Workgroup: IP Flow Information Export
Internet-Draft:
draft-mvmd-opsawg-ipfix-fwd-exceptions-04
Published: 7 February 2022
Intended Status: Standards Track
Expires: 11 August 2022
Authors: C. Munukutla
Juniper Networks, Inc.
S. Vaid
Juniper Networks, Inc.
A. Mahale
Google, Inc.
D. Patel
Google, Inc.

IP Flow Information Export (IPFIX) Information Elements Extension for Forwarding Exceptions

Abstract

This draft proposes couple of new Forwarding exceptions related Information Elements (IEs) and Templates for the IP Flow Information Export (IPFIX) protocol. These new Information Elements and Exception Template can be used to export information about any forwarding errors in a network. This essential information is adequate to correlate packet drops to any control plane entity and map it to an impacted service. Once exceptions are correlated to a particular entity, an action can be assigned to mitigate such problems essentially enabling self-driving networks.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on 11 August 2022.

Table of Contents

1. Introduction

All networks are susceptible to traffic drops due to a number of factors. Traffic drops can go unnoticed unless they are service impacting. In a multi-layered network architecture, it is tedious manual work to localize and root cause traffic blackholing issues. Transient drops are even harder to detect. Existing methodologies that rely on periodically monitoring interfaces on several hosts in a network does not guarantee timely detection, and are not scalable for large networks.

In order to eliminate this tedious monitoring work-flow, objective is to simplify localization and build correlation of dropped packets to particular entity. The network entity shall identify the dropped

packets by monitoring dropped counters or doing a deep packet inspection of the packet discarded by the forwarding ASIC. The implementation of the method used to detect the drop is outside the scope of this document. Dropped packets will be sampled in the forwarding-path and sent to a host or software queue along with type of exception, in/out interface information and other relevant meta data. This will be a push model where the node encountering the error will emit the information about dropped packets and associated meta-data. Techniques for IP Packet Selection [RFC5475] describes Sampling and Filtering techniques for IP packet selection either using Systematic Sampling or Random Sampling.

The IPFIX Protocol Specification [RFC7011] defines a generic exchange mechanism for collecting flow information. It supports source-triggered export of information via the push model approach. The IPFIX Information Model [IANA-IPFIX] defines a list of standard Information Elements (IEs) which can be carried by the IPFIX protocol.

This document focuses on telemetry information for dropped packet exceptions, and proposes an extension to IPFIX message format for collecting sampled exceptions. Some of the IPFIX Information Elements (IEs) already exist, some will be defined along with corresponding formats. It is also possible to achieve sampling of the dropped packets by using sampling methods like SFLOW but details of other sampling methods are outside the scope of this document.

1.1. Terminology

IPFIX-specific terminology (e.g. Information Element, Template, Template Record, Options Template Record, Template Set, Collector, Exporter, Data Record) used in this document is defined in Section 2 of [RFC7011]. As in [RFC7011] these IPFIX-specific terms have the first letter of a word capitalized. This document also makes use of the same terminology and definitions as Section 2 of [RFC5470].

1.2. Requirements Language

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.

2. Scope

This document specifies the information model used for reporting packet-based forwarding exceptions. [RFC7011] provides guidance on the choices of the transport protocols used for IPFIX and their effects. Encoded IPFIX exception packets need to be reliably

transported to the collector. The choice of the actual transport protocol is beyond the scope of this document.

This document assumes that all devices reporting exceptions will use existing IPFIX framework/module to send encoded packets to the collector. This would mean that the network device will specify the template that it is going to use for each of the events. The templates can be of varying length, and there could be multiple templates that a network device could use to encode the exceptions.

The implementation details of the collector application are beyond the scope of this document.

3. Information Elements

The Exception template could contain a subset of the IEs shown in Table 1, depending upon the exception reported.

Whenever packet drop happens inside forwarding plane, following information is key to understanding the issue: reason for packet drop, flow which encountered the drop (packet content), additional meta-data e.g. flow direction (ingress/egress), nexthop index, input interface, output interface, etc. on which this packet was flowing.

The following table includes all the existing IEs that a device reporting IPFIX Exceptions using various Exception Templates would typically need. The formats of IEs and IPFIX IDs are listed in the table below.

Field Name	Size (bits)	IANA IPFIX ID	Description
flowDirection	8	61	The direction of the Flow observed at Observation point.
ingressInterface	32	10	Index of IP interface where packets of this flow are being received.
egressInterface	32	14	Index of IP interface where packets of this flow are being sent.
dataLinkFrameSize	16	312	Specified length of data link frame.
dataLinkFrameSection	65535	315	Carries n octets from data link frame of selected frame.
commonPropertiesID	64	137	Identifier of a set of common properties that is unique per observation domain.

