IDR Working Group
Internet-Draft
Obsoletes: 5575,7674 (if approved)
Intended status: Standards Track
Expires: April 27, 2018

S. Hares
Huawei
C. Loibl
Next Layer Communications
R. Raszuk
Bloomberg LP
D. McPherson
Verisign
M. Bacher
T-Mobile Austria

October 24, 2017

Dissemination of Flow Specification Rules
draft-ietf-idr-rfc5575bis-06

Abstract

This document updates [RFC5575] which defines a Border Gateway Protocol Network Layer Reachability Information (BGP NLRI) encoding format that can be used to distribute traffic Flow Specifications. This allows the routing system to propagate information regarding more specific components of the traffic aggregate defined by an IP destination prefix.

It specifies IPv4 traffic Flow Specifications via a BGP NLRI which carries traffic Flow Specification filter, and an Extended community value which encodes actions a routing system can take if the packet matches the traffic flow filters. The flow filters and the actions are processed in a fixed order. Other drafts specify IPv6, MPLS addresses, L2VPN addresses, and NV03 encapsulation of IP addresses.

This document updates [RFC5575] to correct unclear specifications in the flow filters and to provide rules for actions which interfere (e.g. redirection of traffic and flow filtering).

Applications which use the bgp Flow Specification are: 1) application which automate inter-domain coordination of traffic filtering, such as what is required in order to mitigate (distributed) denial-of-service attacks; 2) applications which control traffic filtering in the context of a BGP/MPLS VPN service, and 3) applications with centralized control of traffic in a SDN or NFV context. Some deployments of these three applications can be handled by the strict ordering of the BGP NLRI traffic flow filters, and the strict actions encoded in the extended community Flow Specification 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 April 27, 2018.

Copyright Notice

Copyright (c) 2017 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Definitions of Terms Used in This Memo 5
3. Flow Specifications 6
4. Dissemination of IPv4 FLow Specification Information 6
4.1. Length Encoding 7
4.2. NLRI Value Encoding 7
4.2.1. Type 1 - Destination Prefix 8
4.2.2. Type 2 - Source Prefix 8
4.2.3. Type 3 - IP Protocol 8
4.2.4. Type 4 - Port 10
4.2.5. Type 5 - Destination Port 10
4.2.6. Type 6 - Source Port 10
4.2.7. Type 7 - ICMP type 10
4.2.8. Type 8 - ICMP code 11

4.2.9. Type 9 - TCP flags 11
4.2.10. Type 10 - Packet length 12
4.2.11. Type 11 - DSCP (Diffserv Code Point) 12
4.2.12. Type 12 - Fragment 12
4.3. Examples of Encodings 12
5. Traffic Filtering 13
5.1. Ordering of Traffic Filtering Rules 14
6. Validation Procedure 16
7. Traffic Filtering Actions 17
7.1. Traffic Rate in Bytes (traffic-rate-bytes) sub-type 0x06 19
7.2. Traffic Rate in Packets (traffic-rate-packets) sub-type TBD 19
7.3. Traffic-action (traffic-action) sub-type 0x07 19
7.4. RT Redirect (rt-redirect) sub-type 0x08 20
7.5. Traffic Marking (traffic-marking) sub-type 0x09 20
7.6. Rules on Traffic Action Interference 21
7.6.1. Examples 21
8. Dissemination of Traffic Filtering in BGP/MPLS VPN Networks . 22
8.1. Validation Procedures for BGP/MPLS VPNs 23
8.2. Traffic Actions Rules 23
9. Limitations of Previous Traffic Filtering Efforts 23
9.1. Limitations in Previous DDoS Traffic Filtering Efforts . 23
9.2. Limitations in Previous BGP/MPLS Traffic Filtering . . . 24
10. Traffic Monitoring 24
11. IANA Considerations 24
11.1. AFI/SAFI Definitions 24
11.2. Flow Component Definitions 25
11.3. Extended Community Flow Specification Actions 26
12. Security Considerations 28
13. Original authors 29
14. Acknowledgements 29
15. References 29
15.1. Normative References 29
15.2. Informative References 31
15.3. URIs 31
Appendix A. Comparison with RFC 5575 32
Authors' Addresses 32

1. Introduction

Modern IP routers contain both the capability to forward traffic according to IP prefixes as well as to classify, shape, rate limit, filter, or redirect packets based on administratively defined policies.

