

IDR  
Internet-Draft  
Intended status: Informational  
Expires: 10 November 2026

Y. Cui  
Tsinghua University  
Y. Gao  
Zhongguancun Laboratory  
S. Hares  
Hickory Hill Consulting  
9 May 2026

Packet Content Filter for BGP FlowSpec  
draft-cui-idr-content-filter-flowspec-04

Abstract

The BGP Flow Specification enables the distribution of traffic filter policies (traffic filters and actions) via BGP, facilitating DDoS traffic filtering. However, the traffic filter in FSv1 and FSv2 predominantly focuses on IP header fields, which may not adequately address volumetric DDoS attack traffic characterized by fixed patterns within the packet content. This document introduces a new flow specification filter type designed for packet content filtering. The match field includes ptype, otype, offset, content-length, content, and mask encoded in the Flowspec NLRI. This new filter aims to leverage network devices such as routers and switches to support controlled traffic handling, traffic optimization, and mitigation of simple volumetric DDoS attacks, reducing the overall processing cost of carrier 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 10 November 2026.

## Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Terminology . . . . .	3
2. Definitions and Acronyms . . . . .	3
3. The Packet Content Filter for FSv1 . . . . .	3
3.1. Ptype Field . . . . .	4
3.2. Otype and Offset Fields . . . . .	4
3.3. Content-length, Content and Mask Fields . . . . .	5
3.4. Example of Encoding . . . . .	6
4. The Packet Content Filter for FSv2 . . . . .	7
4.1. Filter Encoding . . . . .	7
4.2. Filter Ordering Rule . . . . .	8
4.3. Use Cases . . . . .	9
5. Operational Considerations . . . . .	10
6. Scalability Considerations . . . . .	10
7. Security Considerations . . . . .	11
8. IANA Considerations . . . . .	11
9. Normative References . . . . .	11
Acknowledgements . . . . .	12
Authors' Addresses . . . . .	12

## 1. Introduction

BGP flow specification describes the distribution of traffic filter policies through BGP, allowing for efficient traffic management and DDoS attack mitigation. Existing versions, FSv1 and FSv2, primarily offer n-tuple matching conditions for policy enforcement, enabling actions such as packet dropping, re-directing, or other actions. These filter rules can be propagated to all BGP peers simultaneously without necessitating router configuration changes. Despite their utility, the reliance of existing filters on IP header fields may be insufficient for some operational scenarios where packets can be identified by fixed content patterns. Such scenarios may include

DDoS mitigation, traffic filtering, and traffic optimization. Some attacks or traffic classes may contain fixed patterns in the packet payload that can be matched at known offsets.

This document defines a new FlowSpec filter type that supports packet content filtering by using ptype, otype, offset, content-length, content, and mask fields within the FlowSpec NLRI. This filter is intended for controlled operational use cases such as traffic filtering, traffic optimization, and DDoS mitigation.

### 1.1. 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.

## 2. Definitions and Acronyms

- \* DDoS: Distributed Denial of Service.
- \* NLRI: Network Layer Reachability Information.
- \* FSv1: Flow Specification Version 1, defined in [RFC8955] and [RFC8956].
- \* FSv2: Flow Specification Version 2, defined in [I-D.ietf-idr-flowspec-v2].

## 3. The Packet Content Filter for FSv1

This document specifies a new flow specification filter type that is encoded in the BGP FS NLRI, following the FSv1 definition format. The packet content filter is defined as follows:

Type TBD â\200\223 Packet-Content

Encoding:< type (1 octet), value>

The value field is encoded using ptype, otype, offset, content-length, content and mask.

Encoding: < ptype (4 bits), otype (4 bits), offset (2 octets), content-length (1 octet), content (Variable), mask (Variable)>

### 3.1. Ptype Field

The ptype is defined as a 4-bit unsigned integer that defines the packet type via AFI, because some filters are added to hardware that are IPv4 or IPv6 specific.

Value	Description of Ptype
1	IPv4
2	IPv6

Figure 1: Ptype field

### 3.2. Otype and Offset Fields

The otype and offset fields define the starting position of the packet content used for matching.

To avoid the effect of variable header length on the offset, we use the hierarchical way like [I-D.khare-idr-bgp-flowspec-payload-match]. The Otype is defined as a 4-bit unsigned integer. The detail are as follows:

Value	Description of Otype
0	IP Header
1	IP Payload
2	UDP Payload
3	TCP Payload

Figure 2: Otype field

Otype 0 is defined as the start of the IP header. Otype 1 is defined as the start of the data portion of the IP header after the IP options. Otype 2 is defined as the start of the UDP payload. Otype 3 is defined as the start of the TCP payload. Otype 2 MUST only match packets whose upper-layer protocol is UDP (17). Otype 3 MUST only match packets whose upper-layer protocol is TCP (6). For other IP protocols, otype 2 and otype 3 MUST NOT match; otype 1 MAY be used.

The offset is defined as a 2-octet unsigned integer that specifies the count of octets to be bypassed from the otype's starting position to match the packet content. It is worth noting that packet

fragmentation will cause the offset value to change, so it is not enough to filter the fragmented packets through the packet content filter. One possible way is to filter the first packet through the payload filter, and then use its header information along with the fragment filter to filter the subsequent packets.

Example:

- \* By setting otype 0 and an offset of 0, the match is configured to start precisely at the beginning of the IP header.
- \* By setting otype 1 and an offset of 2, the match will start two octets past the initial data portion of the IP header, skipping over any IP options. This configuration, for example, could be used to specifically target the IP payload starting after 2 octets.
- \* By setting otype 2 and an offset of 10, the match will start ten octets into the UDP payload of the packet.
- \* By setting otype 3 and an offset of 10, the match will start ten octets into the TCP payload of the packet.

### 3.3. Content-length, Content and Mask Fields

The content-length is a one-octet unsigned integer field that specifies the length, in octets, of each of the Content field and the Mask field. The Content field and the Mask field have the same length as specified by the content-length.

The Content field carries the octet sequence to be matched. Based on information provided by equipment vendors and operators, 8 octets is usually sufficient for identifying many fixed packet-content patterns used in operational filtering scenarios.

The Mask field is an octet string used as a bit mask for the Content field and the corresponding packet data. Each bit set to 1 indicates that the corresponding bit is significant for matching. Each bit set to 0 indicates that the corresponding bit is ignored.

A packet matches the Packet Content filter if the following comparison is true for the packet data at the specified offset:  
`(packet_content & mask) == (content & mask)`

## 3.4. Example of Encoding

An example of a FlowSpec NLRI encoding is provided for the following rule: "match all packets destined to 192.0.2.0/24 that have the fixed content 0x5858 at offset 0 in the TCP payload".

length	destination	packet content
0x0f	01 18 c0 00 02	TBD 40 12 00 00 02 58 58 ff ff

Table 1

Description of each field of the FlowSpec NLRI.

Value	Description	
0x0f	length	15 octets (if len<240, 1 octet)
0x01	type	Type 1 - Destination Prefix
0x18	length	24 bits
0xc0	prefix	192
0x00	prefix	0
0x02	prefix	2
TBD	type	Type TBD - Packet Content
0x40	length	64 bits
0x12	ptype, otype	IPv4, TCP payload
0x0000	offset	0 octets
0x02	content-length	2 octets
0x5858	content	0x5858
0xffff	mask	0xffff

Table 2

## 4. The Packet Content Filter for FSv2

### 4.1. Filter Encoding

To adapt to the updates of FlowSpec, this document also defines the Packet Content Filter for FSv2. The format follows the NLRI format for Extended IP Filters defined in [I-D.hares-idr-fsv2-more-ip-filters], as shown in Figure 3:

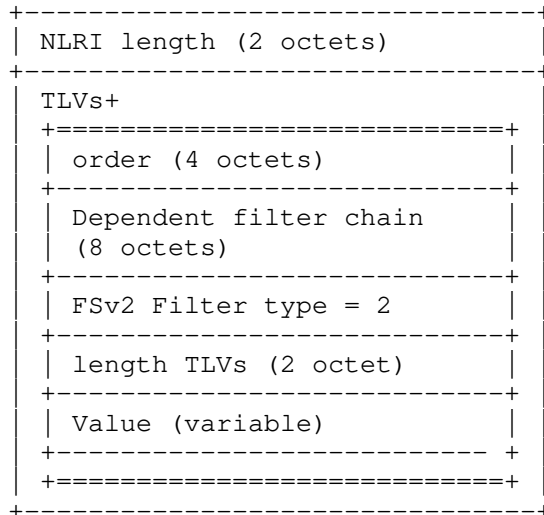


Figure 3: NLRI Format for Extended IP Filters

The format of the dependent filter chain is shown in Figure 4:

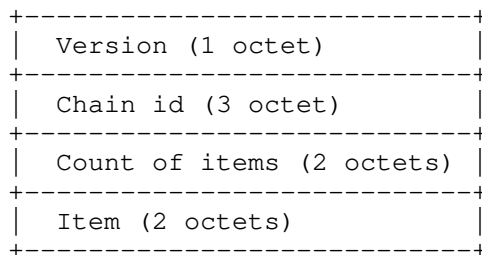


Figure 4: Format of the Dependent Filter Chain

The format of the Component TLV is shown in Figure 5:

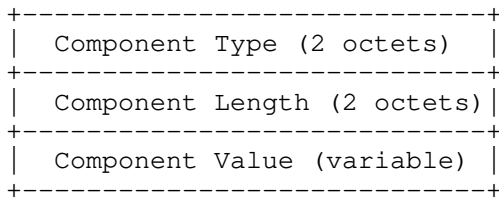


Figure 5: format for Component-TLV

The definition of the Packet Content Filter in the Component-TLV format is as follows:

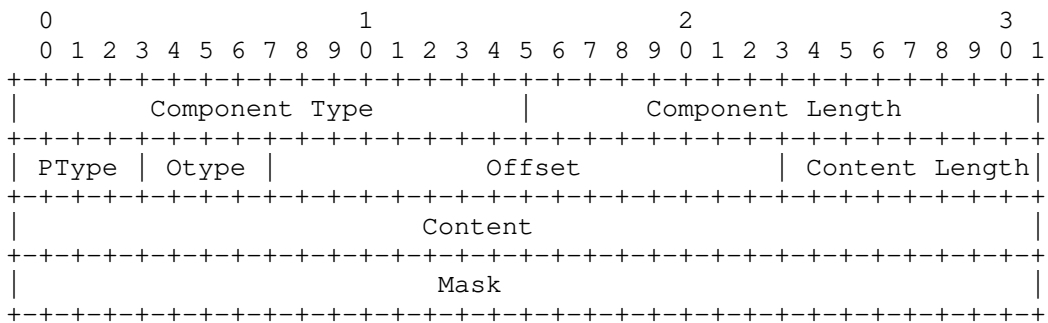


Figure 6: Definition of the Packet Content Filter

where the fields have the same definitions as in the FSv1 encoding.

Encoding: < ptype (4 bits), otype (4 bits), offset (2 octets), content-length (1 octet), content (Variable), mask (Variable)>

#### 4.2. Filter Ordering Rule

Compared to FSv1, FSv2 adds filter ordering function. According to the definition of ordering rules in FSv2, the transmission of Component-TLVs within a flow specification rule MUST be sent ascending order by Component-TLV type. If the Component-TLV types are the same, then the value fields are compared using mechanisms defined in [RFC8955] and [RFC8956] and MUST be in ascending order.

However, due to multiple fields in the value of the packet content filter, the mechanisms defined in [RFC8955] and [RFC8956] do not apply. To give the default ordering rules of packet content filters, this document gives the definition as follows:

1. Filters with a larger content-length are ordered first.

2. If they have the same content-length, compare otype and the larger type is ordered first.
3. If they have the same content-length and otype, compare offset and the larger value is ordered first.
4. If they have the same content-length, otype, and offset, compare the content as an unsigned octet string in lexicographic order, starting from the first octet. If the common prefix is not equal, the string with the lower octet value at the first differing position has higher precedence.

When multiple Packet Content Filter components exist across multiple NLRIs with the same user order, their relative order is determined according to the ordering rules above.

#### 4.3. Use Cases

Here is a use case for ordering rules with multiple NLRI and multiple components. There are five components, with the same destination IP and user order, each of which contains a packet content filter with different values:

User-Order â\200\223 10

FSv2 â\200\223 NLRI with Extended IP Filters

Component 1: Destination IP + Packet content filter (otype 0, offset 50, content-length 2, content 0x1111) + Rate Limit

Component 2: Destination IP + Packet content filter (otype 0, offset 50, content-length 3, content 0x111122) + Discard

Component 3: Destination IP + Packet content filter (otype 2, offset 70, content-length 2, content 0x1111) + Rate Limit

Component 4: Destination IP + Packet content filter (otype 2, offset 70, content-length 3, content 0x111122) + Discard

Component 5: Destination IP + Packet content filter (otype 2, offset 70, content-length 3, content 0x111133) + Rate Limit

The rules will be installed as:

User-Order 200223 10

Component 4: Destination IP + Packet content filter (otype 2, offset 70, content-length 3, content 0x111122) + Discard

Component 5: Destination IP + Packet content filter (otype 2, offset 70, content-length 3, content 0x111133) + Rate Limit

Component 2: Destination IP + Packet content filter (otype 0, offset 50, content-length 3, content 0x111122) + Discard

Component 3: Destination IP + Packet content filter (otype 2, offset 70, content-length 2, content 0x1111) + Rate Limit

Component 1: Destination IP + Packet content filter (otype 0, offset 50, content-length 2, content 0x1111) + Rate Limit

## 5. Operational Considerations

The Packet Content Filter is intended for controlled deployment scenarios, such as traffic filtering, traffic optimization, and DDoS mitigation based on known fixed packet-content patterns. Operators SHOULD deploy this filter only at controlled filtering locations, such as provider edge devices, traffic steering points, mitigation points, or other devices where the traffic impact and rollback procedures are well understood.

Operators SHOULD enable Packet Content Filter processing only on devices that support packet-content parsing and have sufficient filtering resources for the expected rule scale. Unsupported rules SHOULD be rejected or ignored locally according to local policy.

When encapsulation is present, such as MPLS, GRE, or other tunnels, the offset base can become ambiguous if matching is applied to the outer packet. Operators SHOULD apply matching to the decapsulated inner IP packet when applicable, or otherwise ensure that the offset base is unambiguous.

## 6. Scalability Considerations

Packet-content matching may consume limited implementation resources, such as UDF, ACL, or TCAM entries. Operators SHOULD limit Packet Content Filter rules to a small set of high-value entries, such as confirmed attack signatures, operationally validated filtering rules, or traffic optimization policies.

When FSv2 is used, rule ordering SHOULD be used to reduce the amount of traffic requiring packet-content inspection, for example by combining packet-content matching with more specific header-based conditions.

Operators SHOULD restrict the propagation scope of Packet Content Filter rules to avoid unnecessary inter-domain scale impact. Inter-domain propagation SHOULD be used only with explicit operational agreement and suitable policy control.

## 7. Security Considerations

This specification does not change the security properties of BGP itself. However, Packet Content Filter rules can affect traffic treatment and may cause packets to be dropped, redirected, rate-limited, or other actions according to local policy.

Operators MUST apply appropriate import policies, validation procedures, and authorization controls before accepting Packet Content Filter rules. Such rules SHOULD be accepted only from trusted BGP peers, and their propagation scope SHOULD be restricted by local policy.

To reduce false positives, Packet Content Filter rules SHOULD be combined with other FlowSpec match conditions, such as destination prefix, source prefix, protocol, port, TCP flags, or fragment-related conditions, when applicable.

Unsupported rules SHOULD be rejected or ignored locally according to local policy. Implementations and operators SHOULD apply update-rate limits and resource limits to avoid excessive control-plane load and preserve BGP stability.

## 8. IANA Considerations

IANA is requested to assign a new Type Value for the Packet Content Filter from the "Flow Spec Component Types" registry.

Type Value	Name	Reference
TBD	Packet Content filter	this document

For FSv2, a Packet Content Filter Component Type will be requested from the appropriate FSv2 Extended IP Filters component registry after that registry is defined.

## 9. Normative References

- [RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <<https://www.rfc-editor.org/rfc/rfc8955>>.
- [RFC8956] Loibl, C., Ed., Raszuk, R., Ed., and S. Hares, Ed., "Dissemination of Flow Specification Rules for IPv6", RFC 8956, DOI 10.17487/RFC8956, December 2020, <<https://www.rfc-editor.org/rfc/rfc8956>>.
- [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>>.
- [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>>.
- [I-D.ietf-idr-flowspec-v2]  
Hares, S., 3rd, D. E. E., Yadlapalli, C., and S. Maduschke, "BGP Flow Specification Version 2", April 2024, <<https://datatracker.ietf.org/doc/draft-ietf-idr-flowspec-v2/04/>>.
- [I-D.hares-idr-fsv2-more-ip-filters]  
Hares, S. and N. Kao, "BGP Flow Specification Version 2 - More IP Filters", n.d., <<https://datatracker.ietf.org/doc/draft-hares-idr-fsv2-more-ip-filters/>>.
- [I-D.khare-idr-bgp-flowspec-payload-match]  
Khare, A., BERGEON, P., Kestur, V., Jalil, L., and K. Kasavchenko, "BGP Flow Specification for Payload Matching", n.d., <<https://datatracker.ietf.org/doc/draft-khare-idr-bgp-flowspec-payload-match/>>.

#### Acknowledgements

We wish to thank Jeffrey Haas and Li Yang for their valuable comments and suggestions on this document. We also wish to thank Rui Xu and Yannan Hu for their contribution in the implementation and validation of the packet content filter software.

#### Authors' Addresses

Yong Cui  
Tsinghua University  
Beijing, 100084  
China  
Email: cuiyong@tsinghua.edu.cn  
URI: <http://www.cuiyong.net/>

Yujia Gao  
Zhongguancun Laboratory  
Beijing, 100094  
China  
Phone: +86-185-1028-7458  
Email: gaoyj@zgclab.edu.cn

Susan Hares  
Hickory Hill Consulting  
7453 Hickory Hill  
Saline, Michigan 48176  
United States of America  
Email: shares@ndzh.com

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

Y. Cui  
Tsinghua University  
Y. Gao  
L. Zhang  
Zhongguancun Laboratory  
20 October 2025

BGP Flow Specification Extension for Feedback Binding  
draft-cui-idr-flowspec-feedback-binding-00

Abstract

This document specifies a BGP Flow Specification extension that conveys per-route feedback binding for a FlowSpec route using the BGP Extended Community attribute. The proposed mechanism introduces a single Feedback Action, encoded as a Generic Transitive Extended Community, which enables downstream routers to report telemetry information or operational events associated with a FlowSpec rule. The Feedback Action carries parameters including a Feedback Identifier (FID), a window exponent (WINC) that defines the periodic aggregation interval, an event flag, and a scope selector to control where feedback is generated. These parameters are attached to the FlowSpec route and are propagated across AS boundaries unchanged. This document focuses on the signaling aspect; a companion document may define how feedback information is exported as part of a network telemetry framework (e.g., leveraging the BGP Monitoring Protocol (BMP)) or equivalent mechanisms to report periodic and event-driven feedback keyed by the FID.

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 23 April 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- 1. Introduction . . . . . 2
- 2. Terminology . . . . . 3
- 3. Definitions and Acronyms . . . . . 3
- 4. Overview . . . . . 4
- 5. Feedback Action Encoding . . . . . 5
  - 5.1. Encoding Format . . . . . 5
  - 5.2. Fields Descriptions . . . . . 6
    - 5.2.1. Feedback Identifier (FID) . . . . . 6
    - 5.2.2. Window Length (WIND) . . . . . 6
    - 5.2.3. Event Flag (E) . . . . . 7
    - 5.2.4. Scope Selector (S) . . . . . 7
    - 5.2.5. Reserved Bits (RESV) . . . . . 7
  - 5.3. Validation Rules . . . . . 7
- 6. Propagation and Usage . . . . . 8
- 7. Security Considerations . . . . . 9
- 8. IANA Considerations . . . . . 9
- 9. Normative References . . . . . 9
- Authors' Addresses . . . . . 10

1. Introduction

BGP Flow Specification (FlowSpec) defines a method for distributing traffic filtering rules using BGP, enabling operators to deploy network-wide traffic management and DDoS mitigation policies in a scalable manner.

The existing versions, FlowSpec (FSv1) RFC8955 [RFC8956] and FlowSpec Version 2 (FSv2) [draft-ietf-idr-flowspec-v2-04], allow the advertisement of flow-matching conditions and actions to drop, rate-limit, or redirect traffic.

These mechanisms are widely used for dynamic DDoS mitigation and policy enforcement across Autonomous Systems (ASes).

However, current FlowSpec deployments lack a standardized mechanism for feedback reporting that allows the originator of a rule to obtain operational visibility such as installation status, traffic hit counts, or rule effectiveness from downstream routers. Without such feedback, operators must rely on out-of-band telemetry or vendor-specific mechanisms, leading to fragmented monitoring and difficulty in verifying the success of mitigation actions, especially in multi-domain environments.