Table 1: Forwarding Exception Information Elements

Figure 1: Information Elements

4. New Information Elements

4.1. Proposed New Information Elements

The proposed new IEs that a device reporting Exceptions using Exception template would need are listed in Table 2 below.

Field Name	Abstract Data Type	Description
forwardingExceptionCode	unsigned32	Unique code for every exception
forwardingNextHopID	unsigned64	Forwarding NH - index associated with packet that encountered this exception
forwardingLookupType	unsigned8	Last Lookup type performed on the packet in in ingress path. For instance, IPV4, IPV6, Bridge, MPLS, Unknown etc.
underlyingIngressInterface	unsigned32	Underlying interface from which a packet arrived in ingress path. For instance, child interface of aggregate interface on which packet came in ingress; where aggregate interface is captured in ingressInterface

Table 2: New Information Elements

Figure 2: New Information Elements

The Information Elements defined in Table 2 are proposed to be incorporated into the IANA IPFIX Information Elements registry [IANA-IPFIX]

4.2. Definition of Exceptions

Every network will encounter issues like packet loss, from time to time. Some of the causes for such a loss of traffic or a block in transmission of data packets include overloaded system conditions, misconfiguration, profiles and policies that restrict the bandwidth or priority of traffic, network outages, or disruption with physical cable faults. Packet loss could also happen because of incorrect stitching of the forwarding path or a mismatch between control plane and data plane state. Exception code entails the reason/error code due to which this packet has been dropped.

4.2.1. forwardingExceptionCode

forwardingExceptionCode will be defined in "IPFIX Information Elements" registry. This list can be expanded in the future as necessary. The data record will have corresponding exception code value to indicate forwarding error that caused the traffic drop.

An implementation may choose to encode device internal exception codes as forwardingExceptionCode. In such scenarios, Enterprise Bit MUST be set to 1 and corresponding Enterprise Number MUST be present as described in [RFC7011]

There is an existing IE 89 - forwardingStatus [IANA-IPFIX] but it allows a very limited number of exceptions to be reported from the system (6-bit reason code). The exception codes also need to be standardized for use. Different forwarding ASICs would have different pipelines and hence discard reasons (which could be very specific to that pipeline) cannot be generalized. Hence it makes sense to have a standalone IE for reporting exception which not only provides support to report larger number of exceptions but also provides freedom for reporting application specific exceptions using the enterprise bit.

forwardingExceptionCode will also describe status of the flow with first two bits. An implementation may choose to export forwardingExceptionCode instead of IE89 - forwardingStatus.

A list of commonly used forwarding Exception codes will be identified and listed as part of Table 3 below.

Forwarding Exception Code	Reason
1	FIREWALL_DISCARD
2	TTL_EXPIRY
3	DISCARD_ROUTE
4	BAD_IPV4_CHECKSUM
5	REJECT_ROUTE
6	BAD_IPV4_HEADER (Version incorrect or IHL < 5)
7	BAD_IPV6_HEADER (Version incorrect)
8	BAD_IPV4_HEADER_LENGTH (V4 frame is too short)
9	BAD_IPV6_HEADER_LENGTH
10	BAD_IPV6_OPTIONS_PACKET(too many option headers)
..	..

Figure 3: Table 3: Exception Codes

4.2.2. forwardingNexthopId

In terms of a network device, next hop is the gateway to which packet should be forwarded corresponding to the path to final destination. A given router doesn't need to store the entire forwarding path information for a destination. As long as it can identify the next hop to be used for forwarding to a destination, the end to end forwarding can happen. This helps reduce size of forwarding table. The nexthop index uniquely identifies the egress path a packet would take to reach the destination. This could include information about the outgoing interface, layer 2 address to be used, forwarding features configured for the packet path etc.

For instance, consider we have a L3VPN topology like below

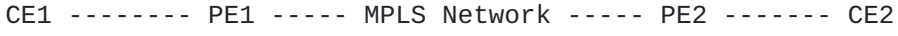

Figure 4: Figure 1: MPLS VPN Network

Figure 1 above illustrates an example where reporting of exception can provide an insight into the error scenario. CE1 and CE2 communicate with each other over an MPLS VPN network. The labels are typically advertised using protocols like RSVP or LDP. When a packet is received from core network on PE1, a lookup on MPLS label results in packet getting forwarded towards CE1. The entries in MPLS table are populated by corresponding protocol. If label entries don't get populated in the MPLS table due to a probable glitch in the protocol configuration or some software inconsistency, the packets traversing on that LSP tunnel path shall get discarded on PE1.