These traffic policy mechanisms allow the router to define match rules that operate on multiple fields of the packet header. Actions such as the ones described above can be associated with each rule.

The n-tuple consisting of the matching criteria defines an aggregate traffic Flow Specification. The matching criteria can include elements such as source and destination address prefixes, IP protocol, and transport protocol port numbers.

This document defines a general procedure to encode flow specification rules for aggregated traffic flows so that they can be distributed as a BGP [RFC4271] NLRI. Additionally, we define the required mechanisms to utilize this definition to the problem of immediate concern to the authors: intra- and inter-provider distribution of traffic filtering rules to filter (distributed) denial-of-service (DoS) attacks.

By expanding routing information with Flow Specifications, the routing system can take advantage of the ACL (Access Control List) or firewall capabilities in the router's forwarding path. Flow specifications can be seen as more specific routing entries to a unicast prefix and are expected to depend upon the existing unicast data information.

A Flow Specification received from an external autonomous system will need to be validated against unicast routing before being accepted. If the aggregate traffic flow defined by the unicast destination prefix is forwarded to a given BGP peer, then the local system can safely install more specific flow rules that may result in different forwarding behavior, as requested by this system.

The key technology components required to address the class of problems targeted by this document are:

1. Efficient point-to-multipoint distribution of control plane information.

2. Inter-domain capabilities and routing policy support.

3. Tight integration with unicast routing, for verification purposes.

Items 1 and 2 have already been addressed using BGP for other types of control plane information. Close integration with BGP also makes it feasible to specify a mechanism to automatically verify flow information against unicast routing. These factors are behind the choice of BGP as the carrier of Flow Specification information.

As with previous extensions to BGP, this specification makes it possible to add additional information to Internet routers. These are limited in terms of the maximum number of data elements they can hold as well as the number of events they are able to process in a

given unit of time. The authors believe that, as with previous extensions, service providers will be careful to keep information levels below the maximum capacity of their devices.

In many deployments of BGP Flow Specification, the Flow Specification information has replace existing host length route advertisements.

Experience with previous BGP extensions has also shown that the maximum capacity of BGP speakers has been gradually increased according to expected loads. Taking into account Internet unicast routing as well as additional applications as they gain popularity.

From an operational perspective, the utilization of BGP as the carrier for this information allows a network service provider to reuse both internal route distribution infrastructure (e.g., route reflector or confederation design) and existing external relationships (e.g., inter-domain BGP sessions to a customer network).

While it is certainly possible to address this problem using other mechanisms, this solution has been utilized in deployments because of the substantial advantage of being an incremental addition to already deployed mechanisms.

In current deployments, the information distributed by the flow-spec extension is originated both manually as well as automatically. The latter by systems that are able to detect malicious flows. When automated systems are used, care should be taken to ensure their correctness as well as to limit the number and advertisement rate of flow routes.

This specification defines required protocol extensions to address most common applications of IPv4 unicast and VPNv4 unicast filtering. The same mechanism can be reused and new match criteria added to address similar filtering needs for other BGP address families such as IPv6 families [I-D.ietf-idr-flow-spec-v6],

2. Definitions of Terms Used in This Memo

NLRI - Network Layer Reachability Information.

RIB - Routing Information Base.

Loc-RIB - Local RIB.

AS - Autonomous System number.

VRF - Virtual Routing and Forwarding instance.

PE - Provider Edge router

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 [RFC2119]

3. Flow Specifications

A Flow Specification is an n-tuple consisting of several matching criteria that can be applied to IP traffic. A given IP packet is said to match the defined flow if it matches all the specified criteria.