To address this limitation, this document introduces a new Feedback Action that extends the FlowSpec capability to include feedback control and telemetry binding at the route level. By signaling feedback parameters directly within BGP, the originator can request periodic or event-driven reports about the operational state of specific FlowSpec rules. This enhancement enables a closed-loop control paradigm where policy dissemination and enforcement can be continuously monitored and optimized.

This document focuses solely on the signaling aspect of feedback within BGP FlowSpec. The actual transport of feedback data such as telemetry reports or event notifications is out of scope and may be realized using the BGP Monitoring Protocol (BMP) [RFC7854] or other telemetry frameworks compliant with the Network Telemetry Framework.

## 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.

## 3. Definitions and Acronyms

- \* DDoS: Distributed Denial of Service
- \* NLRI: Network Layer Reachability Information
- \* FSv1: Flow Specification Version 1, defined in [RFC8955] and [RFC8956]
- \* FSv2: Flow Specification Version 2 define in [draft-ietf-idr-flowspec-v2-04]
- \* Telemetry: A framework for real-time or streaming collection of network operational data, as defined in [RFC9232].

- \* BMP: BGP Monitoring Protocol [RFC7854], a telemetry protocol used for exporting BGP operational and statistical data.

#### 4. Overview

This specification defines the Feedback Action, a new BGP FlowSpec action encoded as a Generic Transitive Extended Community (Type 0x80).

The Feedback Action allows the originator of a FlowSpec route to convey parameters that instruct downstream routers how and where to generate telemetry feedback for that route.

The Feedback Action carries four parameters compactly encoded in its 6-octet Value field:

- \* Feedback Identifier (FID) â\200\223 A unique identifier that associates telemetry reports with a specific FlowSpec rule.
- \* Window Exponent (WINC) â\200\223 Defines the aggregation interval as  $2^{\text{WINC}}$  seconds for periodic reporting.
- \* Event Flag (E) â\200\223 Controls whether feedback is periodic (00) or event-only (01).
- \* Scope (S) â\200\223 Specifies the feedback domain: Global, Inter-AS, or Intra-AS.

When attached to a FlowSpec NLRI, the Feedback Action expresses the originatorâ\200\231s intent for feedback generation.

It is transitive, ensuring that all capable routers along the propagation path can interpret and act upon the feedback instruction, while non-supporting routers transparently forward it unchanged to maintain compatibility.

This signaling mechanism does not define the feedback transport itself.

Actual telemetry or event reports may be exported through BMP or other protocols aligned with [RFC9232], allowing integration into existing network telemetry infrastructures without altering FlowSpec semantics.

In summary, the Feedback Action augments FlowSpec with a lightweight, interoperable feedback control mechanism that enables closed-loop telemetry, enhances operational visibility, and supports both single-domain and multi-domain DDoS defense deployments with minimal protocol overhead. This extension is suitable for both FSv1 and FSv2.

5. Feedback Action Encoding

The Feedback Action is conveyed using a single BGP Generic Transitive Extended Community, attached to the FlowSpec NLRI.

This action encodes a compact set of parameters that define the feedback reporting behavior for a specific FlowSpec route, including a Feedback Identifier (FID), a Window Exponent (WINC), an Event Flag (E), and a Scope Selector (S).

The action MUST include at least the FID, WIND, and Event flag for a valid binding; the Scope Selector is OPTIONAL. If multiple communities with the same tag are present, receivers SHOULD use the first one and ignore duplicates.

5.1. Encoding Format

The Feedback Action uses the standard IPv4 Extended Community encoding format, as illustrated below:

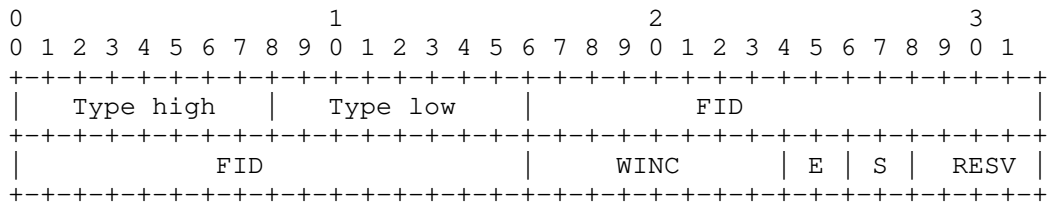


Figure 1: Feedback Action Encoding format

- \* Type high (1 octet): 0x80 indicates a Generic Transitive Extended Community as defined in [RFC4360].
- \* Type low (1 octet): TBD (Feedback Action Sub-Type) value to be assigned by IANA.
- \* FID (4 octets): A 32-bit Feedback Identifier, unique within the originating AS. It serves as the key for feedback correlation between the FlowSpec rule and its associated telemetry data. A value of zero is invalid and MUST be ignored.
- \* WINC (1 octet): Window Exponent, defining the periodic aggregation window as 2^WINC seconds. Valid range: 0-31 (corresponding to 1s to approximately 24 days). Values above 31 are reserved and MUST be ignored.
- \* E (2 bits): Event flag, specifying the feedback triggering mode:
  - 00 Periodic feedback enabled.

- 01  $\hat{\text{a}}\backslash 200\backslash 224$  Event-only feedback (e.g., on rule install, remove, or error).
- 10 and 11  $\hat{\text{a}}\backslash 200\backslash 224$  Reserved for future use (MUST be set to zero on transmission and ignored on receipt).
- \* S (2 bits): Scope selector, specifying where feedback reports are generated:
  - 00  $\hat{\text{a}}\backslash 200\backslash 224$  Global (default; feedback from all capable receivers).
  - 01  $\hat{\text{a}}\backslash 200\backslash 224$  Inter-AS (only from downstream ASes).
  - 10  $\hat{\text{a}}\backslash 200\backslash 224$  Intra-AS (only within the same AS).
  - 11  $\hat{\text{a}}\backslash 200\backslash 224$  Reserved.
- \* \*RESV (4 bits):\* Reserved for future extensions.  
MUST be set to zero by the sender and ignored by the receiver.

## 5.2. Fields Descriptions

### 5.2.1. Feedback Identifier (FID)

The FID uniquely identifies the feedback stream corresponding to the FlowSpec route.

The combination of (Originator ASN, FID) forms a globally unique key for feedback correlation.

This identifier allows the originator to distinguish multiple FlowSpec rules and their respective telemetry results.

Routers that do not recognize this field MUST forward the Extended Community unchanged.

### 5.2.2. Window Length (WIND)

The WIND parameter determines the frequency of periodic feedback reports.

The interval is expressed as an exponent base 2, providing scalability from second to multi-day granularity.

Receivers SHOULD align their reporting schedule to the nearest integer multiple of  $2^{\text{WIND}}$  seconds to maintain synchronization.

### 5.2.3. Event Flag (E)

The Event Flag specifies whether the feedback should be triggered periodically or only upon discrete events. When E=01, routers generate reports only when specific control-plane or data-plane events occur, such as rule installation, withdrawal, update failure, or error detection. This allows efficient on-demand visibility without unnecessary periodic reporting overhead.

### 5.2.4. Scope Selector (S)

The Scope Selector defines the domain from which feedback reports are generated. This enables fine-grained control over where telemetry is collected:

- \* Global (00): Feedback is expected from all capable routers across AS boundaries.
- \* Inter-AS (01): Feedback is limited to routers in downstream ASes.
- \* Intra-AS (10): Feedback is generated only within the same administrative domain.

If a receiver does not match the specified scope, it MAY silently ignore the Feedback Action.

### 5.2.5. Reserved Bits (RESV)

The RESV field (4 bits) is reserved for future extensions. It MUST be set to zero on transmission and ignored on receipt.

## 5.3. Validation Rules

Receivers MUST validate the Feedback Action upon reception to ensure it conforms to the expected encoding and operational constraints. The following validation rules apply:

- \* The Feedback Action MUST be attached to a valid FlowSpec NLRI.
- \* The FID field MUST be non-zero; a value of zero indicates an invalid binding and the attribute MUST be ignored.
- \* The WINC (Window Exponent) field MUST NOT exceed 31. Values above this limit are considered invalid and MUST be ignored.
- \* Reserved or undefined values in the E (Event Flag), S (Scope Selector), or RESV bits MUST be set to zero on transmission and MUST be ignored on receipt.

- \* The total length of the Extended Community attribute MUST conform to the standard 8-octet format defined in [RFC4360]. Any deviation from this format MUST cause the attribute to be ignored.
- \* Malformed attributes or decoding errors MUST NOT result in session reset.  
Such attributes SHOULD be logged locally for operational visibility and SHOULD be propagated unchanged to preserve transitivity.
- \* If multiple Feedback Actions are present for the same NLRI, only the first instance SHOULD be processed; subsequent duplicates SHOULD be ignored.

A Feedback Action that fails any of the above validation checks MUST be silently discarded and MUST NOT affect normal FlowSpec rule installation or propagation.

## 6. Propagation and Usage

The Feedback Action is attached to the BGP UPDATE message that carries the FlowSpec NLRI and MUST be propagated unchanged across all BGP peers, including across AS boundaries.  
This ensures consistent feedback signaling while preserving the original semantics and reachability of the FlowSpec route.

Routers that support feedback reporting SHOULD generate telemetry or event reports according to the parameters conveyed in the attribute. When the local scope matches the value of `_S_` and the Event Flag indicates periodic or event-driven reporting, feedback reports SHOULD be generated accordingly.  
The absence of feedback reports MUST NOT be interpreted as an error, and the FlowSpec rule remains valid for traffic enforcement.

If the attribute cannot be parsed or fails validation, it MUST be silently ignored and MUST NOT trigger a BGP session reset. Such attributes SHOULD be logged locally for operational visibility and MUST be propagated unchanged to preserve transitivity. Routers MUST NOT alter or regenerate the Feedback Action during re-advertisement.

This mechanism provides a lightweight and interoperable means of achieving closed-loop telemetry for FlowSpec deployments, enabling standardized in-band feedback signaling while maintaining full backward compatibility with existing BGP implementations.

7. Security Considerations

This extension inherits the security properties of BGP FlowSpec and Large Communities. Potential issues include:

- \* Resource exhaustion: Receivers SHOULD rate-limit feedback generation and ignore excessive requests.
- \* Privacy: Feedback may reveal traffic patterns; the companion telemetry document SHOULD recommend encryption.
- \* Amplification: Malicious bindings could trigger unwanted reports; originators SHOULD authenticate telemetry receivers.

Operators SHOULD filter invalid or unauthorized communities at AS borders using ingress/egress policies.

8. IANA Considerations

This document requests IANA to assign a new Sub-Type called "feedback action" under the "BGP Extended Communities" registry:

Type	Sub-Type	Name	Reference
TBD	TBD	Feedback Action	This document

Table 1

9. Normative References

[RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <<https://www.rfc-editor.org/rfc/rfc8955>>.

[RFC8956] Loibl, C., Ed., Raszuk, R., Ed., and S. Hares, Ed., "Dissemination of Flow Specification Rules for IPv6", RFC 8956, DOI 10.17487/RFC8956, December 2020, <<https://www.rfc-editor.org/rfc/rfc8956>>.

[RFC9117] Uttaro, J., Alcaide, J., Filsfils, C., Smith, D., and P. Mohapatra, "Revised Validation Procedure for BGP Flow Specifications", RFC 9117, DOI 10.17487/RFC9117, August 2021, <<https://www.rfc-editor.org/rfc/rfc9117>>.

- [RFC8092] Heitz, J., Ed., Snijders, J., Ed., Patel, K., Bagdonas, I., and N. Hilliard, "BGP Large Communities Attribute", RFC 8092, DOI 10.17487/RFC8092, February 2017, <<https://www.rfc-editor.org/rfc/rfc8092>>.
- [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>>.
- [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>>.
- [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>>.
- [draft-ietf-idr-flowspec-v2-04]  
Hares, S., 3rd, D. E. E., Yadlapalli, C., and S. Maduschke, "BGP Flow Specification Version 2", April 2024, <<https://datatracker.ietf.org/doc/draft-ietf-idr-flowspec-v2/04/>>.

#### Authors' Addresses

Yong Cui  
Tsinghua University  
Beijing, 100084  
China  
Email: [cuiyong@tsinghua.edu.cn](mailto:cuiyong@tsinghua.edu.cn)  
URI: <http://www.cuiyong.net/>

Yujia Gao  
Zhongguancun Laboratory  
Beijing, 100094  
China  
Phone: +86-185-1028-7458  
Email: [gaoyj@zgclab.edu.cn](mailto:gaoyj@zgclab.edu.cn)

Lei Zhang  
Zhongguancun Laboratory  
Beijing, 100094  
China  
Email: [zhanglei@zgclab.edu.cn](mailto:zhanglei@zgclab.edu.cn)

IDR Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 7 July 2026

X. He  
A. Wang  
China Telecom  
W. Cheng  
China Mobile  
J. Dong  
Huawei  
X. Min  
ZTE Corp.  
3 January 2026

BGP Extensions to Enable BGP FlowSpec based IFIT  
draft-he-idr-bgp-flowspec-ifit-02

Abstract

Border Gateway Protocol (BGP) Flow Specification (FlowSpec) is an extension to BGP that supports the dissemination of traffic flow specifications and resulting actions to be taken on packets in a specified flow. In-situ Flow Information Telemetry (IFIT) denotes a family of flow-oriented on-path telemetry techniques, which can provide high-precision flow insight and real-time network issue notification. This document defines BGP extensions to distribute BGP FlowSpec based traffic filtering carrying IFIT information. So IFIT behavior can be applied to the specified flow automatically.

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 7 July 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- 1. Introduction . . . . . 3
- 2. Conventions . . . . . 4
  - 2.1. Requirements Language . . . . . 4
  - 2.2. Terminology . . . . . 4
- 3. IFIT Attribute . . . . . 4
- 4. IFIT Attribute Sub-TLVs . . . . . 6
  - 4.1. IOAM Sub-TLVs . . . . . 6
    - 4.1.1. IOAM Pre-allocated Trace Option Sub-TLV . . . . . 6
    - 4.1.2. IOAM Incremental Trace Option Sub-TLV . . . . . 7
    - 4.1.3. IOAM Directly Export (DEX) Option Sub-TLV . . . . . 7
    - 4.1.4. IOAM Edge-to-Edge Option Sub-TLV . . . . . 9
  - 4.2. AltMark Sub-TLVs . . . . . 9
    - 4.2.1. Alternate Marking Sub-TLV . . . . . 9
    - 4.2.2. Enhanced Alternate Marking Sub-TLV . . . . . 10
- 5. Traffic Sampling Action . . . . . 11
  - 5.1. Traffic Sampling Extended Community . . . . . 12
- 6. BGP FlowSpec Operations with IFIT Attributes . . . . . 12
- 7. Validation Procedure and Error Handling . . . . . 13
- 8. IANA Considerations . . . . . 14
  - 8.1. IFIT Attribute Type Code . . . . . 14
  - 8.2. IFIT Type . . . . . 14
  - 8.3. IFIT Attribute Sub-TLVs . . . . . 15
    - 8.3.1. IOAM Type Sub-TLVs . . . . . 15
    - 8.3.2. AltMark Type Sub-TLVs . . . . . 15
  - 8.4. Traffic Sampling Extended Community . . . . . 15
- 9. Security Considerations . . . . . 16
- 10. References . . . . . 16
  - 10.1. Normative References . . . . . 16
  - 10.2. Informative References . . . . . 18
- Authors' Addresses . . . . . 18

## 1. Introduction

Border Gateway Protocol (BGP) Flow Specification defined in [RFC8955] and [RFC8956] (FlowSpec) is an extension to BGP that supports the dissemination of traffic flow specifications and resulting actions to be taken on packets in a specified flow. It leverages the BGP Control Plane to simplify the distribution of ACLs (Access Control Lists). Using the Flow Specification extension, new filter rules can be injected to all BGP peers simultaneously without changing router configuration.

BGP Flow Specification [RFC8955] and [RFC8956] define some BGP Network Layer Reachability Information (NLRI) formats used to distribute traffic flow specification rules. The NLRI for (AFI=1, SAFI=133) specifies IPv4 unicast filtering and the NLRI for (AFI=1, SAFI=134) specifies IPv4 BGP/MPLS VPN filtering [RFC7432]. The NLRI for (AFI=2, SAFI=133) specifies IPv6 unicast filtering and the NLRI for (AFI=2, SAFI=134) specifies IPv6 BGP/MPLS VPN filtering. The Flow Specification match part defined in [RFC8955]and[RFC8956]include L3/L4 information like IPv4/6 source/destination prefix, protocol, ports.

In-situ Flow Information Telemetry (IFIT) denotes a family of flow-oriented on-path telemetry techniques, which can provide high-precision flow insight and real-time network issue notification. In particular, IFIT refers to network OAM (Operations, Administration, and Maintenance) data plane on-path telemetry techniques, including In-situ OAM (IOAM) [RFC9197] and Alternate Marking [RFC9341]. It can provide flow information on the entire forwarding path on a per-packet basis in real time.

With the evolution of IP carried networks towards the intent-based and autonomous networks, flexible deployment of IFIT based on network dynamics and service requirements is getting a must. [I-D.draft-ietf-idr-sr-policy-ifit] defines BGP extensions to distribute SR policies carrying IFIT information so that IFIT behavior can be enabled automatically when the SR policy is applied. IFIT Attributes Sub-TLV is encoded in the Tunnel Encapsulation Attribute (23) defined in [RFC9012] using a new Tunnel-Type called SR Policy Type with codepoint 15. Once the IFIT attributes are signalled, if a packet arrives at the headend and, based on the types of steering described in [RFC9256], it may get steered into an SR Policy where IFIT methods are applied. However, in this way, IFIT is only applicable to SR policy environment. On the other hand, it cannot leverage the BGP FlowSpec to automatically configure traffic flow filtering to steer a packet flow into a valid SR Policy.

This document defines the BGP extensions to distribute BGP FlowSpec based traffic filtering together with IFIT information. So the IFIT behavior can be applied to the specified flow automatically. In this way, IFIT is automatically enabled and running.

## 2. Conventions

### 2.1. 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.

### 2.2. Terminology

Abbreviations used in this document:

ACL: Access Control List

AFI: Address Family Identifier

AS: Autonomous System

DEX: Direct Exporting

IFIT: In-situ Flow Information Telemetry

IOAM: In situ Operation, Administration, and Maintenance

NLRI: Network Layer Reachability Information

OAM: Operation, Administration, and Maintenance

SAFI: Subsequent Address Family Identifier

## 3. IFIT Attribute

IFIT attribute is an optional non-transitive BGP path attribute. IANA is requested to allocate the reserved value as the type code of the attribute in the "BGP Path Attributes" registry [IANA-BGP-PARAMS]. The attribute is composed of a set of Type-Length-Value (TLV) encodings. Each TLV contains information corresponding to a particular IFIT type. An IFIT TLV is structured as shown in Figure 1.

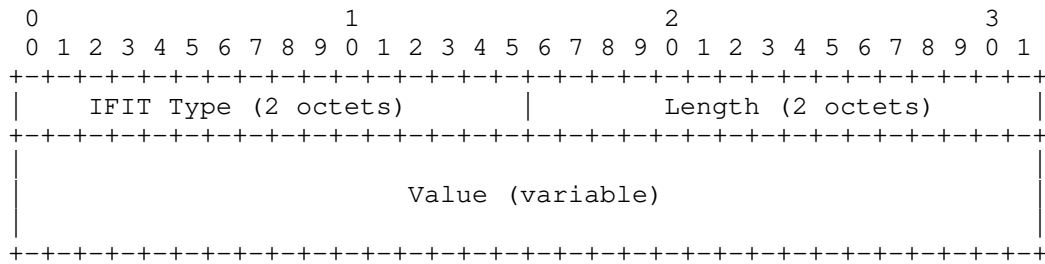


Figure 1: IFIT TLV

IFIT Type (2 octets): Identifies a type of IFIT. This document defines two types of IFIT as follows.

- \* When the IFIT Type is 1, it is the IOAM Type [RFC9197], [RFC9326].
- \* When the IFIT Type is 2, it is the AltMark Type [RFC9343].

Length (2 octets): The total number of octets of the Value field.

Value (variable): Comprised of one or multiple sub-TLVs.

Each sub-TLV consists of three fields: A 1-octet type, a 1-octet length, and zero or more octets of value. A sub-TLV is structured as shown in Figure 2.

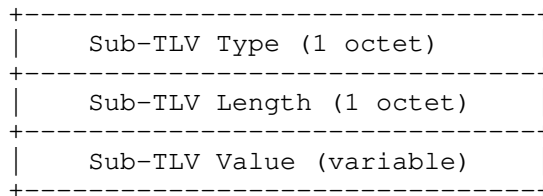


Figure 2: IFIT Sub-TLV

Sub-TLV Type (1 octet): Each sub-TLV type defines a certain OAM option about the IFIT TLV that contains this sub-TLV.

Sub-TLV Length (1 octet): The total number of octets of the Sub-TLV Value field.

Sub-TLV Value (variable): Encoding of the Value field depends on the sub-TLV type. The following subsections define the encoding in detail.

4. IFIT Attribute Sub-TLVs

This section specifies a number of sub-TLVs. These sub-TLVs can be included in two TLVs of the IFIT attribute.

