l2/tx	16
errors/l3	17
errors/l3/rx	18
errors/l3/rx/checksum-error	19
errors/l3/rx/mtu-exceeded	20
errors/l3/rx/invalid-packet	21
errors/l3/ttl-expired	22
errors/l3/no-route	23
errors/l3/invalid-sid	24
errors/l3/invalid-label	25
errors/l3/tx	26
errors/internal	27
errors/internal/parity-error	28
policy	29
policy/l2	30
policy/l2/acl	31
policy/l3	32
policy/l3/acl	33
policy/l3/policer	34
policy/l3/null-route	35
policy/l3/rpf	36
policy/l3/ddos	37
no-buffer	38

Table 1: Flow discard classification values and corresponding discard classes

For discard classes where per-traffic-class granularity is operationally significant (e.g., no-buffer, policy/l3/policer), the traffic class SHOULD be conveyed via companion IEs in the same Flow Record (e.g., ipDiffServCodePoint for L3, dot1qPriority for L2). This enables correlation with per-class interface counters from [I-D.ietf-opsawg-discardmodel].

3.4. Implementation Requirements

3.4.1. Semantics and Scope

1. Scope of this IE. flowDiscardClass MUST be used only to report flow-level discard classification under flow/discards from [I-D.ietf-opsawg-discardmodel]. It MUST NOT be used for interface, device, or control-plane discard counters.

2. Enumeration. Exporters MUST encode values only from the IANA "flowDiscardClass Values" subregistry for this IE. Collectors MUST accept both aggregate and leaf values and interpret aggregates as semantic supersets of their descendants.

3. Unknown/Unassigned values. Collectors receiving an unknown or unassigned value MUST treat it as unknown and MUST NOT remap it to another code. Exporters MUST NOT transmit unassigned values.

4. Reversibility. The value of flowDiscardClass MUST NOT change under biflow reversal as defined by [RFC5103].

3.4.2. Exporter Requirements

1. Cardinality. A Flow Record MUST contain at most one instance of flowDiscardClass.

2. Multiplicity. When multiple discard reasons apply to the same flow interval, exporters MUST export multiple Flow Records, one per discard reason. Each Flow Record MUST carry the same flow keys (5-tuple, interfaces, timestamps) but a distinct flowDiscardClass value. Where possible, exporters SHOULD include per-reason droppedPacketDeltaCount and/or droppedOctetDeltaCount to quantify the volume attributed to each specific discard class.

* While this approach creates multiple Flow Records for the same flow 5-tuple, it provides crucial diagnostic granularity. Collectors can easily aggregate by summing dropped counts across records with the same flow keys, while preserving the ability to attribute loss to specific root causes. This design maintains full fidelity of per-reason discard statistics.

3. Specificity. Exporters SHOULD report the most-specific known class (a leaf). If the specific leaf is unknown, an appropriate parent/aggregate MAY be used.

4. Interval semantics. When exported on an interval Flow Record, the presence of flowDiscardClass indicates that at least one packet in the interval matched that class. Exporters MUST include droppedPacketDeltaCount and/or droppedOctetDeltaCount in the same record to quantify the volume attributed to that specific discard reason. When multiple discard reasons affect the same flow (per point 2), the sum of per-reason dropped counts across all records for that flow represents the total flow-level discards.

5. Traffic class context. For discard classes where per-class correlation is operationally significant (e.g., no-buffer, policy/l3/policer), exporters SHOULD include a traffic-class IE in the same record (e.g., ipDiffServCodePoint or ipClassOfService for L3, dot1qPriority for L2). If classification occurs after remarking, exporters SHOULD use the post-remark class, or provide a device queue-ID→class mapping via IPFIX Options data.

6. Context. To aid correlation with interface/device/control-plane counters, exporters SHOULD include time bounds (flowStart/flowEnd or an observation-time IE), ingressInterface/egressInterface as applicable, and observationPointId when multiple pipeline stages/ taps exist.