A given flow may be associated with a set of attributes, depending on the particular application; such attributes may or may not include reachability information (i.e., NEXT_HOP). Well-known or AS-specific community attributes can be used to encode a set of predetermined actions.

A particular application is identified by a specific (Address Family Identifier, Subsequent Address Family Identifier (AFI, SAFI)) pair [RFC4760] and corresponds to a distinct set of RIBs. Those RIBs should be treated independently from each other in order to assure non-interference between distinct applications.

BGP itself treats the NLRI as an opaque key to an entry in its databases. Entries that are placed in the Loc-RIB are then associated with a given set of semantics, which is application dependent. This is consistent with existing BGP applications. For instance, IP unicast routing (AFI=1, SAFI=1) and IP multicast reverse-path information (AFI=1, SAFI=2) are handled by BGP without any particular semantics being associated with them until installed in the Loc-RIB.

Standard BGP policy mechanisms, such as UPDATE filtering by NLRI prefix as well as community matching and manipulation, MUST apply to the Flow Specification defined NLRI-type, especially in an inter-domain environment. Network operators can also control propagation of such routing updates by enabling or disabling the exchange of a particular (AFI, SAFI) pair on a given BGP peering session.

4. Dissemination of IPv4 FLow Specification Information

We define a "Flow Specification" NLRI type (Figure 1) that may include several components such as destination prefix, source prefix, protocol, ports, and others (see Section 4.2 below). This NLRI is treated as an opaque bit string prefix by BGP. Each bit string

identifies a key to a database entry with which a set of attributes can be associated.

This NLRI information is encoded using MP_REACH_NLRI and MP_UNREACH_NLRI attributes as defined in [RFC4760]. Whenever the corresponding application does not require Next-Hop information, this shall be encoded as a 0-octet length Next Hop in the MP_REACH_NLRI attribute and ignored on receipt.

The NLRI field of the MP_REACH_NLRI and MP_UNREACH_NLRI is encoded as a 1- or 2-octet NLRI length field followed by a variable-length NLRI value. The NLRI length is expressed in octets.

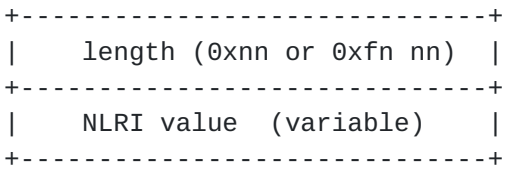


Figure 1: Flow-spec NLRI for IPv4

Implementations wishing to exchange Flow Specification rules MUST use BGP's Capability Advertisement facility to exchange the Multiprotocol Extension Capability Code (Code 1) as defined in [RFC4760]. The (AFI, SAFI) pair carried in the Multiprotocol Extension Capability MUST be the same as the one used to identify a particular application that uses this NLRI-type.

4.1. Length Encoding

- If the NLRI length value is smaller than 240 (0xf0 hex), the length field can be encoded as a single octet.

- Otherwise, it is encoded as an extended-length 2-octet value in which the most significant nibble of the first byte is all ones.

In figure 1 above, values less-than 240 are encoded using two hex digits (0xnn). Values above 239 are encoded using 3 hex digits (0xfnnn). The highest value that can be represented with this encoding is 4095. The value 241 is encoded as 0xf0f1.

4.2. NLRI Value Encoding

The Flow Specification NLRI-type consists of several optional subcomponents. A specific packet is considered to match the flow specification when it matches the intersection (AND) of all the components present in the specification.

The encoding of each of the NLRI components begins with a type field (1 octet) followed by a variable length parameter. Section 4.2.1 to Section 4.2.12 define component types and parameter encodings for the IPv4 IP layer and transport layer headers. IPv6 NLRI component types are described in [I-D.ietf-idr-flow-spec-v6].

Flow Specification components must follow strict type ordering by increasing numerical order. A given component type may or may not be present in the specification, but if present, it MUST precede any component of higher numeric type value.