For IFIT Type 1, namely the IOAM Type, four sub-TLVs are defined in this document as follows:

- \* IOAM Pre-allocated Trace Option [RFC9197] Sub-TLV, Type=1.
- \* IOAM Incremental Trace Option [RFC9197] Sub-TLV, Type=2.
- \* IOAM Directly Export (DEX) Option [RFC9326] Sub-TLV, Type=3.
- \* IOAM Edge-to-Edge Option [RFC9197] Sub-TLV, Type=4.

For IFIT Type 2, namely the AltMark Type, two sub-TLVs are defined in this document as follows:

- \* Alternate Marking [RFC9343] Sub-TLV, Type=1.
- \* Enhanced Alternate Marking sub-TLV, Type=2.

4.1. IOAM Sub-TLVs

IOAM Sub-TLVs include four sub-TLVs, and every sub-TLV structure is defined in the following subsections.

4.1.1. IOAM Pre-allocated Trace Option Sub-TLV

The IOAM tracing data is expected to be collected at every node that a packet traverses to ensure visibility into the entire path a packet takes within an IOAM domain. The preallocated tracing option will create pre-allocated space for each node to populate its information. The structure of IOAM pre-allocated trace option sub-TLV is defined as follows:

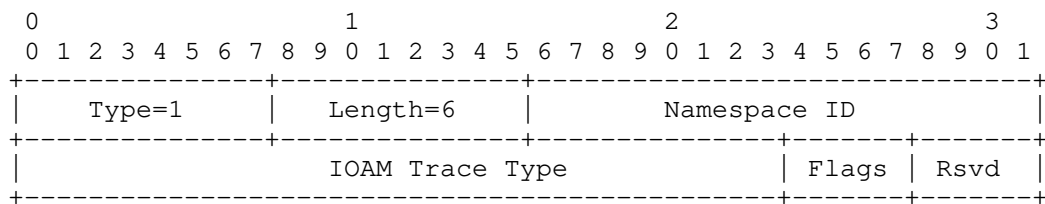


Figure 3: IOAM Pre-allocated Trace Option Sub-TLV

Type: 1 (to be assigned by IANA).

Length: 6, the total number of octets of the Sub-TLV Value field.

Namespace ID: A 16-bit identifier of an IOAM-Namespace. The definition is described in section 4.4 of [RFC9197].

IOAM Trace Type: A 24-bit identifier which specifies which data types are used in the node data list. The definition is described in section 4.4 of [RFC9197].

Flags: A 4-bit field. The definition is described in [RFC9322] and section 4.4 of [RFC9197].

Rsvd: A 4-bit field reserved for further usage. It MUST be zero and ignored on receipt.

#### 4.1.2. IOAM Incremental Trace Option Sub-TLV

The incremental tracing option contains a variable node data fields where each node allocates and pushes its node data immediately following the option header. The structure of IOAM incremental trace option sub-TLV is defined as follows:

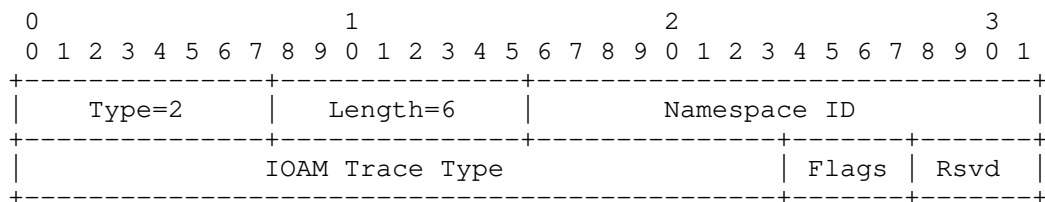


Figure 4: IOAM Incremental Trace Option Sub-TLV

Type: 2 (to be assigned by IANA).

Length: 6, the total number of octets of the Sub-TLV Value field.

All the other fields definitions are the same as the pre-allocated trace option sub-TLV in section 4.1.1.

#### 4.1.3. IOAM Directly Export (DEX) Option Sub-TLV

IOAM DEX option is used as a trigger for IOAM data to be directly exported to a collector without being pushed into in-flight data packets. The structure of IOAM DEX sub-TLV is defined as follows:

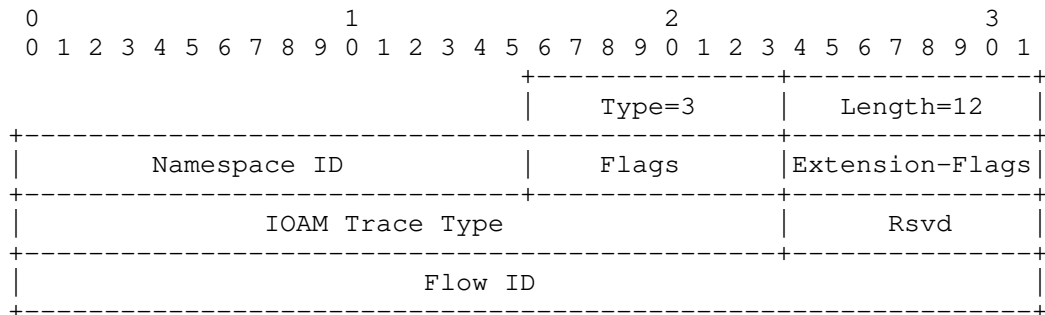


Figure 5: IOAM DEX Option Sub-TLV

Type: 3 (to be assigned by IANA).

Length: 12, the total number of octets of the Sub-TLV Value field.

Namespace ID: A 16-bit identifier of an IOAM-namespace. The definition is described in section 4.4 of [RFC9197].

Flags: A 8-bit field. The definition is described in section 3.2 of [RFC9326].

Extension-Flags: A 8-bit field. The definition is described in section 3.2 of [RFC9326]. Every bit in the Extension-Flag field that is set to 1 indicates the existence of a corresponding optional 4-octet field. Bit 0 (the most significant bit) is defined as Flow ID and bit 1 as Sequence Number. Flow ID may be uniquely assigned by the collector. In this document, when bit 1 is set to 1, Sequence Number MUST be sequentially assigned by the headend device (encapsulating node).

IOAM Trace Type: A 24-bit identifier which specifies which data types are used in the node data list. The definition is described in section 4.4 of [RFC9197].

Rsvd: A 4-bit field reserved for further usage. It MUST be zero and ignored on receipt.

Flow ID: A 32-bit flow identifier. The definition is described in section 3.2 of [RFC9326]. Flow ID may be uniquely assigned by the collector.

4.1.4. IOAM Edge-to-Edge Option Sub-TLV

The IOAM edge-to-edge option is to carry data that is added by the IOAM encapsulating node and interpreted by IOAM decapsulating node. The structure of IOAM edge-to-edge option sub-TLV is defined as follows:

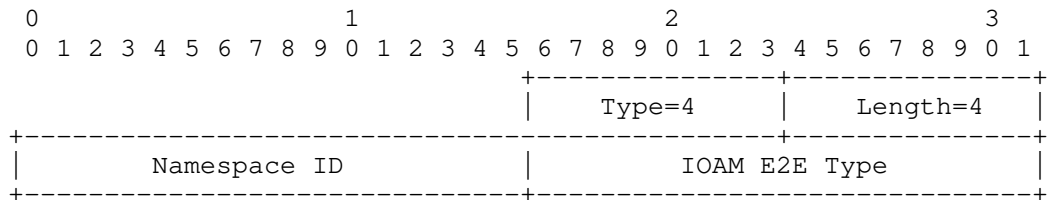


Figure 6: IOAM Edge-to-Edge Option Sub-TLV

Type: 4 (to be assigned by IANA).

Length: 4, the total number of octets of the Sub-TLV Value field.

Namespace ID: A 16-bit identifier of an IOAM-namespace. The definition is described in section 4.4 of [RFC9197].

IOAM E2E Type: A 16-bit identifier which specifies which data types are used in the E2E option data. The definition is described in section 4.6 of [RFC9197].

4.2. AltMark Sub-TLVs

AltMark Sub-TLVs include two sub-TLVs, and every sub-TLV structure is defined in the following subsections.

4.2.1. Alternate Marking Sub-TLV

The structure of Alternate Marking sub-TLV is defined as follows:

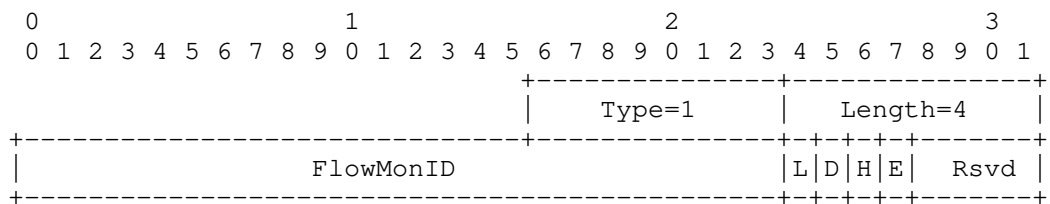


Figure 7: Alternate Marking Sub-TLV

Type: 1 (to be assigned by IANA).

Length: 4, the total number of octets of the Sub-TLV Value field.

FlowMonID: A 20-bit identifier to uniquely identify a monitored flow within the measurement domain. The definition is described in section 5.3 of [RFC9343].

L: 1-bit Loss flag set to 1 indicating Packet Loss Measurement as described in Section 5.1 of [RFC9343].

D: 1-bit Delay flag set to 1 indicating Packet Delay Measurement as described in Section 5.2 of [RFC9343].

H: 1-bit flag set to 1 indicating that the measurement is Hop-by-Hop.

E: 1-bit flag set to 1 indicating that the measurement is End-to-End.

Rsvd: 4-bit field reserved for further usage. It MUST be zero and ignored on receipt.

4.2.2. Enhanced Alternate Marking Sub-TLV

The structure of Enhanced Alternate Marking sub-TLV is defined as follows:

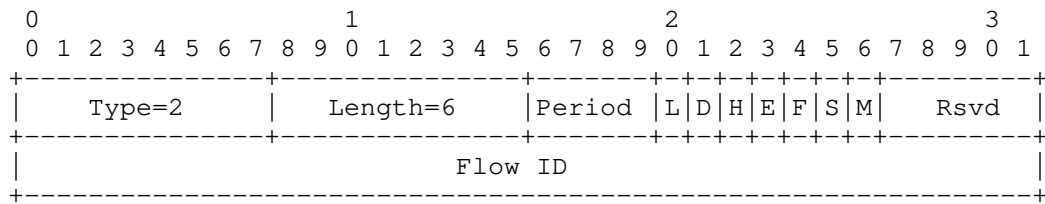


Figure 8: Enhanced Alternate Marking Sub-TLV

Type: 2 (to be assigned by IANA).

Length: 6, the total number of octets of the Sub-TLV Value field.

Period: 4-bit field used for encoding at most 16 measurement periods. The definition of its value and the corresponding measurement period is out of this document.

L: 1-bit Loss flag set to 1 indicating Packet Loss Measurement as described in Section 5.1 of [RFC9343].

D: 1-bit Delay flag set to 1 indicating Delay Measurement as described in Section 5.2 of [RFC9343].

H: 1-bit flag set to 1 indicating that the measurement is Hop-by-Hop.

E: 1-bit flag set to 1 indicating that the measurement is End-to-End.

F: 1-bit flag set to 1 indicating 32-bit Flow ID, which uniquely identify a monitored flow within the measurement domain. The definition and usage is described in section 7 of [I-D.draft-he-ippm-ioam-dex-extensions-incorporating-am-03]. In the centralized way, Flow ID is uniquely assigned by the controller; in the distributed way, Flow ID is locally assigned by the headend device (encapsulating node).

S: 1-bit flag set to 1 indicating an optional 32-bit Sequence Number, starting from 0 and incremented by 1 for each packet from the same flow at the encapsulating node. The field is set at the encapsulating node and exported to the receiving entity by the forwarding nodes. The Sequence Number, when combined with the Flow ID, provides a convenient approach to correlate the exported data from the same user packet. In this document, when bit S is set to 1, Sequence Number MUST be sequentially assigned by the headend device (encapsulating node).

M: 1-bit flag set to 1 indicating an optional 32-bit Measurement Period Number (MPN), starting from 0 and incremented by 1 for the specified flow with the same Flow ID. The field is set at the encapsulating node and exported to the receiving entity by the forwarding nodes. The MPN, when combined with the Flow ID, provides a convenient approach to correlate the exported data of the same flow during the same measurement period from multiple nodes. In this document, when bit M is set to 1, MPN MUST be sequentially assigned by the headend device (encapsulating node).

Rsvd: 5-bit field reserved for further usage. It MUST be zero and ignored on receipt.

## 5. Traffic Sampling Action

IFIT may be applied on all the traffic flow or a subset of the traffic. For the IOAM Type, take IOAM trace monitoring as example, when an IOAM encapsulating node incorporates the IOAM Pre-allocated Trace Option type (passport mode) or the DEX Option type (postcard mode) into all packets of the user traffic it forwards, more bandwidth and processing resources are required. So it is appropriate for an IOAM encapsulating node to apply the IOAM functionality to the selected subset of the traffic. But for the AltMark Type, it is more preferable for an encapsulating node to color all the traffic of interest it forwards, not a subset of the traffic, thus the fidelity of performance measurement for user

traffic flow (e.g., packet loss, delay and jitter) can be ensured.

This document defines a new Traffic Sampling Action that it standardizes as BGP Extended Communities [RFC4360].

5.1. Traffic Sampling Extended Community

The structure of the traffic sampling Extended Community is defined as follows:

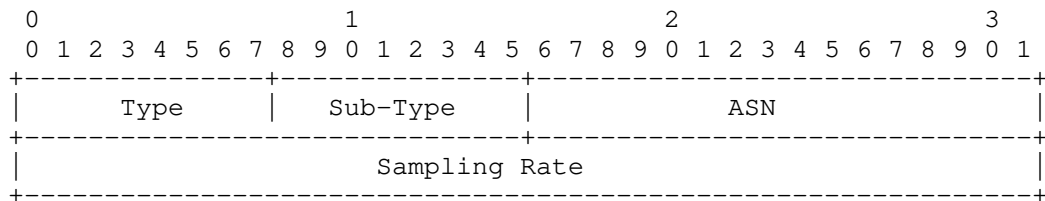


Figure 9: Traffic Sampling Extended Community

Type: 1-octet, BGP Transitive Extended Community Type, set to 0x80.

Sub-Type: 1-octet, TBA (to be assigned by IANA).

ASN: 2-octet AS number, which can be assigned from a 2-octet AS number. When a 4-octet AS number is locally present, the 2 least significant octets of such an AS number can be used. This value is purely informational and SHOULD NOT be interpreted by the implementation.

Sampling Rate: 4-octet float, which carries the sampling rate information in IEEE floating point [IEEE.754.1985] format. A sampling rate of 0 should result on no packet for the particular flow to be applied to IFIT, and a sampling rate of 100% should result on all traffic for the particular flow to be applied to IFIT. On encoding, the sampling rate MUST NOT be negative.

6. BGP FlowSpec Operations with IFIT Attributes

A Flow Specification is an n-tuple consisting of several matching criteria that can be applied to IP traffic flow. A given IP packet is said to match the defined Flow Specification if it matches all the specified criteria. This n-tuple is encoded into a BGP NLRI. Flow Specifications can be seen as more specific routing entries to a unicast prefix, and the routing system can take advantage of the ACL (Access Control List) capabilities in the router's forwarding path.

Generally, in operator network, the centralized controller determines the particular user traffic to be monitored according to service requirements; at the same time, the centralized controller also need to determine th IFIT type applied to the specified traffic flow. Based on BGP FlowSpec, the centralized controller sends the BGP FlowSpec update message to the headend device (i.e,. IOAM encapsulating node), carrying NLRI and IFIT attributes along with the traffic sampling Extended Community(if present). The BGP FlowSpec update message is used to instruct the headend device to perform IFIT automatic configuration on the monitored traffic flows. The headend device automatically generates ACLs according to the received traffic filter rules, and encapsulates IFIT for the incoming specified traffic flow packets, achieving the automated configuration for the monitored traffic flows.

Once the centralized controller determines to terminate monitoring the specified traffic flow, it can withdraw the corresponding NRLI routes, indicating that the headend device will remove the ACLs related to the traffic filter rules and stop encapsulating IFIT for the incoming specified traffic flow packets.

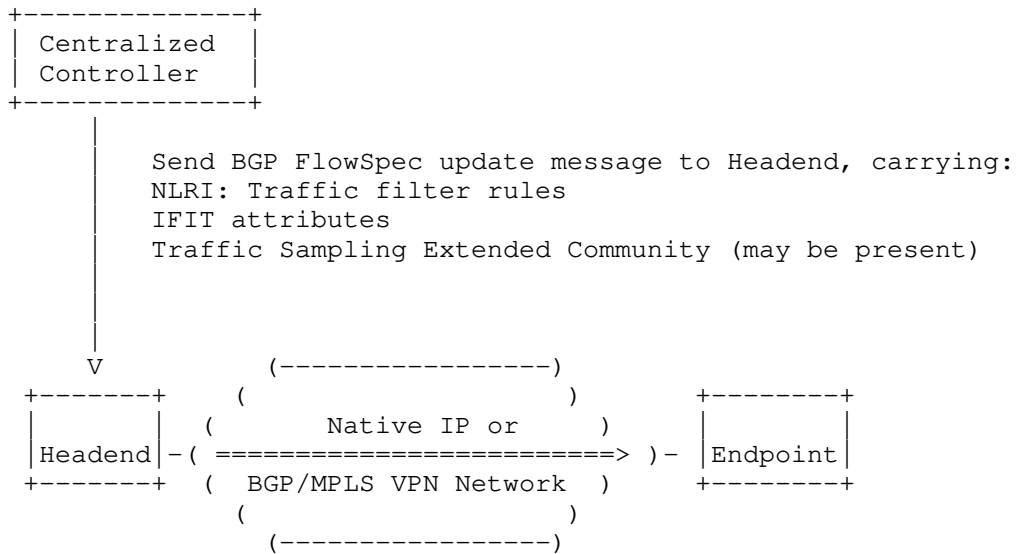


Figure 10: IFIT applied to the specified traffic flow

## 7. Validation Procedure and Error Handling

The validation procedure is the same as specified in Section 6 of [RFC8955] and Section 5 of [RFC8955].

Additionally, The IFIT Attribute MUST be attached to the BGP Update and MUST have an IFIT Type TLV set to the IOAM Type (1) or the AltMark Type (2).

When the IFIT Type TLV includes any sub-TLV that is unrecognized or unsupported, the update SHOULD NOT be considered usable. An implementation MAY provide an option for ignoring unsupported sub-TLVs.

A router that receives an BGP update that is not valid according to these criteria MUST treat the update as malformed.

The validation of the TLVs/sub-TLVs introduced in this document and defined in their respective sub-sections of Section 4 MUST be performed to determine if they are malformed or invalid. In case of any error detected, either at the attribute or its TLV/sub-TLV level, the "treat-as-withdraw" strategy MUST be applied. This is because a BGP Flowspec update without a valid IFIT Attribute (comprising of all valid TLVs/sub-TLVs) is not usable.

A BGP Flowspec update that is determined to be not valid, and therefore malformed, MUST be handled by the "treat-as-withdraw" strategy.

An implementation SHOULD log any errors found during the above validation for further analysis.

## 8. IANA Considerations

### 8.1. IFIT Attribute Type Code

IANA is requested to allocate the reserved value as the type code of the attribute in the "BGP Path Attributes" registry [IANA-BGP-PARAMS].

Type Code	Description	Reference
TBA	IFIT Attribute	This document

### 8.2. IFIT Type

IANA is requested to create a IFIT Type registry. IANA is requested to allocate the following values as the type code of the IFIT Attribute TLVs. Unassigned Type values will be assigned on a First Come First Served (FCFS) basis.

Value	Description	Reference
1	IOAM Type	This document
2	AltMark Type	This document

### 8.3. IFIT Attribute Sub-TLVs

#### 8.3.1. IOAM Type Sub-TLVs

IANA is requested to create a IOAM Type registry. IANA is requested to allocate the following values as the type code of the IOAM Type Sub-TLVs. Unassigned Type values will be assigned on a First Come First Served (FCFS) basis.

Value	Description	Reference
1	IOAM Pre-allocated Trace Option Sub-TLV	This document
2	IOAM Incremental Trace Option Sub-TLV	This document
3	IOAM Directly Export (DEX) Option Sub-TLV	This document
4	IOAM Edge-to-Edge Option Sub-TLV	This document

#### 8.3.2. AltMark Type Sub-TLVs

IANA is requested to create an AltMark Type registry. IANA is requested to allocate the following values as the type code of the AltMark Type Sub-TLVs. Unassigned Type values will be assigned on a First Come First Served (FCFS) basis.

Value	Description	Reference
1	Alternate Marking Sub-TLV	This document
2	Enhanced Alternate Marking sub-TLV	This document

### 8.4. Traffic Sampling Extended Community

IANA is requested to allocate the reserved value as the Sub-type code of Traffic Sampling Extended Community in the registry entitled "Generic Transitive Experimental Use Extended Community Sub-Types".