3.4.3. Collector Requirements

1. Multiple records per flow. When multiple Flow Records carry different flowDiscardClass values for the same flow keys and overlapping time intervals, collectors MUST treat them as indicating distinct discard reasons affecting the same flow. Collectors SHOULD aggregate these records when computing per-flow total discards, while preserving per-reason breakdowns for root cause analysis.

2. Aggregate handling. When a parent/aggregate class is received, collectors MUST treat it as a coarse classification that may encompass multiple leaves.

3. Traffic class correlation. When a traffic-class IE is present alongside no-buffer or policy/l3/policer, collectors SHOULD use it to correlate with per-class interface counters. If absent, collectors MAY apply local device mappings if available.

4. Unknown values. Collectors MUST handle unknown/unassigned values gracefully (e.g., categorize as “unknown”) without rejecting the record.

3.4.4. Interoperability with Existing IPFIX IEs

1. Exporters and collectors MAY also use existing IEs (e.g., flowDirection, ipVersion, addresses, ipDiffServCodePoint) for filtering, correlation, or redundancy.

2. flowDiscardClass alone MUST be sufficient to recover the discard classification.

3. Exporters MAY continue to export forwardingStatus ([RFC7270]) in parallel. When both are present, flowDiscardClass MUST be considered authoritative for discard classification.

4. When flow sampling is active, the presence of flowDiscardClass indicates at least one sampled packet matched that class.

4. Security Considerations

This document defines a new Information Element for flow-level discard classification to align with the information model defined in [I-D.ietf-opsawg-discardmodel]. As such, there are no security issues related to this document, which are additional to those discussed in [RFC7011], [RFC7012].

5. IANA Considerations

IANA is requested to make the following changes under the IP Flow Information Export (IPFIX) Information Elements registry.

5.1. New IPFIX Information Element: flowDiscardClass

IANA is requested to register a new Information Element as follows:

* Name: flowDiscardClass

* ElementId: TBD (to be assigned by IANA)

* Description: Classifies the reason a packet was discarded in a flow, using the hierarchical classification scheme defined in [I-D.ietf-opsawg-discardmodel].

* Abstract Data Type: unsigned8

* Data Type Semantics: identifier

* Units: none

* Range: 0..38 (values are listed in the “flowDiscardClass Values” subregistry created below; other values are unassigned and MUST be treated as unknown)

* Reversibility: reversible (value does not change under flow reversal as per [RFC5103])

* Status: current

* Reference: This document; [RFC7013]

5.2. New Subregistry: “flowDiscardClass Values”

IANA is requested to create a new subregistry titled "flowDiscardClass Values" under the IPFIX Information Elements registry. This subregistry contains the enumerated values for the flowDiscardClass IE.

* Registration Procedure: Expert Review ([RFC8126])

* Reference: This document; [RFC7013]

* Fields:

  - Value (integer)

  - Name (path under flow/discards/...)

  - Description (optional)

  - Reference

Designated Expert guidance: New code points should reflect additions to or clarifications of discard reasons in [I-D.ietf-opsawg-discardmodel] (or its successor). Existing code points MUST NOT be repurposed. Backwards-compatible additions are

preferred. Experts SHOULD maintain the hierarchical structure (e.g., assigning aggregates and leaves consistently) and, where practical, preserve preorder (depth-first) numbering to align with the existing tree.

6. References

6.1. Normative References

[I-D.ietf-opsawg-discardmodel]
Evans, J., Pylypenko, O., Haas, J., Kadosh, A., and M. Boucadair, "Information and Data Models for Packet Discard Reporting", Work in Progress, Internet-Draft, draft-ietf-opsawg-discardmodel-10, 26 November 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-discardmodel-10>.

