IDR Working Group
Internet-Draft
Intended status: Standards Track
Expires: 23 August 2025

N. Kao
Individual Contributor

19 February 2025

Bitwise IP Filters for BGP FlowSpec
draft-kao-idr-bitwise-ip-filters-00

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 23 August 2025.

Table of Contents

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.

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

1. No duplicates are allowed. The Sub-TLVs with the same value MUST appear only once.

2. The Sub-TLV instance with the highest precedence MUST precede the ones with lower precedence.

3. 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.

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

1. The Sub-TLV instances in the same NLRI received must be in a strict value-ascending order, or it 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]. For strings of equal lengths, 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, 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.

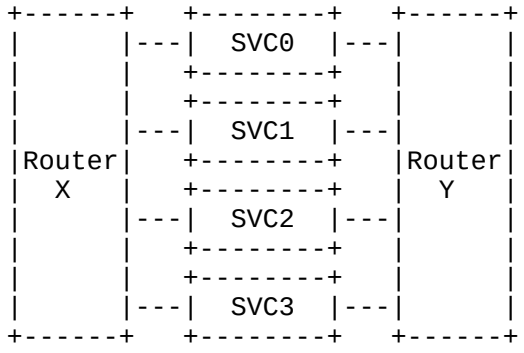

Figure 1: Symmetric Traffic Load Balancing Between Routers

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 two least significant bits as 00 in the source address field and directs traffic to SVC0.

* FSv2 rule X1 matches the two least significant bits as 01 in the source address field and directs traffic to SVC1.

* FSv2 rule X2 matches the two least significant bits as 10 in the source address field and directs traffic to SVC2.

* FSv2 rule X3 matches the two least significant bits as 11 in the source address field and directs traffic to SVC3.

On Y:

* FSv2 rule Y0 matches the two least significant bits as 00 in the destination address field and directs traffic to SVC0.

* FSv2 rule Y1 matches the two least significant bits as 01 in the destination address field and directs traffic to SVC1.

* FSv2 rule Y2 matches the two least significant bits as 10 in the destination address field and directs traffic to SVC2.

* FSv2 rule Y3 matches the two least significant 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 while keeping the traffic of the same session on the same service instance using these rules.

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.

4. Comparisons with Other Approaches

This section compares the proposed solution with other existing approaches.

4.1. Comparison with Existing Prefix Filters

In [RFC8955], [RFC8956], and [I-D.ietf-idr-fsv2-ip-basic] only IPv4/IPv6 longest-prefix-matching(LPM) filters(Sub-TLV Type 1 and 2) are defined, which cannot match discontiguous bits. 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 LPM and bitwise matching are needed, using the bitwise matching filter is recommended since it provides both functionalities in the same filter. If only LPM is needed, 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 suitable 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 MultiNexhop(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 unless the router provides an interface to query that.

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

5. 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

6. Security Considerations

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

7. Normative References

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

[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-02, 14 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-fsv2-ip-basic-02>.

[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.

8. Informative References

[I-D.ietf-idr-flowspec-redirect-ip]
Uttaro, J., Haas, J., akarch@cisco.com, Ray, S., Mohapatra, P., Henderickx, W., Simpson, A., and M. Texier, "BGP Flow-Spec Redirect-to-IP Action", Work in Progress, Internet-Draft, draft-ietf-idr-flowspec-redirect-ip-03, 8 September 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-flowspec-redirect-ip-03>.

[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-05, 6 January 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-ts-flowspec-srv6-policy-05>.

[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-03, 21 September 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-multinexthop-attribute-03>.

Acknowledgements

TBD

Contributors

TBD

Author’s Address

Nat Kao
Individual Contributor
Email: pyxislx@gmail.com