Type Value	Sub-Type Value	Description	Reference
0x80	TBA	Traffic Sampling	This document

## 9. Security Considerations

The security mechanisms of the base BGP security model apply to the extensions described in this document as well. See the Security Considerations section of [RFC4271] for a discussion of BGP security.

The BGP extensions specified in this document enable IFIT within an controlled domain, as defined in [RFC9378] and [I.D.draft-ietf-ippm-alt-mark-deployment]. IFIT operates within a controlled domain and its security considerations also apply to BGP sessions when carrying IFIT information. The IFIT configurations distributed by BGP are expected to be used entirely within this trusted IFIT domain which comprises a single AS or multiple ASes/ domains within a single provider network. Therefore, precaution is necessary to ensure that the IFIT information advertised via BGP sessions is limited to nodes in a secure manner within this trusted IFIT domain.

Flow Specification BGP speakers (e.g., the centralized controller) also need to be cautious for sending BGP updates. For example, sending updates at a high rate, or generating a high number of Flow Specifications may stress the receiving systems (the Headend devices), or exceed their capabilities.

Another major concern is that enabling IFIT on all the traffic flows may have some impact on network forwarding performance, thus the traffic sampling action is needed for protecting network resources. In particular, setting the appropriate traffic sampling rate for the IOAM trace monitoring is also necessary.

## 10. References

### 10.1. Normative References

- [IEEE.754.1985] IEEE, "IEEE Standard for Binary Floating-Point Arithmetic", IEEE ANSI/IEEE 754-1985, DOI 10.1109/IEEESTD.1985.82928, 5 April 2019, <<https://ieeexplore.ieee.org/document/30711>>.
- [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>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.

- [RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <<https://www.rfc-editor.org/info/rfc4360>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<https://www.rfc-editor.org/info/rfc4760>>.
- [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/info/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/info/rfc8174>>.
- [RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <<https://www.rfc-editor.org/info/rfc8955>>.
- [RFC8956] Loibl, C., Ed., Raszuk, R., Ed., and S. Hares, Ed., "Dissemination of Flow Specification Rules for IPv6", RFC 8956, DOI 10.17487/RFC8956, December 2020, <<https://www.rfc-editor.org/info/rfc8956>>.
- [RFC9197] Brockners, F., Ed., Bhandari, S., Ed., and T. Mizrahi, Ed., "Data Fields for In Situ Operations, Administration, and Maintenance (IOAM)", RFC 9197, DOI 10.17487/RFC9197, May 2022, <<https://www.rfc-editor.org/info/rfc9197>>.
- [RFC9326] Song, H., Gafni, B., Brockners, F., Bhandari, S., and T. Mizrahi, "In Situ Operations, Administration, and Maintenance (IOAM) Direct Exporting", RFC 9326, DOI 10.17487/RFC9326, November 2022, <<https://www.rfc-editor.org/info/rfc9326>>.
- [RFC9341] Fioccola, G., Ed., Cociglio, M., Mirsky, G., Mizrahi, T., and T. Zhou, "Alternate-Marking Method", RFC 9341, DOI 10.17487/RFC9341, December 2022, <<https://www.rfc-editor.org/info/rfc9341>>.
- [RFC9343] Fioccola, G., Zhou, T., Cociglio, M., Qin, F., and R. Pang, "IPv6 Application of the Alternate-Marking Method", RFC 9343, DOI 10.17487/RFC9343, December 2022, <<https://www.rfc-editor.org/info/rfc9343>>.

## 10.2. Informative References

- [I-D.he-ippm-ioam-dex-extensions-incorporating-am]  
hexiaoming, X., Brockners, F., Song, H., Fioccola, G., and A. Wang, "IOAM Direct Exporting (DEX) Option Extensions for Incorporating the Alternate-Marking Method", Work in Progress, Internet-Draft, draft-he-ippm-ioam-dex-extensions-incorporating-am-03, 13 November 2025, <<https://datatracker.ietf.org/doc/html/draft-he-ippm-ioam-dex-extensions-incorporating-am-03>>.
- [I-D.he-ippm-ioam-extensions-incorporating-am]  
hexiaoming, X., Min, X., Brockners, F., Fioccola, G., and C. Xie, "IOAM Trace Option Extensions for Incorporating the Alternate-Marking Method", Work in Progress, Internet-Draft, draft-he-ippm-ioam-extensions-incorporating-am-05, 13 November 2025, <<https://datatracker.ietf.org/doc/html/draft-he-ippm-ioam-extensions-incorporating-am-05>>.
- [I-D.ietf-idr-sr-policy-ifit]  
Qin, F., Yuan, H., Yang, S., Zhou, T., and G. Fioccola, "BGP SR Policy Extensions to Enable IFIT", Work in Progress, Internet-Draft, draft-ietf-idr-sr-policy-ifit-11, 15 October 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-sr-policy-ifit-11>>.
- [I-D.ietf-ippm-alt-mark-deployment]  
Fioccola, G., Zhu, K., Graf, T., Nilo, M., and L. Zhang, "Alternate Marking Deployment Framework", Work in Progress, Internet-Draft, draft-ietf-ippm-alt-mark-deployment-04, 29 August 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-ippm-alt-mark-deployment-04>>.
- [RFC9378] Brockners, F., Ed., Bhandari, S., Ed., Bernier, D., and T. Mizrahi, Ed., "In Situ Operations, Administration, and Maintenance (IOAM) Deployment", RFC 9378, DOI 10.17487/RFC9378, April 2023, <<https://www.rfc-editor.org/info/rfc9378>>.

## Authors' Addresses

Xiaoming He  
China Telecom  
Email: [hexm4@chinatelecom.cn](mailto:hexm4@chinatelecom.cn)

Aijun Wang  
China Telecom  
Email: wangaj3@chinatelecom.cn

Weiqiang Cheng  
China Mobile  
Email: chengweiqiang@chinamobile.com

Jie Dong  
Huawei  
Email: jie.dong@huawei.com

Xiao Min  
ZTE Corp.  
Email: xiao.min2@zte.com.cn

IDR Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 10 November 2026

S. Hares  
Hickory Hill Consulting  
D. Eastlake  
Independent  
J. Dong  
Huawei Technologies  
C. Yadlapalli  
ATT  
S. Maduscke  
Verizon  
J. Haas  
HPE  
9 May 2026

BGP Flow Specification Version 2 - for Basic IP  
draft-ietf-idr-fsv2-ip-basic-06

Abstract

BGP flow specification version 1 (FSv1), defined in RFC 8955, RFC 8956, and RFC 9117, describes the distribution of traffic filter policy (traffic filters and actions) distributed via BGP. During the deployment of BGP FSv1 a number of issues were detected, so version 2 of the BGP flow specification (FSv2) protocol addresses these issues. In order to provide a clear demarcation between FSv1 and FSv2, a different NLRI encapsulates FSv2.

The IDR WG requires two implementation. Early feedback on implementations of FSv2 indicate that FSv2 has a correct design direction, but that breaking FSv2 into a progression of documents would aid deployment of the draft (basic, adding more filters, and adding more actions). This document specifies the basic FSv2 NLRI with user ordering of filters added to FSv1 IP Filters and FSv2 actions.

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 10 November 2026.

#### Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

#### Table of Contents

1.	Introduction . . . . .	3
1.1.	Why Flow Specification v2 . . . . .	4
1.2.	Definitions and Acronyms . . . . .	6
1.3.	Requirements Language . . . . .	7
2.	Flow Specification Version 2 Primer . . . . .	8
2.1.	Flow Specification v1 (FSv1) SAFIs . . . . .	8
2.2.	Transition to FSv2 . . . . .	9
2.3.	FSv2 Overview . . . . .	10
3.	FSv2 NLRI Formats and Actions . . . . .	11
3.1.	FSv2 NLRI Format . . . . .	12
3.1.1.	FSv2 Filter Family TLVs . . . . .	13
3.1.2.	FSv2 Filter Component TLVs . . . . .	14
3.2.	FSv2 Dependencies . . . . .	16
3.3.	Ordering of TLVs within the FSv2 NLRI . . . . .	17
3.4.	Partial Deployments . . . . .	17
4.	FSv2 IP Basic Filters (Filter Family Type TBD) . . . . .	17
4.1.	Operators . . . . .	17
4.1.1.	Numeric Operator (numeric_op) . . . . .	18
4.1.2.	Bitmask Operator (bitmask_op) . . . . .	19
4.2.	FSv2 IP Basic Filter Components . . . . .	20
4.3.	FSv2 Flow Specification Order of IP Basic Components . . . . .	21
4.4.	FSv2 Components for IP Basic TLVs . . . . .	21
4.4.1.	IP Destination Prefix (component type = 10) . . . . .	21
4.4.2.	IP Source Prefix (type = 20) . . . . .	22
4.4.3.	IP Protocol/IPv6 Upper Layer Protocol (type = 30) . . . . .	23

4.4.4.	Port (type = 40)	23
4.4.5.	Destination Port (type = 50)	24
4.4.6.	Source Port (type = 60)	24
4.4.7.	ICMP Type (type = 70)	25
4.4.8.	ICMP Code (type = 80)	25
4.4.9.	TCP Flags (type = 90)	26
4.4.10.	Packet length (type = 100)	27
4.4.11.	DSCP (Differentiated Services Code Point) (type = 110)	27
4.4.12.	Fragment (type = 120)	27
4.4.13.	Flow Label (type = 130), AFI=2 only	28
4.5.	FSv2 Traffic Filtering Actions for FSv2 IP Basic	29
4.5.1.	Categories of FSv2 Actions and their Interactions	29
4.5.2.	FSv2 Extended Community Actions	30
4.5.3.	Failure of an FS-EC Action	32
4.5.4.	Unknown FSv2-EC Actions	33
4.5.5.	Action Chain Ordering FSv2-EC (ACO) (optional)	34
5.	Validation and Ordering of FS Routes	35
5.1.	Validating FSv2 NLRI	35
5.2.	Validation of FSv2 BGP Routes	37
5.2.1.	AFI/SAFIs Used For Validation	37
5.2.2.	FSv2 Route Validation Procedure	38
5.2.3.	Validation of Flow Specification Actions for FSv2 for IP Basic	39
6.	Traffic Filtering	39
6.1.	Ordering of FSv2 Flow Specifications	40
6.2.	Installation of FSv2 Filters	42
6.3.	Ordering of FS filters for BGP Peers which support FSv1 and FSv2	42
7.	Scalability and Aspirations for FSv2	43
8.	Optional Security Additions	44
8.1.	BGP FSv2 with ROA	44
9.	IANA Considerations	45
9.1.	Flow Specification V2 SAFIs	45
9.2.	Generic Transitive Extended Community	46
9.3.	FSv2 IP Filters Component Types	46
9.4.	FSv2 Filter Component Types	47
10.	Security Considerations	48
11.	References	49
11.1.	Normative References	49
11.2.	Informative References	52
	Authors' Addresses	53

## 1. Introduction

Version 2 of BGP flow specification was original defined in [fsv2] (BGP FSv2).

FSv2 is an update to BGP Flow specification version 1 (BGP FSv1). BGP FSv1 as defined in [FSv1], [FSv1-IPv6], and [RFC9117] specified 2 SAFIs (133, 134) to be used with IPv4 AFI (AFI = 1) and IPv6 AFI (AFI=2).

The initial BGP FSv2 specification had the correct direction, but it contained more than the early implementers desired. The implmenters desired a progression of documents with smaller incremental changes: Basic FSv2, adding more filters, and adding more actions.

This draft provides the basic FSv2 framework specification for transmitting user-ordered IP filters in the FSV2 NLRI and associating Flow Spec actions by transmitting Flow Spec Extended Communities (FS-EC) with the FSv2 NLRI. If a filter match links to a single FS-EC action, the single action succeeds or fails. If a filter match links to mutiple actions, there is a potential for interactions. Section 4.5.1 discusses how to analyze the interaction by categories and solutions to issues with multiple FSv2-EC actions interacting. A complete solution requires the BGP Community Container Attribute see [I-D.ietf-idr-wide-bgp-communities]) with FSv2 Container defined in the [fsv2-more-ip-filters].

This document defines 2 new SAFIs, TBD1 and TBD2, for FSv2 to be used with 5 AFIs: 1, 2, 6, 25, and 31. FSv2 implementations do not require all 10 combinations of FSv2 AFI/SAFIs to be implemented. An implementation is required to implement only one these AFI/SAFIs to be compliant. For example, a compliant implementation might only define the FSv2 NLRI for IPv4 for IP forwarding (AFI=1, SAFI=TBD1).

FSv1 and FSv2 use different AFI/SAFIs to send their respective flow specification filters. This permits FSv1 and FSv2 to be coexist with each other in a "ships in the night" deployment.

The remainder of Section 1 provides background on why the FSv2 was necessary to fix problems with FSv1. Section 2 contains a primer on FSv2. Section 3 contains the BGP encoding rules for FSv2. Section 5 describes how to validate and order FSv2 NLRI. The remaining sections discuss scalability, optional security additions, security considerations, and IANA considerations.

### 1.1. Why Flow Specification v2

Modern IP routers have the capability to forward traffic and to classify, shape, rate limit, filter, or redirect packets based on administratively defined policies. These traffic policy mechanisms allow the operator to define match rules that operate on multiple fields within header of an IP data packet. The traffic policy allows actions to be taken upon a match to be associated with each match

rule. These rules can be more widely defined as "event-condition-action" (ECA) rules where the event is always the reception of a packet.

BGP ([RFC4271]) flow specification version 1 (FSv1) as defined by [FSv1], [FSv1-IPv6], and [RFC9117] specifies the distribution of traffic filter policy (traffic filters and actions) via BGP to BGP peers, both IBGP and EBGP. The traffic filter is applied when packets are received on a router with the flow specification function enabled.

Multiple deployed applications currently use BGP FSv1 to distribute traffic filters. These applications include:

- \* Mitigation of Denial of Service traffic (DoS).
- \* Traffic filtering in BGP/MPLS VPNS.
- \* Centralized traffic control for networks utilizing SDN control of router firewall functions.
- \* Classifiers for insertion into a SFC.
- \* Filters for SRv6 (segment routing v6).

During the deployment of FSv1, the following issues were noted:

- \* FSv1 NLRI components did not use TLV encoding, which inhibited defining new component types. (The format was type-value, missing a length field.)
- \* FSv1 rules did not have the ability to be ordered by the operator. Instead, only the protocol-defined rule ordering was permitted.
- \* When conflicting outcomes for rule actions was present, the operator was unable to influence their ordering.
- \* When multiple and conflicting rule actions were present, the operator couldn't define their order when some actions could not be implemented on the receiving router.

Networks currently address these issues by constraining deployments or using topology/deployment specific workarounds.

FSv1 is a critical component of deployed applications. Therefore, this specification defines how FSv2 will interact with BGP peers that support combinations FSv1 and FSv2. It is expected that a transition to FSv2 will occur over time as new applications require features enabled by FSv2.

## 1.2. Definitions and Acronyms

**AFI:**

Address Family Identifier [RFC4760]

**AS:**

Autonomous System

**BGP session ephemeral state:**

State which does not survive the loss of BGP peer session.

**BGP Community Path Attribute:**

BGP Community Path attribute with a FS TLV defined by [fsv2-more-ip-filters]

**Configuration state:**

State which persist across a reboot of software module within a routing system or a reboot of a hardware routing device.

**CPA:**

BGP Community Path Attribute.

**DDoS:**

Distributed Denial of Service.

**Ephemeral state:**

State which does not survive the reboot of a software module, or a hardware reboot. Ephemeral state can be ephemeral configuration state or operational state.

**Extended Community:**

BGP Path Attribute defined by [RFC4360].

**FS:**

Flow Specification (either v1 or v2).

**FSv1:**

Flow Specification version 1 [FSv1] [FSv1-IPv6].

**FSv2:**

Flow Specification version 2 (this document and its extensions).

FS-CPA:  
Flow Specification Actions defined in Community Path Attribute.

FS-EC:  
FS related Extended Community with FS actions.

FSv1-EC:  
FSv1 Extended Community with FS Actions supported by FSv1.

FSv2-EC:  
FSv2 Extended Community with FS Actions supported by FSv2.

NETCONF:  
The Network Configuration Protocol [RFC6241].

NLRI:  
Network Layer Reachability Information [RFC4271] [RFC4760]. The "destination" portion of a Flowspec route carried in a BGP UPDATE message.

RESTCONF:  
The RESTCONF Protocol [RFC8040].

RIB:  
Routing Information Base.

ROA:  
Route Origin Authentication [RFC9582].

RR:  
Route Reflector [RFC4456].

SAFI:  
Subsequent Address Family Identifier [RFC4760].

SFC:  
Service Function Chaining [RFC7665].

### 1.3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD 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. Flow Specification Version 2 Primer

A BGP Flow Specification (v1 or v2) is an n-tuple containing one or more match criteria that can be applied to data-plane traffic. The exact traffic match depends on the FSv2 AFI/SAFI.

Flows Specification routes carried in BGP UPDATEs may carry BGP Path Attributes that have additional match or action consequences. This includes, but is not limited to: Extended Communities [RFC4360] and Community Container Path attributes [I-D.ietf-idr-wide-bgp-communities].

Flow Specification NLRI for a given AFI/SAFI is used as they key for Flow Specification routes in the BGP RIBs. Flow Specification routes that are selected for the Loc-RIB are then associated with a given set of semantics which are application dependent. Standard BGP policy mechanisms for BGP routes are applicable to Flow Specification routes, including AS\_PATH and community filtering.

This FSv2 for basic IP forwarding specification only requires the use of Extended Communities to associate FS actions with FSv2 filters found in FSv2 NLRI.

FSv2 features implementing multiple actions with user ordering of actions or dependencies between actions requires the BGP Community Attribute [I-D.ietf-idr-wide-bgp-communities] with a FSv2 Component as defined in [fsv2-more-ip-filters].

Network operators can control the propagation of Flow Specification BGP routes by enabling or disabling the exchange of routes for a particular AFI/SAFI pair on a particular peering session. BGP policy mechanisms, including [RFC1997] scoping communities, can also be used. Thus, Flow Specification routes may be distributed to only a portion of a BGP deployment.

### 2.1. Flow Specification v1 (FSv1) SAFIs

The FSv1 NLRI defined in [FSv1] and [FSv1-IPv6] includes 13 match conditions encoded for the following AFI/SAFIs:

- \* IPv4 traffic: AFI:1, SAFI:133
- \* IPv6 Traffic: AFI:2, SAFI:133
- \* BGP/MPLS IPv4 VPN: AFI:1, SAFI: 134
- \* BGP/MPLS IPv6 VPN: AFI:2, SAFI: 134

FSv1 match conditions are ordered by component type in ascending order. The ordering within a component type is defined by that component's definition.

The Flow Specification actions standardized by [FSv1] and [FSv1-IPv6] are:

- \* accept packet (default),
- \* traffic rate limiting by bps (0x6),
- \* traffic-action: sample, or terminate rule (0x7),
- \* redirect traffic to VPN by route target(0x8),
- \* traffic marking (DSCP) (0x9), and
- \* traffic rate limiting by pps (0xC)

A SFC action [RFC9015] defines a redirection of a data flow to an entry point into a specific SFP (Service Function Path).

Other Extended Community actions have been proposed in IDR, but have not completed the standardization process.

## 2.2. Transition to FSv2

This specification defines AFI/SAFIs to support Flow Specification version 2 for IPv4, IPv6, Layer 2, IPv4 VPNs, IPv6 VPNs, Layer 2 VPNs (L2VPN), Service Function Chaining (SFC), and SFC VPNs:

- \* IPv4 traffic: AFI=1, SAFI=TBD1,
- \* IPv6 traffic: AFI=2, SAFI=TBD1,
- \* L2: AFI=6, SAFI=TBD1 (defined in [I-D.ietf-idr-flowspec-l2vpn]),
- \* BGP/MPLS IPv4 VPN: AFI=1, SAFI=TBD2,
- \* BGP/MPLS IPv6 VPN: AFI=2, SAFI=TBD2,
- \* BGP/MPLS L2VPN: AFI=25, SAFI=TBD2 (defined in [I-D.ietf-idr-flowspec-l2vpn]),
- \* SFC: AFI=31, SAFI=TBD1,
- \* SFC VPN: AFI=31, SAFI=TBD2

One question asked by developers is what AFI/SAFI is required for FSv2 IP Basic compliance. BGP negotiates support for each AFI/SAFI, so FSv2 IP Basic support for non-VPN could be as little as FSv2 for IPv4 forwarding (AFI/SAFI: 1/TBD1),

The IDR specification for L2 VPN traffic was specified in [I-D.ietf-idr-flowspec-l2vpn]. An IDR specification for tunneled traffic is in [I-D.ietf-idr-flowspec-nvo3]. Both of these drafts were targeted for FSv1, but the WG decided to require these to FSv2 TLV formats.

### 2.3. FSv2 Overview

FSv2 allows the user to order the flow specification rules and the actions associated with a rule. Each FSv2 rule may have one or more match conditions and one or more associated actions.

FSv2 operates in a ships-in-the night model with FSv1. This permits operators to manage the interaction of FSv2 and FSv1 via configuration.