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

[RFC5103] Trammell, B. and E. Boschi, "Bidirectional Flow Export Using IP Flow Information Export (IPFIX)", RFC 5103, DOI 10.17487/RFC5103, January 2008, <https://www.rfc-editor.org/rfc/rfc5103>.

[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/rfc/rfc7011>.

[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/rfc/rfc7012>.

[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/rfc/rfc7013>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

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

6.2. Informative References

[RFC7270] Yourtchenko, A., Aitken, P., and B. Claise, "Cisco-Specific Information Elements Reused in IP Flow Information Export (IPFIX)", RFC 7270, DOI 10.17487/RFC7270, June 2014, <https://www.rfc-editor.org/rfc/rfc7270>.

Appendix A. Correlating Flow Discards with Interface/Device/Control-Plane Discards

This appendix is non-normative. It describes how to map high-level interface discard counters (from [I-D.ietf-opsawg-discardmodel]) to the specific flows responsible for or affected by those discards.

A.1. Correlation Keys

To correlate a discard counter anomaly with flow records, the collector must join data on three key dimensions:

1. Time: Align the counter collection interval with the Flow Record start/end times (allowing for small clock skew).

2. Location: Match the Observation Domain and Interface.

    * For Ingress discards: match ingressInterface.

    * For Egress discards: match egressInterface.

3. Discard Class: Match the YANG discard-class leaf with the IPFIX flowDiscardClass value.

    * Note: If the drop is specific to a traffic class (e.g., no-buffer), the collector must also match the traffic class identifier (e.g., ipDiffServCodePoint) to the specific queue experiencing loss.

A.2. Analysis Strategies

Once flow records are correlated with discard counters, operators can rank or group flows to determine:

* Impacted analysis: Which flows suffered loss? This is determined by grouping flows by the presence of the flowDiscardClass (or summing dropped-octets/packets) to identify the symptomatic flows of the event.

* Causal analysis (when meaningful): Which flows likely contributed to the interface/device condition? For congestive discards (e.g. no-buffer), this is determined by identifying the top senders (by total volume or rate) in the same traffic class and egress interface during the anomaly.

A.3. Operational Example: Impacted Flows (Congestion Drops)

Scenario: an anomaly is detected in no-buffer discards on Ethernet1/0 (ifIndex 10) in the egress direction. The drops are occurring in the Best Effort queue (DSCP 0).

1. Signal: Interface discard counter

    * Time: 2025-09-18 10:00:00 10:01:00

    * Observation Domain: 1234

    * Interface: 10 (egress)

    * Class: no-buffer (value 38; see Table 1)

    * Queue/DSCP: 0

2. Correlation: SQL Query

    The operator queries the IPFIX store to perform impact analysis — identifying symptomatic flows of the congestion event:

    ```
    sql SELECT src_addr, dst_addr, l4_dst_port, protocol,
    SUM(droppedPacketDeltaCount) AS total_pkt_discards FROM
    flow_records WHERE -- 0. Match Observation Domain
    observationDomainId = 1234 -- 1. Match Location (egress
    interface) AND egressInterface = 10 -- 2. Match Time Window (any
    overlap with counter interval) AND flowEnd >= '2025-09-18
    10:00:00' AND flowStart <= '2025-09-18 10:01:00' -- 3. Match
    Discard Class (no-buffer) AND flowDiscardClass = 38 -- 4. Match
    Traffic Class context (Best Effort) AND ipDiffServCodePoint = 0
    GROUP BY src_addr, dst_addr, l4_dst_port, protocol ORDER BY
    total_pkt_discards DESC LIMIT 10;
    ```

3. Result

The query returns the top flows most affected by the discard event, allowing the operator to pinpoint specific applications or users impacted by the congestion:

src_addr	dst_addr	l4_dst_port	protocol	total_pkt_discards
192.0.2.10	198.51.100.55	443	6 (TCP)	15,400
192.0.2.12	198.51.100.80	80	6 (TCP)	2,100

Table 2

A.4. Operational Example: Causal Flows (Congestion Drops)

Using the same scenario as in Appendix A.3, the operator now wants to identify flows that likely caused the congestion — that is, heavy senders in the affected queue and interface during the anomaly. These flows may or may not themselves have experienced drops.