All combinations of component types within a single NLRI are allowed, even if the combination makes no sense from a semantical perspective. If a given component type within a prefix in unknown, the prefix in question cannot be used for traffic filtering purposes by the receiver. Since a Flow Specification has the semantics of a logical AND of all components, if a component is FALSE, by definition it cannot be applied. However, for the purposes of BGP route propagation, this prefix should still be transmitted since BGP route distribution is independent on NLRI semantics.

The <type, value> encoding is chosen in order to allow for future extensibility.

4.2.1. Type 1 - Destination Prefix

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

Defines: the destination prefix to match. Prefixes are encoded as in BGP UPDATE messages, a length in bits is followed by enough octets to contain the prefix information.

4.2.2. Type 2 - Source Prefix

Encoding: <type (1 octet), prefix-length (1 octet), prefix>

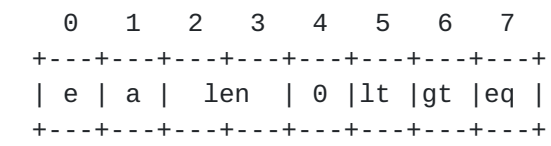
Defines the source prefix to match.

4.2.3. Type 3 - IP Protocol

Encoding:<type (1 octet), [op, value]+>

Contains a set of {operator, value} pairs that are used to match the IP protocol value byte in IP packets.

The operator byte is encoded as:



Numeric operator

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

a - AND bit. If unset, the previous term is logically ORed with the current one. If set, the operation is a logical AND. It should be unset in the first operator byte of a sequence. The AND operator has higher priority than OR for the purposes of evaluating logical expressions.

len - length of the value field for this operand encodes 1 (00) - 4 (11) bytes. Type 3 flow component values are always encoded as single byte (len = 00).

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".

lt	gt	eq	Resulting operation
0	0	0	true (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	false (independent of the value)

Table 1: Comparison operation combinations

4.2.4. Type 4 - Port

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs that matches source OR destination TCP/UDP ports. This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded as 1- or 2-byte quantities.

Port, source port, and destination port components evaluate to FALSE if the IP protocol field of the packet has a value other than TCP or UDP, if the packet is fragmented and this is not the first fragment, or if the system in 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.

4.2.5. Type 5 - Destination Port

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match the destination port of a TCP or UDP packet. This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded as 1- or 2-byte quantities.

4.2.6. Type 6 - Source Port

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match the source port of a TCP or UDP packet. This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded as 1- or 2-byte quantities.

4.2.7. Type 7 - ICMP type

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match the type field of an ICMP packet. This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded using a single byte.

The ICMP type specifiers evaluate to FALSE whenever the protocol value is not ICMP.

4.2.8. Type 8 - ICMP code

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match the code field of an ICMP packet. This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded using a single byte.

The ICMP code specifiers evaluate to FALSE whenever the protocol value is not ICMP.

4.2.9. Type 9 - TCP flags

Encoding:<type (1 octet), [op, bitmask]+>

Bitmask values can be encoded as a 1- or 2-byte bitmask. When a single byte is specified, it matches byte 13 of the TCP header [RFC0793], which contains bits 8 though 15 of the 4th 32-bit word. When a 2-byte encoding is used, it matches bytes 12 and 13 of the TCP header with the data offset field having a "don't care" value.

This component evaluates to FALSE for packets that are not TCP packets.

This type uses the bitmask operand format, which differs from the numeric operator format in the lower nibble.

```
 0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
| e | a |  len  | 0 | 0 |not| m |
+---+---+---+---+---+---+---+---+
```

Bitmask format

e, a, len - Most significant nibble: (end-of-list bit, AND bit, and length field), as defined for in the numeric operator format in Section 4.2.3.

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

4.2.10. Type 10 - Packet length

Encoding:<type (1 octet), [op, bitmask]+>

Defines a list of {operator, value} pairs used to match on the total IP packet length (excluding Layer 2 but including IP header). This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded using 1- or 2-byte quantities.