The basic principles regarding the ordering and installation of flow specification filter rules are:

1. In the absence of a matching filter for the traffic, that traffic is permitted. That is, the default is permit. Implementations MAY implement a default reject behavior by configuration.
2. FSv2 filter rules are processed prior to FSv1 rules. FSv1 NLRI are processed according to the procedures defined in [FSv1] and [FSv1-IPv6]. FSv2 filter rules thus have a better precedence vs. FSv1.
3. FSv2 filter rules are ordered based on user-specified order, carried in each FSv2 NLRI. Numerically smaller user-specified order values have better precedence than larger values.
4. For rules with the same user-specified order, the filter rules are then ordered by FSv2 component type and then rules for each component type.
5. FSv2 filter rules can carry actions. These actions can be encoded via one or more FSv2 Extended Communities, or within the FSv2 Action Community Container.

Some FSv2 Extended Communities may not be understood by every FSv2 implementation. Since they are encoded as [RFC4360] Extended Communities, they are propagated with the BGP routes regardless of whether they are understood based on the particular Extended Community's transitivity.

When FSv2 Extended Communities are understood, they have precedence and interaction rules governing the actions they encode. (See XXX JMH TODO)

The FSv2 Action Community Container defines its own rules governing FSv2 actions. See that document (XXX JMH TODO) for additional details.

6. FSv2 filter match and action criteria may be considered "optional". For match, the FSv2 NLRI encoding carries a per-component flag set by the operator or implementation that marks that match component as optional or mandatory. For actions, FSv2 Extended Communities will document whether they are considered optional or mandatory as part of their definition. The optionality of FSv2 Action Community Containers is defined in its defining document.

If a mandatory match component or action component cannot be locally implemented, the flowspec rule is marked as ineligible to be installed.

7. FSv2 filter rules carry a "Dependency" value in the FSv2 NLRI. When this value is non-zero, this value associates multiple received FSv2 filters with each other. If a FSv2 filter rule is ineligible to be installed due to an inability to implement a mandatory match or action component, all other filters carrying the same dependency value will be made ineligible for installation. See Section 3.2 for more details.

### 3. FSv2 NLRI Formats and Actions

BGP Flow Specifications are encoded in BGP NLRI as an ordered list of TLVs of "filter families", where each filter family consists of an ordered list of TLVs of "filter components" for that family. Filter families are groupings of related filtering functionality, typically at the same network layer. Filter components match specific network elements for a filter family.

Each FSv2 NLRI has a default sort order, documented in section TODO. This sort order determines the order of installation for the Flow Specification in the BGP speaker. Operators MAY override this default ordering by causing the FSv2 User Order field to be set to a non-zero value.

Sets of FSv2 NLRI might share fate with each other. In the event that a Flow Specification is unable to be installed by the BGP speaker, dependent Flow Specifications MUST NOT also be installed, even if they are otherwise valid. These dependencies are encoded in the Dependent Filters Chain field of a FSv2 Flow Specification.

FSv2 is carried in BGP using standard [RFC4760] multiprotocol extensions. FSv2 supports NRLI with formats for following AFIs:

- \* IPv4 (AFI = 1)
- \* IPv6 (AFI = 2)
- \* L2 (AFI = 6)
- \* L2VPN (AFI=25)
- \* SFC (AFI=31)

These AFIs will be paired with the following SAFIs:

- \* TBD1 (Flow Spec Version 2)
- \* TBD2 (Flow Spec Version 2 for VPNs)

A compliant FSv2 implementation only has to implement one AFI/SAFI pair out of the full list of NRLIs. For example, a compliant FSv2 implementation might only implement IPv4 FSv2 (AFI=1, SAFI=TBD1).

FSv2 NLRI are encoded in BGP UPDATES using the MP\_REACH\_NLRI and MP\_UNREACH\_NLRI attributes defined in [RFC4760]. When advertising FSv2 NLRI, the length of the Next-Hop Network Address MUST be set to 0. Upon reception, the MP\_REACH\_NLRI "Network Address of NextHop" field MUST be ignored.

### 3.1. FSv2 NLRI Format

FSv2 Flow Specifications are encoded as an ordered list of TLVs of filter families. FSv2 filter families are typically associated with match criteria for a given networking layer; for example, 802.2 Layer 2, MPLS, IPv4/IPv6, Segment Routing, etc.

The AFI/SAFI NLRI for BGP Flow Specification version 2 (FSv2) has the format:

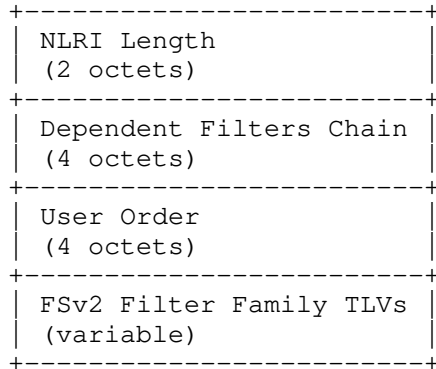


Figure 1: FSv2 NLRI Format

Where:

- \* NLRI Length: Length of the NLRI field in octets excluding the NLRI Length field. The minimum NLRI Length is 8 (Dependent Filter Chain + User Order).
- \* Dependent Filters Chain (DFC): A 32-bit unsigned integer in network byte order. When non-zero, the Dependent Filters Chain value is used to associate multiple NLRI together that share dependencies. See Section 3.2 for further information on its use.
- \* User Order: A 32-bit unsigned integer in network byte order. FSv2 rules with a lower User Order value have a better precedence for filter ordering.
- \* FSv2 Filter Family TLVs: An ordered list of TLVs of FSv2 filter families. The encoding of these filter families is documented in the next section.

### 3.1.1. FSv2 Filter Family TLVs

Each each FSv2 Filter Family TLV has the format:

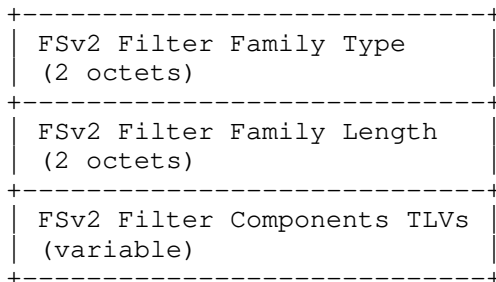


Figure 2: FSv2 Filter Family TLV Format

Where:

- \* FSv2 Filter Family Type: A 16-bit unsigned integer in network byte order defining the FSv2 filter that is carried in this TLV. For sorting purposes, lower value FSv2 Filter Types have a better precedence than higher values.
- \* FSv2 Filter Family Length: Length of the FSv2 Filter Components TLVs in octets.

### 3.1.2. FSv2 Filter Component TLVs

Each each FSv2 Filter Component TLV has the Format:

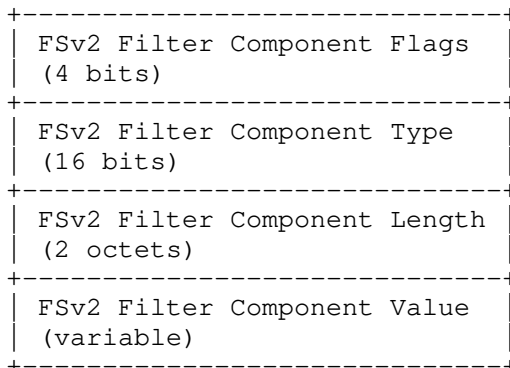


Figure 3: FSv2 Filter Component TLV Format

Where:

- \* The FSv2 Filter Component Flags are defined as:

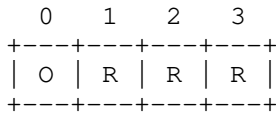


Figure 4: FSv2 Filter Component Flags Format

The fields of the FSv2 Filter Component Flags are defined as:

- O - Optional: When 0, the FSv2 filter-type-specific filter component is mandatory and MUST be supported by the local implementation. Otherwise, when 1, the component is considered "optional". When the component is mandatory and is not supported, the FSv2 filter rule is considered "invalid" for validation purposes.
- R - Reserved: When not otherwise re-defined in a later document, this bit MUST be set to zero when sent and SHOULD be ignored on reception.
- \* FSv2 Filter Component Type: A 12-bit unsigned integer in network byte order defining the match component for a given FSv2 filter type. For sorting purposes, lower value FSv2 Filter Component Types have a better precedence than higher values.

This document defines the following FSv2 Filter Component Types. The definition of the type-specific filter components may be defined in other documents:

- 0 - Reserved
- 50 - L2 Traffic fules
- 100 - MPLS traffic rules
- 150 - SFC Traffic rules
- 200 - Tunneled traffic
- 256 - IP Basic Filter Rules (bit 1 of high bit)
- 280 - IP Extended Filter Rules

- \* FSv2 Filter Component Length: A 16-bit unsigned integer in network byte order containing the length of the FSv2 Filter Components Value field.
- \* FSv2 Filter Component Value: Each FSv2 filter type will define one or more FSv2 filter-type-specific filter components. See each FSv2 filter-type's specification for a component's definition.

FSv2 implementations MUST pass valid filter TLVs even if the implementation does not support these installation of these a particular type of filter rules.

This specification only defines operation of the IP Basic Filter Rules that all FSv2 must support.

### 3.2. FSv2 Dependencies

Flow Specifications are implemented using ordered terms. The sorting rules for flow specification routes is intended to, by default, produce a reasonably ordered set of rules for common deployment scenarios.

When the FSv2 rule ordering wouldn't accomplish the operator's intent when deploying FSv2, the User Order field can permit the operator to influence the Flow Specification installation order in a deployment.

When set of Flow Specifications are required to implement an operator's intent and that set of rules has interdependencies, the failure to install a Flow Specification, or part of that specification's actions, may result in incorrect deployment. An example of such a dependency is two rules covering an IP destination, one with a more-specific and one with a less-specific prefix relationship. As an example:

1. match dst=10.1.1.1/8 tcp port=25 then dscp=AF1 and permit
2. match dst=10.0.0.0/8 tcp port=25 then drop

If an implementation couldn't support the DSCP action and failed to install the first rule, SMTP traffic to the host 10.1.1.1 would fail to be delivered due to the second rule's drop action. In other words, these two entries have a dependency.

When an implementation is unable to install a Flow Specification for some reason, that Flow Specification is locally "invalid". In many circumstances, Flow Specifications that do not have dependencies may be installed on a best-effort basis by an implementation. However, in the case of dependent rules, installing some rules selectively but not others can be problematic.

FSv2 defines for each FSv2 NLRI a Dependent Filters Chain (DFC). When the value of DFC is zero (0), no special consideration is given for dependencies. When the value of DFC is non-zero, when a rule is locally considered invalid, all rules sharing the same DFC value are also considered invalid, and not installed.

### 3.3. Ordering of TLVs within the FSv2 NLRI

For NLRI canonicalization purposes, and also to ease processing, all TLVs within the FSv2 NLRI MUST be ordered in a strictly increasing fashion. FSv2 filter types and FSv2 filter-type-specific component types for a given component MUST NOT occur more than once.

See Section 5.1 for further details.

### 3.4. Partial Deployments

Partial deployments can occur for two reasons:

- \* Only a portion of the nodes in a network with FSv2 support installing new FSv2 Filter types with new FSv2 components. Other nodes (such as RRs), check the syntax, but do not handle the semantic meaning.
- \* During upgrades, a portion of the nodes know about a new Filter type with the components, but other nodes do not.

Editor: Are there others?

## 4. FSv2 IP Basic Filters (Filter Family Type TBD)

FSv2 IP Basic filters provide the same functionality as those specified in FSv1 RFCs [FSv1] and [FSv1-IPv6]. The format of those components has been preserved for ease of implementation.

The FSv2 IP Basic filter has been assigned a FSv2 Filter Type value of TBD.

FSv2 IP Basic Filter component types are numbered differently from those in FSv1. FSv2 components have been numbered with gaps to permit future FSv2 IP Basic filter components to be added in between currently specified IP Basic components. This permits a natural default sort order for those new components in implementations.

### 4.1. Operators

Most of the components described below make use of comparison operators. These operators were originally defined in Section 4.2.1 of [FSv1]. They are repeated here for document clarity.

The operators are encoded as a single octet.

## 4.1.1.1. Numeric Operator (numeric\_op)

This operator is encoded as shown in Figure 3-3.

0	1	2	3	4	5	6	7
e	a	len	0	lt	gt	eq	

Figure 5: Numeric Operator (numeric\_op)

e (end-of-list bit): Set in the last {op, value} pair in the list

a (AND bit): If unset, the result of the previous {op, value} pair is logically ORed with the current one. If set, the operation is a logical AND. In the first operator octet of a sequence, it MUST be encoded as unset and MUST be treated as always unset on decoding. The AND operator has higher priority than OR for the purposes of evaluating logical expressions.

len (length): The length of the value field for this operator given as  $(1 \ll \text{len})$ . This encodes 1 (len=00), 2 (len=01), 4 (len=10), and 8 (len=11) octets.

0 MUST be set to 0 on NLRI encoding and MUST be ignored during decoding

lt less-than comparison between data and value

gt: greater-than comparison between data and value

eq: equality between data and value

The bits lt, gt, and eq can be combined to produce common relational operators, such as "less or equal", "greater or equal", and "not equal to", as shown in Table 3-1.

lt	gt	eq	Resulting operation
0	0	0	false (independent of the value)
0	0	1	== (equal)
0	1	0	> (greater than)
0	1	1	<= (greater than or equal)
1	0	0	< (less than)
1	0	1	<= (less than or equal)
1	1	0	!= (not equal value)
1	1	1	true (independent of the value)

Figure 6: Comparison Operation Combinations

4.1.2. Bitmask Operator (bitmask\_op)

This operator is encoded as shown in Figure 3-4.

0	1	2	3	4	5	6	7
e	a	len	0	0	not	m	

Figure 7: Bitmask Operator (bitmask\_op)

Where:

e, a, len (end-of-list bit, AND bit, and length field): Most significant nibble; defined in the Numeric Operator format in section 3-x.

not (NOT bit): If set, logical negation of operation.

m (Match bit): If set, this is a bitwise match operation defined as "(data AND value) == value"; if unset, (data AND value) evaluates to TRUE if any of the bits in the value mask are set in the data.

0 (all 0 bits): MUST be set to 0 on NLRI encoding and MUST be ignored during decoding

#### 4.2. FSv2 IP Basic Filter Components

FSv2 IP Basic Filter Components are encoded in FSv2 Filter Component TLVs as described in Section 3.1.2.

The list of valid Basic IP types, covering the functionality defined in [FSv1] and [FSv1-IPv6] are documented below. Additional IP filters are documented in defined in [I-D.hares-idr-fsv2-more-ip-filters].

Type	Definition
0	Reserved
10	IP Destination Prefix
20	IP Source Prefix
30	IPv4 Protocol / IPv6 Upper Layer Protocol
40	Port
50	Destination Port
60	Source Port
70	ICMPv4 Type / ICMPv6 Type
80	ICMPv4 Code / ICPv6 Code
90	TCP Flags
100	Packet Length
110	DSCP
120	Fragment
130	Flow Label
4095	Reserved

Table 1: FSv2 IP Basic Components

### 4.3. FSv2 Flow Specification Order of IP Basic Components

For Flow Specification ordering purposes, IP Basic Filter components are ordered similar the FSv1 comparison rules documented in Section 5.1 of [FSv1].

The relative order of two Flow Specifications with IP Basic filter family components is determined by comparing their respective family-specific components. The algorithm starts by comparing the lowest component type value of the Flow Specifications. If the types differ, the Flow Specification with lowest numeric type value has higher precedence (and thus will match before) than the Flow Specification that doesn't contain that component type. If the component types are the same, then a type-specific comparison is performed (see below). If the types are equal, the algorithm continues with the next component.

For IP prefix values (IP destination or source prefix), if one of the two prefixes to compare is a more specific prefix of the other, the more specific prefix has higher precedence. Otherwise, the one with the lowest IP value has higher precedence.

For all other component types, unless otherwise specified, the comparison is performed by comparing the component data as a binary string using the memcmp() function as defined by [ISO\_IEC\_9899]. For strings with equal lengths, the lowest string (memcmp) has higher precedence. For strings of different lengths, the common prefix is compared. If the common prefix is not equal, the string with the lowest prefix has higher precedence. If the common prefix is equal, the longest string is considered to have higher precedence than the shorter one.

### 4.4. FSv2 Components for IP Basic TLVs

#### 4.4.1. IP Destination Prefix (component type = 10)

##### 4.4.1.1. IPv4 Destination Prefix (AFI=1)

Encoding: <prefix length (1 octet), prefix (variable)>

Defines the IPv4 destination prefix to match.

\*prefix length:\* Length of the prefix in bits.

\*prefix:\* IPv4 Prefix encoded using [RFC4271] NLRI format.

## 4.4.1.2. IPv6 Destination Prefix (AFI=2)

Encoding: <length (1 octet), offset (1 octet), pattern (variable), padding (variable)>

This defines the IPv6 destination prefix to match. The offset has been defined to allow for flexible matching to portions of an IPv6 address where one is required to skip over the first N bits of the address. (These bits skipped are often indicated as "don't care" bits.) This can be especially useful where part of the IPv6 address consists of an embedded IPv4 address, and matching needs to happen only on the embedded IPv4 address. The encoded pattern contains enough octets for the bits used in matching (length minus offset bits).

- \*length:\* This indicates the N-th most significant bit in the address where bitwise pattern matching stops.
- \*offset:\* This indicates the number of most significant address bits to skip before bitwise pattern matching starts.
- \*pattern:\* This contains the matching pattern. The length of the pattern is defined by the number of bits needed for pattern matching (length minus offset).
- \*padding:\* This contains the minimum number of bits required to pad the component to an octet boundary. Padding bits MUST be 0 on encoding and MUST be ignored on decoding.

If length = 0 and offset = 0, this component matches every address; otherwise, length MUST be in the range offset < length < 129 or the component is malformed.

Note: This Flow Specification component can be represented by the notation ipv6address/length if offset is 0 or ipv6address/offset-length. The ipv6address in this notation is the textual IPv6 representation of the pattern shifted to the right by the number of offset bits.

## 4.4.2. IP Source Prefix (type = 20)

## 4.4.2.1. IPv4 Source Prefix (AFI=1)

Encoding: <prefix length (1 octet), prefix (variable)>

Defines the IPv4 source prefix to match.

- \*prefix length:\* Length of the prefix in bits.
- \*prefix:\* IPv4 Prefix encoded using [RFC4271] NLRI format.

#### 4.4.2.2. IPv6 Source Prefix (AFI=2)

Encoding: <length (1 octet), offset (1 octet), pattern (variable), padding (variable)>

This defines the source prefix to match. The length, offset, pattern, and padding are the same as in Section 4.4.1.2.

#### 4.4.3. IP Protocol/IPv6 Upper Layer Protocol (type = 30)

Encoding: <[numeric\_op, value]+>

##### 4.4.3.1. IPv4 Protocol (AFI=1)

Contains a list of {numeric\_op, value} pairs that are used to match the IP protocol value octet in IPv4 packet header Section 3.1 of [RFC0791].

This component uses the Numeric Operator (numeric\_op) described in Section 4.1.1. Type 30 component values SHOULD be encoded as single octet (numeric\_op len=00).

##### 4.4.3.2. IPv6 Upper Layer Protocol (AFI=2)

This contains a list of {numeric\_op, value} pairs that are used to match the first Next Header value octet in IPv6 packets that is not an extension header and thus indicates that the next item in the packet is the corresponding upper-layer header (see Section 4 of [RFC8200] Section 4).

This component uses the Numeric Operator (numeric\_op) described in Section 4.1.1. Type 30 component values SHOULD be encoded as a single octet (numeric\_op len=00).

Note: While IPv6 allows for more than one Next Header field in the packet, the main goal of the Type 30 Flow Specification component is to match on the first upper-layer IP protocol value. Therefore, the definition is limited to match only on this specific Next Header field in the packet.

#### 4.4.4. Port (type = 40)

Encoding: <[numeric\_op, value]+>

Defines a list of {numeric\_op, value} pairs that match source OR destination TCP/UDP ports (see Section 3.1 of [RFC0793] and the "Format" section of [RFC0768]). This component matches if either the destination port OR the source port of an IP packet matches the value.

This component uses the Numeric Operator (numeric\_op) described in Section 4.1.1. Type 40 component values SHOULD be encoded as 1- or 2-octet quantities (numeric\_op len=00 or len=01).

In case of the presence of the port (destination-port (Section 4.4.5), source-port (Section 4.4.6) component, only TCP or UDP packets can match the entire Flow Specification. The port component, if present, never matches when the packet's IP protocol value is not 6 (TCP) or 17 (UDP), if the packet is fragmented and this is not the first fragment, or if the system is unable to locate the transport header. Different implementations may or may not be able to decode the transport header in the presence of IP options or Encapsulating Security Payload (ESP) NULL [RFC4303] encryption.

Note: This component only matches the first upper layer protocol value in IPv6.

#### 4.4.5. Destination Port (type = 50)

Encoding: <[numeric\_op, value]+>

Defines a list of {numeric\_op, value} pairs used to match the destination port of a TCP or UDP packet (see also Section 3.1 of [RFC0793] and the "Format" section of [RFC0768]).