1. Signal: Interface discard counter

   Same as in Section A.3:

   * Time: 2025-09-18 10:00:00 10:01:00
   * Observation Domain: 1234
   * Interface: 10 (egress)
   * Class: no-buffer (value 38)
   * Queue/DSCP: 0

2. Correlation: SQL Query

   The operator queries the IPFIX store to perform a causal analysis by ranking flows by total traffic volume in the same time window, interface, and traffic class. The query does not require flowDiscardClass = 38, since flows can contribute to congestion even if only some packets (or none of the sampled packets) were dropped.

   ```
   sql SELECT src_addr, dst_addr, l4_dst_port, protocol,
   SUM(octetDeltaCount) AS total_bytes, SUM(packetDeltaCount) AS
   total_pkts, SUM(droppedPacketDeltaCount) AS total_pkt_discards
   FROM flow_records WHERE -- 0.  Match Observation Domain
   ```

observationDomainId = 1234 -- 1. Match Location (egress interface) AND egressInterface = 10 -- 2. Match Time Window (any overlap with counter interval) AND flowEnd >= '2025-09-18 10:00:00' AND flowStart <= '2025-09-18 10:01:00' -- 3. Match Traffic Class context (Best Effort queue) AND ipDiffServCodePoint = 0 GROUP BY src_addr, dst_addr, l4_dst_port, protocol ORDER BY total_bytes DESC LIMIT 10;

3. Result

This query returns flows that carried the most traffic through the congested interface and queue during the interval. These high-volume flows are candidates for having contributed to the congestion. The total_drops column (if present) can still be used to see which of these heavy flows also suffered loss.

src_addr	dst_addr	l4_dst_port	protocol	total_bytes	total_pkts	total_pkt_discards
10.0.0.5	192.0.2.200	443	6 (TCP)	850,000,000	1,214,285	2,100
192.0.2.10	198.51.100.55	443	6 (TCP)	15,000,000	21,000	15,400

Table 3

In this example, the flow from 10.0.0.5 transferred 850 MB with limited discards, while the smaller flow from 192.0.2.10 suffered significant packet loss.

A.5. Implementation Note on Sampling

When flow sampling is active, flowDiscardClass indicates that a sampled packet was dropped. To estimate the total (unsampled) flow-level impact and compare it with interface counters (which are typically unsampled), operators can apply a sampling-rate multiplier to the flow counters (droppedPacketDeltaCount and/or droppedOctetDeltaCount).

Let: * p = the sampling probability (e.g., 0.01 for 1-in-100 sampling) * N = 1 / p be the corresponding "1-in-N" sampling interval.

The sampling-rate multiplier is N. Estimated totals are then: * estimated_total_dropped_packets = droppedPacketDeltaCount * N * estimated_total_dropped_octets = droppedOctetDeltaCount * N

For example, if the exporter samples 1 in every 100 packets, the multiplier is 100. If the exporter samples 1 in every 1000 packets, the multiplier is 1000.

Exporters typically report their sampling configuration via IPFIX (samplingInterval and/or samplingProbability). Because sampling is probabilistic, these estimates are approximate; using larger time windows and higher-volume aggregates tends to make them more robust.

Authors' Addresses

John Evans
Amazon
1 Principal Place, Worship Street
London
EC2A 2FA
United Kingdom
Email: jevanamz@amazon.co.uk

Oleksandr Pylypenko
Amazon
410 Terry Ave N
Seattle, WA 98109
United States of America
Email: opyl@amazon.com

Karim Cheaito
Amazon
410 Terry Ave N
Seattle, WA 98109
United States of America
Email: kcheaito@amazon.com