In case of route lookups, that result in hierarchical forwarding chains, the mis-programming may manifest at different levels of the forwarding structure. The forwarding lookup may fail on any level of the hierarchy in the forwarding chain. It is expected that software at least report the nexthop where the lookup terminates. Its desirable for software to report the top level nexthop in the chain.

Using the mechanism described in this RFC, it will be possible to capture such packets and report them in IPFIX format with corresponding exception set (eg. DISCARD_ROUTE) along with relevant packet bytes and meta-data. This can help the operator/software to immediately understand root cause of the problem and take appropriate action.

An implementation may choose to report linecard number, linecard type, forwarding ASIC type and forwarding ASIC number on which an exception occurs, but mechanism to export these fields is out of the scope of this document.

4.2.3. forwardingLookupType

A packet might undergo multiple lookups in the forwarding chain. Lookup may fail at any level of the lookup hierarchy. When an exception is reported in such cases, type of the last lookup performed on the packet may help in identifying nature of the erroneous path.

For instance, a Firewall Discard may happen for Layer2 or Layer3 packet. All such packets may be treated as FIREWALL_DISCARD for generic exception reporting purposes. However, exact place of error in the pipeline (IPV4, IPV6, MPLS etc.) may help with easily correlating the exception.

4.2.4. underlyingIngressInterface

A packet can arrive on an aggregate ethernet(ae) interface where the receive interface is the ae but actual physical interface is a child member of this ae. If such a packet gets dropped because of an exception, it will be very useful not only to know about the ae on which it arrived but also the child link of that ae on which the packet was received.

underlyingIngressInterface represents the interface underlying the received interface (which in case of ae would be its child link) on which the packet arrived in ingress. This helps in providing more context about the nature of the packet processing for this path.

5. Exception Templates

This section presents a list of templates for reporting exceptions using newly proposed IEs in addition to few existing Information Elements (IEs).

Templates listed below are sample templates to demonstrate the utility of newly introduced Information Elements in conjuction with existing Information Elements to report meaningful data to the collector. A specific implementation may add or remove Information Elements from below templates based on their reporting requirements.

5.1. IPFIX Exception Template 1 for Forwarding Exceptions

Exception Template defined in Figure 1 demonstrates a sample set of data to export forwarding Exceptions.