This component uses the Numeric Operator (numeric\_op) described in Section 4.1.1. Type 50 component values SHOULD be encoded as 1- or 2-octet quantities (numeric\_op len=00 or len=01).

The last paragraph of Section 4.4.4 also applies to this component.

#### 4.4.6. Source Port (type = 60)

Encoding: <[numeric\_op, value]+>

Defines a list of {numeric\_op, value} pairs used to match the source port of a TCP or UDP packet (see also Section 3.1 of [RFC0793] and the "Format" section of [RFC0768]).

This component uses the Numeric Operator (numeric\_op) described in Section 4.1.1. Type 60 component values SHOULD be encoded as 1- or 2-octet quantities (numeric\_op len=00 or len=01).

The last paragraph of Section 4.4.4 also applies to this component.

#### 4.4.7. ICMP Type (type = 70)

Encoding: <[numeric\_op, value]+>

Defines a list of {numeric\_op, value} pairs used to match the type field of an ICMP packet (see also the "Message Formats" section of [RFC0792]).

This component uses the Numeric Operator (numeric\_op) described in Section 4.1.1. Type 70 component values SHOULD be encoded as single octet (numeric\_op len=00).

##### 4.4.7.1. ICMP IPv4 Type (AFI=1)

In case of the presence of the ICMP type component, only ICMP packets can match the entire Flow Specification. The ICMP type component, if present, never matches when the packet's IP protocol value is not 1 (ICMP), if the packet is fragmented and this is not the first fragment, or if the system is unable to locate the transport header. Different implementations may or may not be able to decode the transport header in the presence of IP options or Encapsulating Security Payload (ESP) NULL [RFC4303] encryption.

##### 4.4.7.2. ICMP IPv6 Type (AFI=2)

In case of the presence of the ICMPv6 type component, only ICMPv6 packets can match the entire Flow Specification. The ICMPv6 type component, if present, never matches when the packet's upper-layer IP protocol value is not 58 (ICMPv6), if the packet is fragmented and this is not the first fragment, or if the system is unable to locate the transport header. Different implementations may or may not be able to decode the transport header.

#### 4.4.8. ICMP Code (type = 80)

Encoding: <[numeric\_op, value]+>

##### 4.4.8.1. ICMP Code IPv4 Type (AFI=1)

Defines a list of {numeric\_op, value} pairs used to match the code field of an ICMP packet (see also the "Message Formats" section of [RFC0792]).

This component uses the Numeric Operator (numeric\_op) described in Section 4.1.1. Type 80 component values SHOULD be encoded as single octet (numeric\_op len=00).

In case of the presence of the ICMP code component, only ICMP packets can match the entire Flow Specification. The ICMP code component, if present, never matches when the packet's IP protocol value is not 1 (ICMP), if the packet is fragmented and this is not the first fragment, or if the system is unable to locate the transport header. Different implementations may or may not be able to decode the transport header in the presence of IP options or Encapsulating Security Payload (ESP) NULL [RFC4303] encryption.

#### 4.4.8.2. ICMP Code IPv6 Type (AFI=2)

This defines a list of {numeric\_op, value} pairs used to match the code field of an ICMPv6 packet (see also Section 2.1 of [RFC4443]).

This component uses the Numeric Operator (numeric\_op) described in Section 4.1.1. Type 80 component values SHOULD be encoded as a single octet (numeric\_op len=00).

In case of the presence of the ICMPv6 code component, only ICMPv6 packets can match the entire Flow Specification. The ICMPv6 code component, if present, never matches when the packet's upper-layer IP protocol value is not 58 (ICMPv6), if the packet is fragmented and this is not the first fragment, or if the system is unable to locate the transport header. Different implementations may or may not be able to decode the transport header.

#### 4.4.9. TCP Flags (type = 90)

Encoding: <[bitmask\_op, bitmask]+>

Defines a list of {bitmask\_op, bitmask} pairs used to match TCP control bits (see also Section 3.1 of [RFC0793]).

This component uses the Bitmask Operator (bitmask\_op) described in Section 4.1.2. Type 90 component bitmasks MUST be encoded as 1- or 2-octet bitmask (bitmask\_op len=00 or len=01).

When a single octet (bitmask\_op len=00) is specified, it matches octet 14 of the TCP header (see also Section 3.1 of [RFC0793]), which contains the TCP control bits. When a 2-octet (bitmask\_op len=01) encoding is used, it matches octets 13 and 14 of the TCP header with the data offset (leftmost 4 bits) always treated as 0.

In case of the presence of the TCP flags component, only TCP packets can match the entire Flow Specification. The TCP flags component, if present, never matches when the packet's IP protocol value is not 6 (TCP), if the packet is fragmented and this is not the first fragment, or if the system is unable to locate the transport header.

Different implementations may or may not be able to decode the transport header in the presence of IP options or Encapsulating Security Payload (ESP) NULL [RFC4303] encryption.

#### 4.4.10. Packet length (type = 100)

Encoding: <[numeric\_op, value]+>

Defines a list of {numeric\_op, value} pairs used to match on the total IP packet length (excluding Layer 2 but including IP header).

This component uses the Numeric Operator (numeric\_op) described in Section 4.1.1. Type 100 component values SHOULD be encoded as 1- or 2-octet quantities (numeric\_op len=00 or len=01).

#### 4.4.11. DSCP (Differentiated Services Code Point) (type = 110)

Encoding: <[numeric\_op, value]+>

Defines a list of {numeric\_op, value} pairs used to match the 6-bit DSCP field (see also [RFC2474]).

This component uses the Numeric Operator (numeric\_op) described in Section 4.1.1. Type 110 component values MUST be encoded as single octet (numeric\_op len=00).

The six least significant bits contain the DSCP value. All other bits SHOULD be treated as 0.

#### 4.4.12. Fragment (type = 120)

Encoding: <[bitmask\_op, bitmask]+>

Defines a list of {bitmask\_op, bitmask} pairs used to match specific IP fragments.

This component uses the Bitmask Operator (bitmask\_op) described in Section 4.1.2. Type 120 component bitmask MUST be encoded as single octet bitmask (bitmask\_op len=00).

##### 4.4.12.1. IPv4 Fragment (AFI=1)

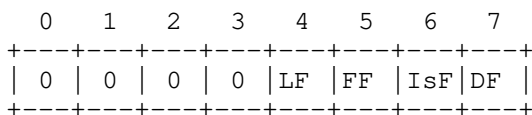


Figure 8: IPv4 Fragment Bitmask Operand

Bitmask values:

\*DF (Don't Fragment):\* match if IP Header Flags Bit-1 (DF) [RFC0791] is 1

\*IsF (Is a fragment other than the first):\* match if the [RFC0791] IP Header Fragment Offset is not 0

\*FF (First Fragment):\* match if the [RFC0791] IP Header Fragment Offset is 0 AND Flags Bit-2 (MF) is 1

\*LF (Last Fragment):\* match if the [RFC0791] IP Header Fragment Offset is not 0 AND Flags Bit-2 (MF) is 0

\*0:\* MUST be set to 0 on NLRI encoding and MUST be ignored during decoding

#### 4.4.12.2. IPv6 Fragment (AFI=2)

0	1	2	3	4	5	6	7
0	0	0	0	LF	FF	IsF	0

Figure 9: IPv6 Fragment Bitmask Operand

Bitmask values:

\*IsF:\* Is a fragment other than the first -- match if IPv6 Fragment Header (Section 4.5 of [RFC8200]) Fragment Offset is not 0

\*FF:\* First fragment -- match if IPv6 Fragment Header (Section 4.5 of [RFC8200]) Fragment Offset is 0 AND M flag is 1

\*LF:\* Last fragment -- match if IPv6 Fragment Header (Section 4.5 of [RFC8200]) Fragment Offset is not 0 AND M flag is 0

\*0:\* MUST be set to 0 on NLRI encoding and MUST be ignored during decoding

#### 4.4.13. Flow Label (type = 130), AFI=2 only

Encoding: <[numeric\_op, value]+>

This contains a list of {numeric\_op, value} pairs that are used to match the 20-bit Flow Label IPv6 header field (Section 3 of [RFC8200]).

This component uses the Numeric Operator (numeric\_op) described in Section 4.1.1. Type 130 component values SHOULD be encoded as 4-octet quantities (numeric\_op len=10).

#### 4.5. FSv2 Traffic Filtering Actions for FSv2 IP Basic

Traffic matching a flow specification filter may have selected `_traffic actions_` applied to it that have various impacts on the matched traffic. FSv2 IP Basic allows flow specification actions to be attached to flow specification routes using BGP Extended Communities (FSv2-EC) encoded using the Extended Community formats [RFC4360] or in the IPv6 Address Specific Extended Community format [RFC5701].

Section 4.5.1 describes the interaction between FS-EC action, and categories of actions. Section 4.5.2 describes the existing FS-EC action formats. Section 4.5.5 defines an optional FS-EC to pass information ordering of categories (user/this standard) and failure action (stop or best effort).

##### 4.5.1. Categories of FSv2 Actions and their Interactions

FSv2-EC actions fall into the following categories:

- \* Further constraint of the match criteria for the traffic in addition to that which is encoded in the NLRI.
- \* Apply traffic shaping mechanisms, such as bps/pps rate limiters.
- \* Change IP packet properties, such as DSCP.
- \* Redirect (change the forwarding) of the traffic. Examples include redirecting to VPN VRFs, or forwarding to tunneled destinations.
- \* Flag the traffic for sampling.
- \* Terminate the evaluation of further flow specification matches in the forwarding plane.

When multiple actions from a given FSv2-EC category are present in a FSv2 route, these actions may `_conflict_`. Conflicting actions result in ambiguity as to what traffic action behavior is applied to traffic matching the flow specification.

FSv2 actions passed in a BGP Community Container Attribute can provide ordering of actions, dependencies, or signal which actions are valid within a category (see [fsv2-more-ip-filters]). However, these features are beyond the Basic FSv2 for IP forwarding and are out of scope for this specification.

#### 4.5.2. FSv2 Extended Community Actions

FSv2 IP Basic uses FSv1 actions and these are referenced in Section 4.5.2.1 and Section 4.5.2.2.

One additional, optional, FSv2 specific FS-EC: the Action Chain Ordering (ACO) Extended Community (ACO-EC), is defined in Section 4.5.5. ACO-EC can carry defaults currently only available by configuration in FSv1.

##### 4.5.2.1. Existing Flow Specification Action Extended Communities

FSv1 defines a set of [RFC4360] encoded extended communities implementing actions also applicable to FSv2 IP Basic match types. They are:

Type/Sub-Type	Description	Short-ID	Reference
0x01/0x0c	Redirect to IP	RD-IP	[redirect-ip]
0x07/0x02	Match Interface set	TA-IS	[interface-set]
0x09/0xxx	Redirect to Indirection ID	RD-IID	[path-redirect]
0x0b/0x00	SFC Reserved	SFC-R	[RFC9015]
0x0b/0x01	SFVC SFIR POOL Identifier	SFIR-PI	[RFC9015]
0x0b/0x02	SFC MPLS label stack Swapping or stacking labels	SFC-MPLS	[RFC9015]
0x80/0x06	Traffic rate limit by bytes	TR-BPS	[FSv1]
0x80/0x07	Traffic Action (sample, terminal)	TA	[FSv1]
0x80/0x08	Redirection to VRF (2-octet AS form)	RD-VRF-AS2	[FSv1]
0x80/0x09	Traffic mark DSCP	TM-DSCP	[FSv1]
0x80/0x0C	Traffic rate limit by packets	TR-BPS	[FSv1]
0x81/0x08	Redirect to VPN (IPv4 form)	RD-VRF-IPv4	[FSv1]
0x81/0x08	Redirect to VPN (4-octet AS form)	RD-VRF-AS4	[FSv1]

Table 2: FSv1 Extended Communities Used by FSv2

Note the Short ID is simply a quick way for this document to reference a particular action.

#### 4.5.2.2. Existing Flow Specification Actions IPv6 Address Specific Extended Communities

FSv1 defines a set of [RFC5701] encoded extended communities implementing actions also applicable to FSv2 IP Basic match types. They are:

Type	Description	Short-ID	Reference
0x000C	FS Redirect to IPv6	RD-IP6	[redirect-ip]
0x000D	FS Redirect to VPN by IPv6 route target	RD-VRF-IPv6	[FSv1-IPv6]

Table 3: FSv1 IPv6 Address Specific Extended Communities Used by FSv2

#### 4.5.3. Failure of an FS-EC Action

Devices implementing flow specification matching and traffic actions may be unable, for whatever reason, to carry out the signaled actions for the matched traffic. Some examples of this inability include:

- \* The action is not implemented in the forwarding plane.
- \* Combinations of non-conflicting actions may not be able to be simultaneously executed due to limitation in the implementation's forwarding plane.

When FS-EC actions known to the implementation are attached to a flow specification route and an action cannot be executed, there are three potential options:

- Option 1: Stop processing additional filters and (optionally) signal failure to the management process.
- Option 2: Continue on processing in "best effort" for the next filters.
- Option 3: Decide between 1 and 2 based on dependencies between filters and actions.

Option 1 and 2 can be signaled by configuration within a Flow Specification implementation.

Option 3 requires the encoding dependency lists in ordered filters and ordered actions. The FSv2 NLRI format has a field to carry filter dependency information, but these functions are beyond the FSv2 Basic IP functions and out of scope for this specification.

Consider an example where three FSv2-EC actions are present on the route: Set the DSCP value, request sampling of the traffic, redirect to a VRF. If the implementation is unable to set the DSCP value:

Option 1 would be to stop processing and not do the other two actions.

Option 2 would be to continue processing and do the other two actions.

Currently, for FSv1, local configuration or implementation behavior determines what happens if one of the actions fails within a set of multiple actions attached to a filter rule.

One option for FSv2 is to pass another FS-EC indicating what the originator expects will happen upon failure of an action.

#### 4.5.4. Unknown FSv2-EC Actions

A flow specification implementation that understands extended communities for a traffic action may not necessarily be able to implement them. Another problematic case for consistent deployment of flow specification within a network is understanding that an implementation may be ignorant of some FSv2-ECs.

FSv2-ECs are carried in the general purpose BGP Extended Community features. The expected behavior for an implementation receiving unknown Extended Communities, depending on configuration and policy, will be to ignore the contents of these communities and propagate them according to the transitivity rules in [RFC4360].

Newly defined FSv2-ECs may be unknown to the implementation, typically as a result of incremental deployment newer flow specification traffic actions. When a network with older implementations receive such newly defined FSv2-ECs, the older implementations are unable to determine that an action has been requested at all. The default behavior thus becomes "best effort" for executing the known FSv2-ECs.

When specifying new FSv2-ECs, operational consideration MUST be given to what the behavior of such ignorant implementations may do to the desired traffic forwarding throughout the FS deployment.

4.5.5. Action Chain Ordering FSv2-EC (ACO) (optional)

Summary: This optional FSv2-EC passes information on what the BGP peer originating the FSv2-EC expects will happen with multiple actions attached to a single filter.

Description: The BGP peer originating multiple FSv2 FS-EC actions attached to FSv2 NLRI (filters) may attach the Action Chain Ordering (ACO) FS-EC to inform BGP Peers receiving the FSv2 information how the originating pair expects action interactions and actions failures will be handled. Two fields are encoded in this FS-EC:

AC-interaction - What happens if two actions are specified in a category, and

AC-Failure - what happens if an action with multiple action set fails.

Encoding: The Generic Transitive encoding is shown in Figure 10 with the field definitions below.

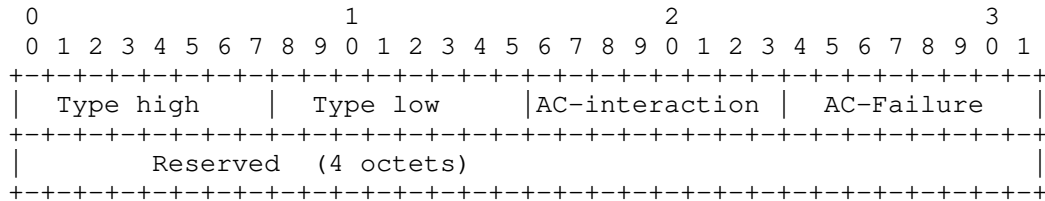


Figure 10: Action Chain Ordering FSv2-EC

where:

\*Type high:\* This 1 octet field has a value of 0x80 For the Generic Transitive EC.

\*Type low:\* This one octet field identifies the ACO-Action. The value is TBD4.

\*AC-interaction:\* This field indicates whether the FS-EC category order is the pre-defined order or an implementation specific order.

\* 0 (default): Do not install actions with two actions per category.

\* 1 (local config): Allow under local configuration.

\*AC-failure:\* 1 octet byte that determines the action on failure. Actions may succeed or fail and an Action chain must deal with it. The default value stored for an action chain that does not have this action chain is "stop on failure". AC-Failure types are:

- \* 0x00: Stop on failure of an action.
- \* 0x01: Continue on failure of an action.

\*Reserved:\* Reserved for future use. Must be set to all zeros, and ignored upon reception.

## 5. Validation and Ordering of FS Routes

The validation of FSv2 routes adheres to the combination of rules for general BGP FSv1 routes found in [FSv1], [FSv1-IPv6], and [RFC9117]. These FSv1 rules are sufficient for FSv2 for IP traffic.

Specific additions have been defined for IP Filters used for guiding IP traffic into Service Function Service Function Pathways SFC NLRI in [RFC9015], or validation of L2VPN FS NLRI (see [I-D.ietf-idr-flowspec-l2vpn]). These additions are not required for the FSv2 for IP Basic functions.

Validating FSv2 routes proceeds through the following steps:

- \* Syntactic and semantic validation for FSv2 NLRI (Section 5.1).
- \* Validating FSv2 route properties (Section 5.2.2).
- \* Validating FSv2 route actions (Section 5.2.3).

The full validation process for FSv2 routes for all AFI/SAFIs is described below in Section 5.2.2 rather than simply referring to the relevant portions of the previously referenced RFCs.

### 5.1. Validating FSv2 NLRI

All FSv2 NLRI MUST be well-formed.

Failure of the following NLRI validation conditions MUST use "session reset" for [RFC7606] purposes since recovery from NLRI malformation cannot is not possible:

- \* NLRI Length (Section 3.1) MUST be at least 20 octets:
  - Dependent Filters Chain (4 octets) + User Order (4 octets) +

- One non-empty FSv2 Filter Family TLV (FSv2 Filter Family Type (2 octets) + FSv2 Filter Family Length (2 octets)) +
  - One possibly-empty FSv2 Filter Component (FSv2 Filter Component Flags + Type (2 octets)+ FSv2 Filter Component Length (2 octets))
- \* All TLVs and sub-TLVs MUST be well-formed and exactly contained in their parent TLVs: The total length of all sub-TLVs must be identical to the length field of the parent TLV.

Failure of the following NLRI validation conditions MUST use "treat-as-withdraw" for the NLRI for [RFC7606] purposes. In these cases, it is possible to parse the boundaries of individual NLRI in a BGP UPDATE message and thus the BGP speaker can continue to parse the next NLRI in the UPDATE. Implementations also MUST notify the operator of this behavior: In circumstances where routes have been announced by a previously valid NLRI but failed to be properly withdrawn due to an implicit or explicit withdraw of a malformed NLRI, "stuck" routes may result in the network.

- \* TLVs of a given class (FSv2 Filter Family, FSv2 Filter Component for a FSv2 Filter Family) MUST be present no more than once in an NLRI. (No duplicates TLVs.)
- \* TLVs of a given class MUST be ordered from lowest to highest. (TLVs need to be sorted.)
- \* When a TLV's value field is understood by the implementation, the value MUST have a length appropriate for that TLV type.
- \* When a TLV's value field is understood by the implementation, the value field MUST be well-formed according the definition of that TLV type.

Implementations MAY, depending on configuration, restrict propagation of FSv2 routes with NLRI containing Filter Families or Filter Components that they are ignorant of the encodings for. This is permitted only when the NLRI are otherwise not considered malformed by the implementation. This behavior is useful for BGP speakers, such as route reflectors, to generically disseminate FSv2 routes that they themselves might not utilize for traffic filtering.

The above rules permit FSv2 implementations that are ignorant of a given Filter Family, or Filter Family's Component encoding to propagate the FSv2 route to other BGP speakers in the deployment. However, since semantic checks for a given Filter Family's components can only be effected by implementations aware of that component, ignorant upstream BGP speakers may propagate semantically-incorrect NLRI until it reaches a BGP speaker that understands the encoding.

## 5.2. Validation of FSv2 BGP Routes

By Section 1.1 of [RFC4271] definition, a BGP route is a pairing of its destination (NLRI) and Path Attributes. The prior section discussed the validation of the NLRI. This section discusses validation of the pairing of the NLRI in an UPDATE along with their Path Attributes as BGP routes.

Flow specifications received from a BGP peer that are accepted in the respective Adj-RIB-In are used as input to the route selection process. Although the forwarding attributes of the two routes for the same prefix may be the same, BGP is still required to perform its path selection algorithm in order to select the correct set of attributes to advertise.

The first step of the BGP Route selection procedure (Section 9.1.2 of [RFC4271]) is to exclude from the selection procedure routes that are considered unfeasible. In the context of IP routing information, this is used to validate that the next hop of a given route is resolvable.

This concept can be extended in the case of the Flow Specification NLRI to allow other validation procedures.

### 5.2.1. AFI/SAFIs Used For Validation

The FSv2 validation process validates the FSv2 NLRI with following unicast routes received over the same AFI (1 or 2) but different SAFIs:

Received AFI/SAFI	Validate Route Using AFI/SAFI
1/TBD1	1/1 (IPv4-Unicast)
1/TBD2	1/128 (IPv4-Labeled Unicast)
2/TBD1	2/1 (IPv6-Unicast)
2/TBD2	2/128 (IPv6-Labeled Unicast)
31/TBD1	1/1 (IPv4-Unicast)
31/TBD2	1/128 (IPv4-Labeled Unicast)
6/TBD1	1/1 (IPv4-Unicast)
256/TBD2	1/128 (IPv4-Labeled Unicast)

Table 4: FSV2 Flowspec BGP Route AFI/SAFI Validation

#### 5.2.2. FSV2 Route Validation Procedure

In the absence of explicit configuration, a Flow specification NLRI (FSv1 or FSv2) MUST be validated such that it is considered feasible if and only if all of the conditions are true:

- a. A destination prefix component is embedded in the Flow Specification.
- b. One of the following conditions holds true:
  1. The originator of the Flow Specification matches the originator of the best-match unicast route for the destination prefix embedded in the flow specification (this is the unicast route with the longest possible prefix length covering the destination prefix embedded in the flow specification).
  2. The AS\_PATH attribute of the flow specification is empty or contains only an AS\_CONFED\_SEQUENCE segment. [RFC5065].
    - 2a This condition SHOULD be enabled by default.
    - 2b This condition MAY be disabled by explicit configuration on a BGP Speaker.

2c As an extension to this rule, a given non-empty AS\_PATH (besides AS\_CONFED\_SEQUENCE segments) MAY be permitted by policy.

- c. There are no "more-specific" unicast routes when compared with the flow destination prefix that have been received from a different neighbor AS than the best-match unicast route, which has been determined in rule b.

However, part of rule a may be relaxed by explicit configuration, permitting Flow Specifications that include no destination prefix component. If such is the case, rules b and c are moot and MUST be disregarded.

By "originator" of a BGP route, we mean either the address of the originator in the ORIGINATOR\_ID Attribute [RFC4456] or the source address of the BGP peer, if this path attribute is not present.

A BGP implementation MUST enforce that the AS in the left-most position of the AS\_PATH attribute of a Flow Specification Route (FSv1 or FSv2) received via the Exterior Border Gateway Protocol (eBGP) matches the AS in the left-most position of the AS\_PATH attribute of the best-match unicast route for the destination prefix embedded in the Flow Specification (FSv1 or FSv2) NLRI.

The best-match unicast route may change over time independently of the Flow Specification NLRI (FSv1 or FSv2). Therefore, a revalidation of the Flow Specification MUST be performed whenever unicast routes change. Revalidation is defined as retesting rules a to c as described above.

### 5.2.3. Validation of Flow Specification Actions for FSv2 for IP Basic

FSv2 routes can carry one or more filtering action extended communities (FS-EC) that are executed when the flow specification filter matches traffic. These extended communities are syntactically validated using the procedures in [RFC4360] and [RFC7606].

Section 4.5.2 discusses the procedures for utilizing FSv2-EC actions as part of traffic filtering.

## 6. Traffic Filtering

Section 5 of [FSv1] discusses the general behavior of using flow specification for traffic filtering. FSv2 provides the additional ability to apply traffic filtering at different portions of a forwarding path.

The code points assigned to the Filter Family Types and Filter Component Types for a given Filter Family are arranged to support a reasonable default traffic filtering (match, and actions) behavior. For example, the Component orders for FSv1 and FSv2 IP Basic can match traffic as part of monitoring, or mitigating, distributed denial of service (DDoS) attacks. However, that default ordering may be unsuitable for all filtering situations. FSv1 provided no mechanism to deviate from the ordering rules in Section 5.1 of [FSv1].

The User Order field of the NLRI (Section 3.1) permits an operator to override the default sort ordering of FSv2 rules to effect their desired traffic filtering behavior when it deviates from the default order.

Note that the procedures in Section 5.1 have ensured that TLVs are distinctly numbered and sorted. This assists with the procedures in the following section.

#### 6.1. Ordering of FSv2 Flow Specifications

More than one Flow Specification may match a particular traffic flow. Thus, it is necessary to define the order in which Flow Specifications get matched and actions being applied to a particular traffic flow. This ordering function is such that it does not depend on the arrival order of the Flow Specification via BGP and thus is consistent in the network.

FSv2 routes consist of a series of Filter Families containing Filter Components for those Filter Families. Filter Families are generally ordered where their match criteria match lower network layers based on lower-numbered Filter Family Types. However, they may also be ordered based on where the default match order for that Filter Family vs. other Filter Families should occur.

Similarly, within a Filter Family, Filter Components are ordered based to permit the default match order for that Filter Family to be naturally ordered as part of sorting FSv2 routes.

Thus, for FSv2, the choice of code point for Filter Family, or Filter Component is chosen to represent the default sort order for traffic filtering.

The relative order of two FSv2 flow specifications is determined in the following fashion:

1. A route with a lower User Order value (Section 3.1) comes before a route with a higher User Order value.

2. Each route's Filter Family TLVs are then compared in a pair-wise fashion. A route with a lower FSv2 Filter Family Type value (Section 3.1.1) comes before a route with a higher Filter Family Type value.
  3. When both routes have the same Filter Family Type, each Filter Component TLV for that Filter Family are compared in a pair-wise fashion. A route with a lower FSv2 Filter Component Type value (Section 3.1.2) comes before a route with a higher Filter Component Type value.
  4. When Filter Component Types are identical, Filter Component Values are compared:
    - \* For IP prefix values (IP destination or source prefix), if one of the two prefixes to compare is a more specific prefix of the other, the route with the more-specific prefix comes before the route with the less-specific prefix. Otherwise, the route with the lowest IP value comes before the route with the higher IP value.
    - \* For all other Filter Component Types, unless otherwise specified, the comparison is performed by comparing the Filter Component data as a binary string using the memcmp() function as defined by [ISO\_IEC\_9899]. For strings with equal lengths, the lowest string (memcmp) has higher precedence. For strings of different lengths, the common prefix is compared. If the common prefix is not equal, the string with the lowest prefix has higher precedence. If the common prefix is equal, the longest string is considered to have higher precedence than the shorter one.
- \*Warning:\* Specifications for FSv2 Filter Components are permitted to define their sort comparison criteria for that component. However, when implementations are ignorant of that Filter Component, it can only sort the components based on the general memcmp mechanism described above. In the case where a deployment contains implementations that are ignorant of a given filtering behavior, one of the two things SHOULD be done by the operator to avoid inappropriate traffic filtering or forwarding:
- \* The User Order field should be utilized to prevent inappropriate ordering of FSv2 routes that ignorant implementations may misorder.
  - \* The Filter Component Type should be marked as "mandatory" (Section 3.1.2) and dependent FSv2 filters placed in an appropriate, non-zero, Dependent Filter Chain (Section 3.1).

## 6.2. Installation of FSv2 Filters

Once FSv2 flow specifications have been ordered according to the rules of the prior section, they are eligible to be installed for traffic filtering purposes. However, it is possible that a given device is incapable of implementing all match components, or actions.

FSv2 flow specifications are evaluated to see if their match and action elements are able to be executed on the device. When the evaluation is "valid", the flow specification (match and actions) are eligible to be installed in the relative sort order determined in the prior section.

When FSv2 flow specifications are determined to be "invalid", it impacts not only the individual flow specification that has been deemed invalid, but also all FSv2 entries sharing the same non-zero Dependent Filter Chain value (Section 3.1).

For filtering components, Section 3.1.2 defines the FSv2 Filter Component Flags field. When a device is unable to implement the match criteria contained in that Filter Component - for whatever reason - the "Optional" bit is checked. If the Optional bit is unset (zero), the Filter Component is "mandatory" and the flow specification filter is considered "invalid". If the filter bit is set (one), the Filter Component is "optional", and the device is free to install the flow specification in the sorted order minus the Filter Component in question as a "valid" entry.

Similarly, if a flow specification's traffic filtering actions are unable to be installed by the device, the implementation may determine whether or not the flow specification is valid or invalid based on implementation defaults, or configuration. The ACO community may be used on supporting implementations to influence validity in these circumstances.

Features governing ordered FSv2 action and validity evaluation may be considered in the future.

Once validation of all FSv2 flow specification is complete, eligible FSv2 flow specifications are installed as traffic filters.

## 6.3. Ordering of FS filters for BGP Peers which support FSv1 and FSv2

FSv2 allows the user to order flow specification rules and the actions associated with a rule. Each FSv2 rule has one or more match conditions and one or more actions associated with each rule.

FSv1 and FSV2 filters are sent as different AFI/SAFI pairs so FSV1 and FSV2 operate as ships-in-the-night. Some BGP peers in an AS may support both FSV1 and FSV2. Other BGP peers may support FSV1 or FSV2. Some BGP will not support FSV1 or FSV2. A coherent flow specification technology must have consistent best practices for ordering the FSV1 and FSV2 filter rules.

One simple rule captures the best practice: Order the FSV1 filters after the FSV2 filter by placing the FSV1 filters after the FSV2 filters.

To operationally make this work, all flow specification filters should be included the same data base with the FSV1 filters being assigned a user-defined order beyond the normal size of FSV2 user-ordered values. A few examples, may help to illustrate this best practice.

Example 1: User ordered numbering - Suppose you might have 1,000 rules for the FSV2 filters. Assign all the FSV1 user defined rules to 1,001 (or better yet 2,000). The FSV1 rules will be ordered by the components and component values.

Example 2: Storage of actions - All FSV1 actions are defined ordered actions in FSV2. Translate your FSV1 actions into FSV2 ordered actions for storing in a common FSV1-FSV2 flow specification data base.

## 7. Scalability and Aspirations for FSV2

Operational issues drive the deployment of BGP flow specification as a quick and scalable way to distribute filters. The early operations accepted the fact validation of the distribution of filter needed to be done outside of the BGP distribution mechanism. Other mechanisms (NETCONF/RESTCONF or PCEP) have reply-request protocols.

These features within BGP have not changed. BGP still does not have an action-reply feature.

NETCONF/RESTCONF latest enhancements provide action/response features which scale. The combination of a quick distribution of filters via BGP and a long-term action in NETCONF/RESTCONF that ask for reporting of the installation of FSV2 filters may provide the best scalability.

The combination of NETCONF/RESTCONF network management protocols and BGP focuses each protocol on the strengths of scalability.

FSv2 will be deployed in webs of BGP peers which have some BGP peers passing FSv1, some BGP peers passing FSv2, some BGP peers passing FSv1 and FSv2, and some BGP peers not passing any routes.

The TLV encoding and deterministic behaviors of FSv2 will not deprecate the need for careful design of the distribution of flow specification filters in this mixed environment. The needs of networks for flow specification are different depending on the network topology and the deployment technology for BGP peers sending flow specification.

Suppose we have a centralized RR connected to DDoS processing sending out flow specification to a second tier of RR who distribute the information to targeted nodes. This type of distribution has one set of needs for FSv2 and the transition from FSv1 to FSv2.

Suppose we have Data Center with a 3-tier backbone trying to distribute DDoS or other filters from the spine to combinational nodes, to the leaf BGP nodes. The BGP peers may use RR or normal BGP distribution. This deployment has another set of needs for FSv2 and the transition from FSv1 to FSv2.

Suppose we have a corporate network with a few AS sending DDoS filters using basic BGP from a variety of sites. Perhaps the corporate network will be satisfied with FSv1 for a long time.

These examples are given to indicate that BGP FSv2, like so many BGP protocols, needs to be carefully tuned to aid the mitigation services within the network. This protocol suite starts the migration toward better tools using FSv2, but it does not end it. With FSv2 TLVs and deterministic actions, new operational mechanisms can start to be understood and utilized.

This FSv2 specification is merely the start of a revolution of work â\200\223 not the end.

## 8. Optional Security Additions

This section discusses the optional BGP Security additions for BGP-FSv2 relating ROA [RFC9582].

### 8.1. BGP FSv2 with ROA

BGP FSv2 can utilize ROAs in the validation. If BGP FSv2 is used with BGPSEC and ROA, the first thing is to validate the route within BGPSEC and second to utilize BGP ROA to validate the route origin.

The BGP-FS peers using both ROA and BGP-FS validation determine that a BGP Flow specification is valid if and only if one of the following cases:

- \* If the BGP Flow Specification NLRI has a IPv4 or IPv6 address in destination address match filter and the following is true:
  - A BGP ROA has been received to validate the originator, and
  - The route is the best-match unicast route for the destination prefix embedded in the match filter; or
- \* If a BGP ROA has not been received that matches the IPv4 or IPv6 destination address in the destination filter, the match filter must abide by the [FSv1] and [FSv1-IPv6] validation rules as follows:
  - The originator match of the flow specification matches the originator of the best-match unicast route for the destination prefix filter embedded in the flow specification", and
  - No more specific unicast routes exist when compared with the flow destination prefix that have been received from a different neighboring AS than the best-match unicast route, which has been determined in step A.

The best match is defined to be the longest-match NLRI with the highest preference.

## 9. IANA Considerations

This section complies with [RFC7153].

### 9.1. Flow Specification V2 SAFIs

IANA is requested to assign two SAFI Values in the registry at <https://www.iana.org/assignments/safi-namespace> from the Standard Action Range as follows:

Table 7-1 SAFIs

Value	Description	Reference
TBD1	BGP FSv2	[this document]
TBD2	BGP FSv2 VPN	[this document]

## 9.2. Generic Transitive Extended Community

IANA is requested to assign a type value from the "Generic Transitive Extended Community Sub-Types" registry at <https://www.iana.org/assignments/bgp-extended-communities/bgp-extended-communities.xhtml>

Table 7-3 - Generic Transitive Extended Community

Value	Description	Reference	Controller
TBD4	FSv2 Action Chain Ordering	[this document]	IETF

## 9.3. FSv2 IP Filters Component Types

IANA is requested to create a new "BGP FSv2 IP Basic Component Types" registry and indicate [this draft] as a reference. The following assignments in the FSv2 IP Basic Filters Component Types Registry should be made.

Registry Name: BGP FSv2 Component Types

Reference: [this document]

Registration Procedures: 0x01-0x3FFF Standards Action, 0x4000-0xFFFF FCFS.

Type	Definition	Reference
0	Reserved	This document
10	IP Destination Prefix	This document
20	IP Source Prefix	This document
30	IPv4 Protocol / IPv6 Upper Layer Protocol	This document
40	Port	This document
50	Destination Port	This document
60	Source Port	This document
70	ICMPv4 Type / ICMPv6 Type	This document
80	ICMPv4 Code / ICPv6 Code	This document
90	TCP Flags	This document
100	Packet Length	This document
110	DSCP	This document
120	Fragment	This document
130	Flow Label	This document
4095	Reserved	This document

Table 5: BGP FSv2 IP Basic Component Types

#### 9.4. FSv2 Filter Component Types

IANA is requested to create the a new registry for "Flow Specification v2 Filter Component Types".

Registration Procedures: 0x01-0x3FFF Standards Action.

Type	Description	Reference
0	Reserved	[this document]
1-49	Unassigned	[this document]
50	L2 Traffic Rules	[this document]
51-99	Unassigned	[this document]
100	MPLS traffic rules	[this document]
101-149	Unassigned	[this document]
150	SFC Traffic rules	[this document]
151-199	Unassigned	[this document]
200	Tunnel Traffic rules	[this document]
201-255	Unassigned	[this document]
256	IP traffic rules	[this document]
257-279	Unassigned	[this document]
280	Extended IP Rules	[this document]
281-24575	Unassigned	[this document]
24576-32767	Vendor specific	[this document]
32768-65535	Reserved	[this document]

Table 6: Flow Specification v2 Filter Component Types

## 10. Security Considerations

The use of ROA improves on [FSv1] by checking to see of the route origination. This check can improve the validation sequence for a multiple-AS environment.

>The use of BGPSEC [RFC8205] to secure the packet can increase security of BGP flow specification information sent in the packet.

The use of the reduced validation within an AS [RFC9117] can provide adequate validation for distribution of flow specification within a single autonomous system for prevention of DDoS.

Distribution of flow filters may provide insight into traffic being sent within an AS, but this information should be composite information that does not reveal the traffic patterns of individuals.

## 11. References

### 11.1. Normative References

- [FSv1] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <<https://www.rfc-editor.org/info/rfc8955>>.
- [FSv1-IPv6] Loibl, C., Ed., Raszuk, R., Ed., and S. Hares, Ed., "Dissemination of Flow Specification Rules for IPv6", RFC 8956, DOI 10.17487/RFC8956, December 2020, <<https://www.rfc-editor.org/info/rfc8956>>.
- [I-D.ietf-idr-flowspec-l2vpn] Weiguo, H., Eastlake, D. E., Litkowski, S., and S. Zhuang, "BGP Dissemination of L2 Flow Specification Rules", Work in Progress, Internet-Draft, draft-ietf-idr-flowspec-l2vpn-27, 16 March 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-flowspec-l2vpn-27>>.
- [I-D.ietf-idr-flowspec-nvo3] Eastlake, D. E., Weiguo, H., Zhuang, S., Li, Z., and R. Gu, "BGP Dissemination of Flow Specification Rules for Tunneled Traffic", Work in Progress, Internet-Draft, draft-ietf-idr-flowspec-nvo3-23, 5 December 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-flowspec-nvo3-23>>.
- [I-D.ietf-idr-flowspec-srv6] Li, Z., Chen, H., Loibl, C., Mishra, G. S., Fan, Y., Zhu, Y., Liu, L., Liu, X., and S. Zhuang, "BGP Flow Specification for SRv6", Work in Progress, Internet-Draft, draft-ietf-idr-flowspec-srv6-08, 24 November 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-flowspec-srv6-08>>.

## [I-D.ietf-idr-wide-bgp-communities]

Raszuk, R., Haas, J., Lange, A., Decraene, B., Amante, S., and P. Jakma, "BGP Community Container Attribute", Work in Progress, Internet-Draft, draft-ietf-idr-wide-bgp-communities-12, 17 March 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-wide-bgp-communities-12>>.

## [interface-set]

Litkowski, S., Simpson, A., Patel, K., and J. Haas, "Applying BGP flowspec rules on a specific interface-set", Work in Progress, Internet-Draft, draft-ietf-idr-flowspec-interfaceset-06, 2 September 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-flowspec-interfaceset-06>>.

## [path-redirect]

Van de Velde, G., Patel, K., and Z. Li, "Flowspec Indirection-id Redirect", Work in Progress, Internet-Draft, draft-ietf-idr-flowspec-path-redirect-13, 22 April 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-flowspec-path-redirect-13>>.

## [redirect-ip]

Haas, J., Henderickx, W., and A. Simpson, "BGP Flow-Spec Redirect-to-IP Action", Work in Progress, Internet-Draft, draft-ietf-idr-flowspec-redirect-ip-10, 28 April 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-flowspec-redirect-ip-10>>.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <<https://www.rfc-editor.org/info/rfc768>>.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <<https://www.rfc-editor.org/info/rfc792>>.

[RFC0793] Postel, J., "Transmission Control Protocol", RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.

- [RFC1997] Chandra, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997, DOI 10.17487/RFC1997, August 1996, <<https://www.rfc-editor.org/info/rfc1997>>.
- [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>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <<https://www.rfc-editor.org/info/rfc4360>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <<https://www.rfc-editor.org/info/rfc4456>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<https://www.rfc-editor.org/info/rfc4760>>.
- [RFC5065] Traina, P., McPherson, D., and J. Scudder, "Autonomous System Confederations for BGP", RFC 5065, DOI 10.17487/RFC5065, August 2007, <<https://www.rfc-editor.org/info/rfc5065>>.

- [RFC5701] Rekhter, Y., "IPv6 Address Specific BGP Extended Community Attribute", RFC 5701, DOI 10.17487/RFC5701, November 2009, <<https://www.rfc-editor.org/info/rfc5701>>.
- [RFC7153] Rosen, E. and Y. Rekhter, "IANA Registries for BGP Extended Communities", RFC 7153, DOI 10.17487/RFC7153, March 2014, <<https://www.rfc-editor.org/info/rfc7153>>.
- [RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <<https://www.rfc-editor.org/info/rfc7606>>.
- [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>>.
- [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>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC9015] Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L. Jalil, "BGP Control Plane for the Network Service Header in Service Function Chaining", RFC 9015, DOI 10.17487/RFC9015, June 2021, <<https://www.rfc-editor.org/info/rfc9015>>.
- [RFC9117] Uttaro, J., Alcaide, J., Filsfils, C., Smith, D., and P. Mohapatra, "Revised Validation Procedure for BGP Flow Specifications", RFC 9117, DOI 10.17487/RFC9117, August 2021, <<https://www.rfc-editor.org/info/rfc9117>>.
- [RFC9582] Snijders, J., Maddison, B., Lepinski, M., Kong, D., and S. Kent, "A Profile for Route Origin Authorizations (ROAs)", RFC 9582, DOI 10.17487/RFC9582, May 2024, <<https://www.rfc-editor.org/info/rfc9582>>.