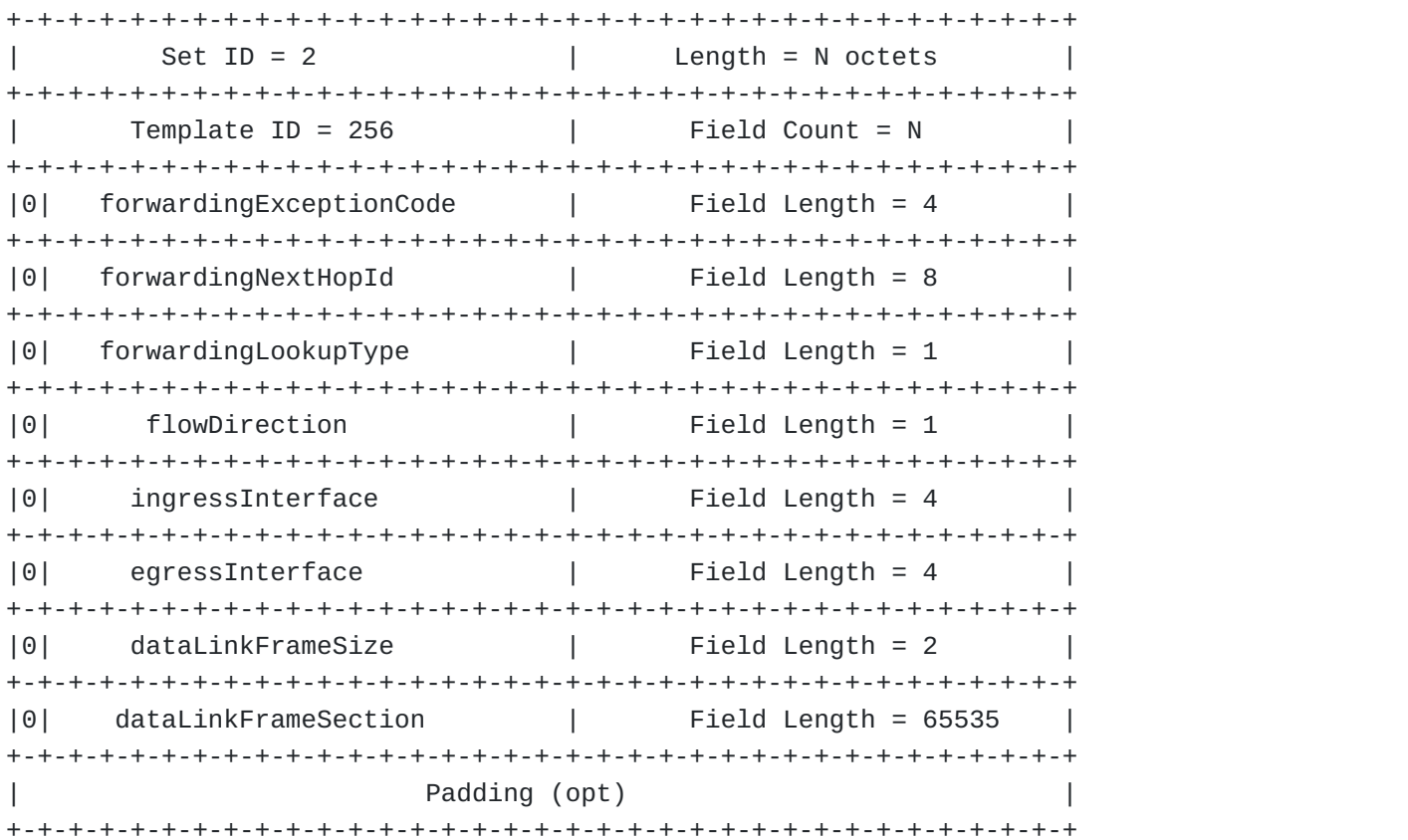

Figure 5: IPFIX Exception Template for Forwarding Exceptions

5.2. IPFIX Exception Template 2 for Forwarding Exceptions

Alternatively, Exception Template defined in Figure 2 is a sample template to export forwarding exceptions. This template demonstrates the use of Information Element 137 to represent following fields:

forwardingExceptionCode, forwardingNexthopId, ingressInterface, underlyingIngressInterface and egressInterface.