## 11.2. Informative References

- [fsv2] Hares, S., Eastlake, D. E., Yadlapalli, C., and S. Maduschke, "BGP Flow Specification Version 2", Work in Progress, Internet-Draft, draft-ietf-idr-flowspec-v2-04, 28 April 2024, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-flowspec-v2-04>>.
- [fsv2-more-ip-filters] Hares, S., "BGP Flow Specification Version 2 - More IP Actions", Work in Progress, Internet-Draft, draft-hares-idr-fsv2-more-ip-actions-03, 17 October 2024, <<https://datatracker.ietf.org/doc/html/draft-hares-idr-fsv2-more-ip-actions-03>>.
- [I-D.hares-idr-fsv2-more-ip-filters] Hares, S. and N. Kao, "BGP Flow Specification Version 2 - More IP Filters", Work in Progress, Internet-Draft, draft-hares-idr-fsv2-more-ip-filters-05, 17 March 2026, <<https://datatracker.ietf.org/doc/html/draft-hares-idr-fsv2-more-ip-filters-05>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8205] Lepinski, M., Ed. and K. Sriram, Ed., "BGPsec Protocol Specification", RFC 8205, DOI 10.17487/RFC8205, September 2017, <<https://www.rfc-editor.org/info/rfc8205>>.

## Authors' Addresses

Susan Hares  
Hickory Hill Consulting  
7453 Hickory Hill  
Saline, MI 48176  
United States of America  
Phone: +1-734-604-0332  
Email: shares@ndzh.com

Donald Eastlake  
Independent  
2386 Panoramic Circle  
Apopka, FL 32703  
United States of America  
Phone: +1-508-333-2270  
Email: d3e3e3@gmail.com

Jie Dong  
Huawei Technologies  
No. 156 Beiqing Road  
Beijing  
China  
Email: jie.dong@huawei.com

Chaitanya Yadlapalli  
ATT  
United States of America  
Email: cy098d@att.com

Sven Maduschke  
Verizon  
Germany  
Email: sven.maduschke@de.verizon.com

Jeffrey Haas  
HPE  
United States of America  
Email: jeffrey.haas@hpe.com

IDR Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 7 August 2026

N. Kao  
Individual Contributor  
S. Hares  
Hickory Hill Consulting  
3 February 2026

Bitwise IP Filters for BGP FlowSpec  
draft-cao-idr-bitwise-ip-filters-04

Abstract

This draft introduces the bitwise matching filters for source or destination IPv4/IPv6 address fields. These filters enhance the functionalities of the BGP Flow Specification framework and aid scenarios involving symmetric traffic load balancing.

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 7 August 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1.	Introduction . . . . .	2
1.1.	Requirements Language . . . . .	3
1.2.	Definitions and Acronyms . . . . .	3
2.	Bitwise Address Filters for FSv2 . . . . .	3
2.1.	Destination Address Bitwise Filter Component Sub-TLV . . . . .	3
2.2.	Source Address Bitwise Filter Component Sub-TLV . . . . .	4
2.3.	Ordering Procedures for the Sub-TLVs . . . . .	5
3.	Use Cases . . . . .	6
3.1.	Symmetric Traffic Load Balancing . . . . .	6
3.2.	Dynamic Service Scaling . . . . .	8
3.3.	Symmetric Traffic Sampling . . . . .	8
4.	Comparisons with Other Approaches . . . . .	8
4.1.	Comparison with Existing Prefix Filters . . . . .	9
4.2.	Comparison with the Content Filter . . . . .	9
4.3.	Comparison with the Hash-based ECMP . . . . .	9
5.	Deployment Guidelines . . . . .	10
6.	IANA Considerations . . . . .	10
7.	Security Considerations . . . . .	10
8.	Normative References . . . . .	11
9.	Informative References . . . . .	11
	Acknowledgements . . . . .	12
	Authors' Addresses . . . . .	12

## 1. Introduction

Symmetric paths for both directions are required to allow session inspecting service instances (such as the deep packet inspection(DPI) or the firewall service instances) to process the traffic correctly. If a single instance cannot handle the traffic load, symmetric load balancing between multiple inspecting service instances is needed.

Hash-based load balancing may not be suitable for this purpose since the order of fields for the hashing mechanism may not be configurable, and different vendors implement different proprietary hashing functions. In a multi-vendor environment, it is desirable to load-balance traffic between each instance using a standardized approach.

The BGP Flow Specification(BGP-FS) Network Layer Reachability Information(NLRI) is a standardized approach to distributing traffic filters via BGP. The BGP Flow Specification version 2(FSv2) defined in [I-D.ietf-idr-fsv2-ip-basic] enhances BGP-FS, allowing user-defined order of filters. The Extended IP Filters defined in [I-D.hares-idr-fsv2-more-ip-filters] further extend the filter types of FSv2.

This draft defines the bitwise matching filters for source or destination IPv4/IPv6 address fields. We can achieve dynamic symmetric traffic load-balancing using these filters if the traffic is distributed almost uniformly among combinations of the least significant bits of the source or destination address fields.

### 1.1. 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.

### 1.2. Definitions and Acronyms

AFI: Address Family Identifier

BGP-FS: BGP Flow Specification [RFC8955] [RFC8956]

DPI: Deep Packet Inspection

ECMP: Equal-Cost Multipath

FSv2: BGP Flow Specification Version 2 [I-D.ietf-idr-fsv2-ip-basic]

SAFI: Subsequent Address Family Identifier

Service Instance: A service instance is an instance of VNF or a physical device that instantiates a service function. It is spawned and terminated dynamically based on the traffic loads.

VNF: Virtual Network Function

## 2. Bitwise Address Filters for FSv2

This section defines the bitwise address filter component Sub-TLVs for FSv2. These Sub-TLVs are classified as "IP Extended Filters" as defined in Section 2.5 of [I-D.hares-idr-fsv2-more-ip-filters]. Sub-TLVs for both source and destination address fields are defined.

### 2.1. Destination Address Bitwise Filter Component Sub-TLV

Summary: This section defines bitwise matches for the destination address field.

Component type: TBD1

Description: This component performs matching against designated bits in the destination address field.

The field that the filter targets in the IPv4 Packets: Destination address field

The field that the filter targets in the IPv6 Packets: Destination address field

Length: This field indicates the length of the value field in octets. The length is 8 for AFI = 1 (IPv4) and 32 for AFI = 2 (IPv6). If the length is neither 8 nor 32, the NLRI is considered MALFORMED. If the length is inconsistent with the AFI definition, the NLRI is also considered MALFORMED.

Encoding of Component Value field: The match is encoded as a 2-tuple of the form <Prefix, Mask> , where:

Prefix: a 4-octet bit string for AFI = 1 and a 16-octet bit string for AFI = 2. It indicates the destination address value to match against.

Mask: a 4-octet bit string for AFI = 1 and a 16-octet bit string for AFI = 2. It indicates the bit positions to match. In the matching process, we only consider bit positions designated with value 1 in the Mask field. An address is a match if and only if the value of the address matches the value in the Prefix field in every designated bit position.

Conflicts with other filters: None

## 2.2. Source Address Bitwise Filter Component Sub-TLV

Summary: This section defines bitwise matches for the source address field.

Component type: TBD2

Description: This component performs matching against designated bits in the source address field.

The field that the filter targets in the IPv4 Packets: Source address field

The field that the filter targets in the IPv6 Packets: Source address field

Length: This field indicates the length of the value field in

octets. The length is 8 for AFI = 1 (IPv4) and 32 for AFI = 2 (IPv6). If the length is neither 8 nor 32, the NLRI is considered MALFORMED. If the length is inconsistent with the AFI definition, the NLRI is also considered MALFORMED.

Encoding of Component Value field: The match is encoded as a 2-tuple of the form <Prefix, Mask> , where:

Prefix: a 4-octet bit string for AFI = 1 and a 16-octet bit string for AFI = 2. It indicates the source address value to match against.

Mask: a 4-octet bit string for AFI = 1 and a 16-octet bit string for AFI = 2. It indicates the bit positions to match. In the matching process, we only consider bit positions designated with value 1 in the Mask field. An address is a match if and only if the value of the address matches the value in the Prefix field in every designated bit position.

Conflicts with other filters: None

### 2.3. Ordering Procedures for the Sub-TLVs

This section describes the procedures for sorting the Sub-TLVs defined in this draft of the same type.

Multiple Occurrences of the Sub-TLV of the same type in the same NLRI: The Sub-TLV of the same type with different values MAY appear multiple times in the same NLRI. The address field matches if it matches any instances of the Sub-TLV that appear in the NLRI. When sending multiple instances of the Sub-TLV of the same type, the following rules apply:

1. Duplicate values are not allowed. The Sub-TLVs with the same value MUST appear only once.
2. Comparing the value field in each instance as a binary string using the memcmp() function as defined by [ISO\_IEC\_9899] determines the precedence of the instance. The lowest one (memcmp) has higher precedence.
3. The Sub-TLV instance with the highest precedence MUST precede instances with lower precedence.

Filter Ordering Rules of the Sub-TLV of the same type between different NLRIs: When comparing two NLRIs with the same type of Sub-TLV instances, the following rules apply:

1. The Sub-TLV instances in the same NLRI received must be in strict value-ascending order, or the NLRI is considered MALFORMED.
2. Compare the sequence of the Sub-TLV instances from each NLRI as binary strings using the memcmp() function defined by [ISO\_IEC\_9899]. If there are multiple instances of the Sub-TLV in the same NLRI, treat these instances as a single binary string. For strings of equal length, the lowest string has the highest precedence. For strings of different lengths, compare the common prefix of the string only. If the common prefix is unequal, the string with the lowest common prefix has higher precedence. If the common prefix is equal, the longest string has higher precedence than the shorter one.

### 3. Use Cases

This section describes various use cases for these filters.

#### 3.1. Symmetric Traffic Load Balancing

Referring to Figure 1, subscriber traffic is going through a set of inline service instances (e.g., DPIs, stateful firewalls) before entering the Internet. Inline service instances SVC0, SVC1, SVC2, and SVC3 need to process both directions of traffic in the same session. Any single service instance cannot handle the traffic volume alone. Assuming the traffic is distributed almost uniformly among combinations of the least significant 2 bits of the subscribers' addresses.

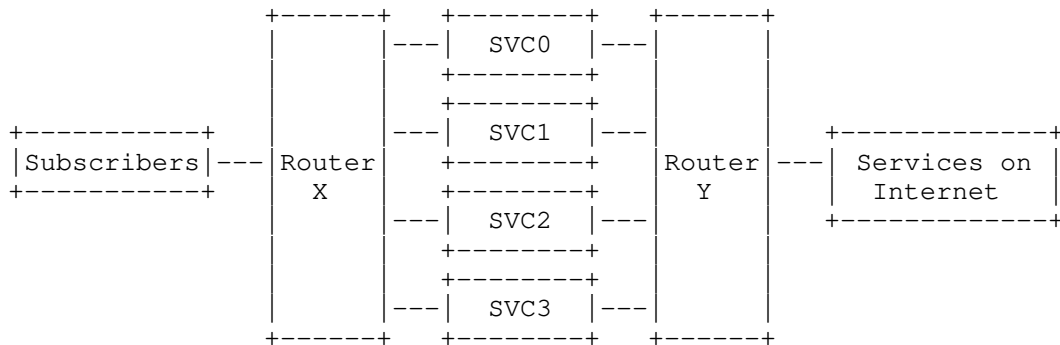


Figure 1: Symmetric Traffic Load Balancing Between Routers

Hash-based load balancing may not be suitable for this case since the order of fields for the hashing mechanism may not be configurable, and different vendors implement different proprietary hashing functions.

Deploying bitwise address filters is a viable multi-vendor solution in this case. For example, we can deploy:

On X:

- \* FSv2 rule X0 matches the least significant 2 bits as 00 in the source address field and directs traffic to SVC0.
- \* FSv2 rule X1 matches the least significant 2 bits as 01 in the source address field and directs traffic to SVC1.
- \* FSv2 rule X2 matches the least significant 2 bits as 10 in the source address field and directs traffic to SVC2.
- \* FSv2 rule X3 matches the least significant 2 bits as 11 in the source address field and directs traffic to SVC3.

On Y:

- \* FSv2 rule Y0 matches the least significant 2 bits as 00 in the destination address field and directs traffic to SVC0.
- \* FSv2 rule Y1 matches the least significant 2 bits as 01 in the destination address field and directs traffic to SVC1.
- \* FSv2 rule Y2 matches the least significant 2 bits as 10 in the destination address field and directs traffic to SVC2.
- \* FSv2 rule Y3 matches the least significant 2 bits as 11 in the destination address field and directs traffic to SVC3.

The matched traffic is directed to a specific service instance by various mechanisms, such as (but not limited to) the following:

- \* RT-redirect action defined in [RFC8955]
- \* Redirect-to-IP action as described in [I-D.ietf-idr-flowspec-redirect-ip]
- \* Indirection-id redirection as described in [I-D.ietf-idr-flowspec-path-redirect]

- \* Redirect into an SR Policy as described in [I-D.ietf-idr-ts-flowspec-srv6-policy]

We can load balance the traffic to N service instances while keeping the same session on the same instance using the rules matching the least significant  $\log(N)$  bits of the address fields.

### 3.2. Dynamic Service Scaling

Consider the same example depicted in Figure 1 . If the traffic load drops and we want to scale down the service by shutting down SVC2 and SVC3 to reduce costs, we can deploy new rules:

On X:

- \* FSv2 rule X0 matches the least significant bit as 0 in the source address field and directs traffic to SVC0.
- \* FSv2 rule X1 matches the least significant bit as 1 in the source address field and directs traffic to SVC1.

On Y:

- \* FSv2 rule Y0 matches the least significant bit as 0 in the destination address field and directs traffic to SVC0.
- \* FSv2 rule Y1 matches the least significant bit as 1 in the destination address field and directs traffic to SVC1.

We can remove the old rules and then shut down SVC2 and SVC3 after the new rules are activated.

### 3.3. Symmetric Traffic Sampling

Some capacity-limited devices, such as DPI equipment, may benefit from symmetric traffic sampling. Since these devices also require traffic from both directions, we can deploy bitwise filters to sample traffic flows by matching against a set of specific least significant bits from different subnets.

## 4. Comparisons with Other Approaches

This section compares the proposed solution with other existing approaches.

#### 4.1. Comparison with Existing Prefix Filters

IPv4 Source/Destination Prefix filters defined in [RFC8955] and [I-D.ietf-idr-fsv2-ip-basic] cannot match the least significant bits. IPv6 Source/Destination Prefix filters defined in [RFC8956] and [I-D.ietf-idr-fsv2-ip-basic] can either match the least significant bits or the IPv6 Prefix, but not both. These filters may not be suitable for use cases described in Section 3.1 without real-time traffic monitoring mechanisms for every possible source/destination prefix. The manageability and flexibility are not as good as the proposed solution either.

If both the prefix and bitwise matching are needed, using the bitwise matching filter is recommended since it provides both functionalities in the same filter. If only prefix matching is required, using filters defined in [RFC8955], [RFC8956], or [I-D.ietf-idr-fsv2-ip-basic] is more efficient and recommended.

#### 4.2. Comparison with the Content Filter

The content filter defined in [I-D.cui-idr-content-filter-flowspec] also provides the bitwise matching capability. Although the filter supports bitwise matching against any position in the packet, address matching is not its primary design goal. Manual calculation of offsets is required to use this filter. Therefore, it may not be optimal for scenarios that match address fields only.

#### 4.3. Comparison with the Hash-based ECMP

With the following conditions met, traditional hash-based ECMP may be used for the scenario described in Section 3.1.

- \* Routers X and Y implement the same hashing algorithm for ECMP, which usually means both routers are of the same vendor.
- \* The order of the fields for hashing must be reversible so that the hashing outcome will be on the same ECMP member for both directions.
- \* A mechanism to signal the order of ECMP members is needed. The BGP MultiNexthop(MNH) attribute defined in [I-D.ietf-idr-multinexthop-attribute] can distribute such information.

Even with all conditions met, we cannot determine which member a specific packet passes through before putting it into the router unless the router provides an interface to query that.

Therefore, the bitwise matching filter-based solution is more suitable for this scenario.

5. Deployment Guidelines

The filter defined in this document matches against arbitrary bits in the address fields. While syntactically this filter does not conflict with existing [RFC8955]/[RFC8956] prefix filters, using either the bitwise filter or the existing filter as described in Section 4.1 is recommended.

When deployed with redirection actions, these FS rules are actually PBR rules. Since these rules bypass the typical routing lookups just as typical PBR rules do, it is possible to form forwarding loops. User discretion is required.

If deployed with redirection actions and there are ECMPs to the redirection destination, the matching packets are subject to hash-based load-balancing towards that destination.

While some platforms are capable of processing bitwise filters at line-rate speeds, some aren't. Therefore, before deployments, it is recommended to determine the performance of bitwise matching.

6. IANA Considerations

IANA is requested to indicate [this draft] as a reference on the following assignments in the Flow Specification Component Types Registry:

Type Value	IPv4 Name	IPv6 Name	Reference
TBD1	Bitwise Destination IPv4 Address Filter	Bitwise Destination IPv6 Address Filter	[this draft]
TBD2	Bitwise Source IPv4 Address Filter	Bitwise Source IPv6 Address Filter	[this draft]

Table 1

7. Security Considerations

No new security issues are introduced to the BGP protocol by this specification.

## 8. 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>>.
- [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>>.
- [RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <<https://www.rfc-editor.org/info/rfc8955>>.
- [RFC8956] Loibl, C., Ed., Raszuk, R., Ed., and S. Hares, Ed., "Dissemination of Flow Specification Rules for IPv6", RFC 8956, DOI 10.17487/RFC8956, December 2020, <<https://www.rfc-editor.org/info/rfc8956>>.
- [I-D.ietf-idr-fsv2-ip-basic]  
Hares, S., Eastlake, D. E., Dong, J., Yadlapalli, C., and S. Maduschke, "BGP Flow Specification Version 2 - for Basic IP", Work in Progress, Internet-Draft, draft-ietf-idr-fsv2-ip-basic-03, 3 March 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-fsv2-ip-basic-03>>.
- [I-D.hares-idr-fsv2-more-ip-filters]  
Hares, S., "BGP Flow Specification Version 2 - More IP Filters", Work in Progress, Internet-Draft, draft-hares-idr-fsv2-more-ip-filters-04, 15 November 2024, <<https://datatracker.ietf.org/doc/html/draft-hares-idr-fsv2-more-ip-filters-04>>.
- [ISO\_IEC\_9899]  
ISO, "Information technology -- Programming languages -- C", ISO/IEC 9899:2018, June 2018.

## 9. Informative References

- [I-D.ietf-idr-flowspec-redirect-ip]  
Haas, J., Henderickx, W., and A. Simpson, "BGP Flow-Spec Redirect-to-IP Action", Work in Progress, Internet-Draft, draft-ietf-idr-flowspec-redirect-ip-04, 2 September 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-flowspec-redirect-ip-04>>.

[I-D.ietf-idr-flowspec-path-redirect]

Van de Velde, G., Patel, K., and Z. Li, "Flowspec Indirection-id Redirect", Work in Progress, Internet-Draft, draft-ietf-idr-flowspec-path-redirect-12, 24 November 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-flowspec-path-redirect-12>>.

[I-D.ietf-idr-ts-flowspec-srv6-policy]

Wenying, J., Liu, Y., Zhuang, S., Mishra, G. S., and S. Chen, "Traffic Steering using BGP FlowSpec with SR Policy", Work in Progress, Internet-Draft, draft-ietf-idr-ts-flowspec-srv6-policy-08, 1 February 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-ts-flowspec-srv6-policy-08>>.

[I-D.cui-idr-content-filter-flowspec]

Cui, Y., Gao, Y., and S. Hares, "Packet Content Filter for BGP FlowSpec", Work in Progress, Internet-Draft, draft-cui-idr-content-filter-flowspec-03, 14 August 2024, <<https://datatracker.ietf.org/doc/html/draft-cui-idr-content-filter-flowspec-03>>.

[I-D.ietf-idr-multinexthop-attribute]

Vairavakkalai, K., Jeganathan, J. M., Nanduri, M., and A. R. Lingala, "BGP MultiNexthop Attribute", Work in Progress, Internet-Draft, draft-ietf-idr-multinexthop-attribute-04, 25 March 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-multinexthop-attribute-04>>.

#### Acknowledgements

The authors would like to thank Robert Raszuk for comments and suggestions.

#### Authors' Addresses

Nat Kao  
Individual Contributor  
Email: [pyxislx@gmail.com](mailto:pyxislx@gmail.com)

Susan Hares  
Hickory Hill Consulting  
7453 Hickory Hill  
Saline, MI 48176  
United States of America  
Phone: +1-734-604-0332

Email: [shares@endzh.com](mailto:shares@endzh.com)