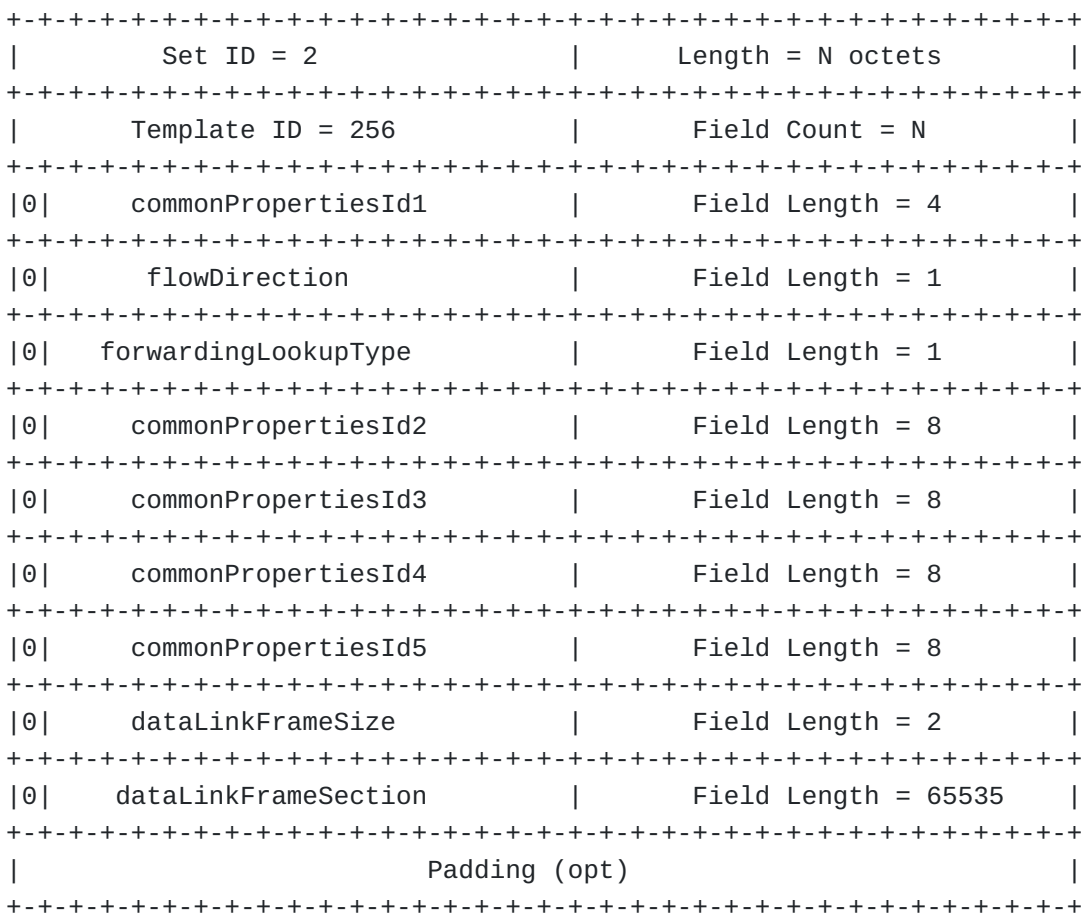

Figure 6: IPFIX Exception Template 2 for Forwarding Exceptions

6. IANA Considerations

6.1. Information Elements

IANA manages the IPFIX Information Elements registry at [IANA-IPFIX]. This document introduces two new IPFIX Information Elements.

Name: forwardingExceptionCode ElementID: TBD Description: Exception code is an identifier uniquely describing cause of irregularity or traffic drop on a device. Abstract Data Type: unsigned32 Data Type Semantics: identifier

Name: forwardingNexthopId ElementID: TBD Description: Nexthop ID is a unique identifier for a Nexthop on a device. Abstract Data Type: unsigned64 Data Type Semantics: identifier

Name: forwardingLookupType ElementID: TBD Description: Represents the last lookup performed on the packet in forwarding path. Abstract Data Type: unsigned8 Data Type Semantics: identifier

Name: underlyingIngressInterface ElementID: TBD Description: The underlying interface index of the interface from where packet of a given flow are received in ingress. For example, child interface of an aggregate ethernet interface. Abstract Data Type: unsigned32 Data Type Semantics: identifier

6.2. Forwarding Exception Codes

This document requests addition of a new registry for Forwarding Exception Codes.

Forwarding Exception Code	Reason
1	FIREWALL_DISCARD
2	TTL_EXPIRY
3	DISCARD_ROUTE
4	BAD_IPV4_CHECKSUM
5	REJECT_ROUTE
6	BAD_IPV4_HEADER (Version incorrect or IHL < 5)
7	BAD_IPV6_HEADER (Version incorrect)
8	BAD_IPV4_HEADER_LENGTH (V4 frame is too short)
9	BAD_IPV6_HEADER_LENGTH
10	BAD_IPV6_OPTIONS_PACKET(too many option headers)
..	..

Figure 7: Table 3: Exception Codes

All assignments in this registry are to be performed via Expert Review.

7. Security Considerations

Security Considerations listed in detail for IPFIX in [RFC7011] apply to this document as well. As described in [RFC7011], the IPFIX messages exchanged between network device and collector MUST be protected to provide confidentiality, integrity, and authenticity. Without those characteristics, the messages are subject to various kinds of attacks. These attacks are described in great detail in [RFC7011].

8. Contributors

Manikandan Musuvathi Poornachary
Juniper Networks, Inc.
Electra Exora Business Park~Marathahalli-Sarjapur Outer Ring Road,
Bangalore, KA - 560103
India
Email: mpoornachary@juniper.net

Vishnu Pavan Beeram
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
USA
Email: vbeeram@juniper.net

Raveendra Torvi
Juniper Networks, Inc.
10 Technology Park Dr
Westford, MA 01886
USA
Email: rtorvi@juniper.net

9. References

9.1. 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>.

[RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", RFC 5475, DOI 10.17487/RFC5475, March 2009, <https://www.rfc-editor.org/info/rfc5475>.

[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>.

[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>.

9.2. Informative References

[IANA-IPFIX]

IANA, "IP Flow Information Export (IPFIX) Entities", <https://www.iana.org/assignments/ipfix>.

[RFC5470] Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", RFC 5470, DOI 10.17487/RFC5470, March 2009, <https://www.rfc-editor.org/info/rfc5470>.

Authors' Addresses

Venkata Naga Chaitanya Munukutla
Juniper Networks, Inc.
10 Technology Park Dr
Westford, MA 01886
United States of America

Email: vmunuku@juniper.net

Shivam Vaid
Juniper Networks, Inc.
Electra, Exora Business Park- Marathahalli-Sarjapur Outer Ring Road
Bangalore 560103
Karnataka
India

Email: shivamv@juniper.net

Aditya Mahale
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
United States of America

Email: amahale@google.com

Devang Patel
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
United States of America

Email: pateldevang@google.com