4.2.11. Type 11 - DSCP (Diffserv Code Point)

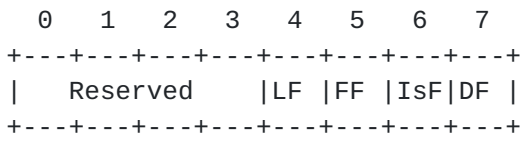
Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match the 6-bit DSCP field [RFC2474]. This list is encoded using the numeric operator format defined in Section 4.2.3. Values are encoded using a single byte. The two most significant bits are zero and the six least significant bits contain the DSCP value.

4.2.12. Type 12 - Fragment

Encoding:<type (1 octet), [op, bitmask]+>

Uses bitmask operand format defined in Section 4.2.9.



Bitmask values:

Bit 7 - Don't fragment (DF)

Bit 6 - Is a fragment (IsF)

Bit 5 - First fragment (FF)

Bit 4 - Last fragment (LF)

4.3. Examples of Encodings

An example of a Flow Specification encoding for: "all packets to 10.0.1/24 and TCP port 25".

```
+------------------+----------+----------+
| destination      | proto    | port     |
+------------------+----------+----------+
| 0x01 18 0a 00 01 | 03 81 06 | 04 81 19 |
+------------------+----------+----------+
```

Decode for protocol:

```
+-------+----------+------------------------------+
| Value |          |                              |
+-------+----------+------------------------------+
|  0x03 | type     |                              |
|  0x81 | operator | end-of-list, value size=1, = |
|  0x06 | value    |                              |
+-------+----------+------------------------------+
```

An example of a Flow Specification encoding for: "all packets to 10.1.1/24 from 192/8 and port {range [137, 139] or 8080}".

```
+------------------+----------+-------------------------+
| destination      | source   | port                    |
+------------------+----------+-------------------------+
| 0x01 18 0a 01 01 | 02 08 c0 | 04 03 89 45 8b 91 1f 90 |
+------------------+----------+-------------------------+
```

Decode for port:

```
+--------+----------+------------------------------+
|  Value |          |                              |
+--------+----------+------------------------------+
|   0x04 | type     |                              |
|   0x03 | operator | size=1, >=                   |
|   0x89 | value    | 137                          |
|   0x45 | operator | "AND", value size=1, <=      |
|   0x8b | value    | 139                          |
|   0x91 | operator | end-of-list, value-size=2, = |
| 0x1f90 | value    | 8080                         |
+--------+----------+------------------------------+
```

This constitutes an NLRI with an NLRI length of 16 octets.

5. Traffic Filtering

Traffic filtering policies have been traditionally considered to be relatively static. Limitations of the static mechanisms caused this mechanism to be designed for the three new applications of traffic filtering (prevention of traffic-based, denial-of-service (DOS)

attacks, traffic filtering in the context of BGP/MPLS VPN service, and centralized traffic control for SDN/NFV networks) requires coordination among service providers and/or coordination among the AS within a service provider. Section 8 has details on the limitation of previous mechanisms and why BGP Flow Specification version 1 provides a solution for to prevent DOS and aid BGP/MPLS VPN filtering rules.

This Flow Specification NLRI defined above to convey information about traffic filtering rules for traffic that should be discarded or handled in manner specified by a set of pre-defined actions (which are defined in BGP Extended Communities). This mechanism is primarily designed to allow an upstream autonomous system to perform inbound filtering in their ingress routers of traffic that a given downstream AS wishes to drop.

In order to achieve this goal, this draft specifies two application specific NLRI identifiers that provide traffic filters, and a set of actions encoding in BGP Extended Communities. The two application specific NLRI identifiers are:

- IPv4 Flow Specification identifier (AFI=1, SAFI=133) along with specific semantic rules for IPv4 routes, and

- BGP NLRI type (AFI=1, SAFI=134) value, which can be used to propagate traffic filtering information in a BGP/MPLS VPN environment.

Distribution of the IPv4 Flow Specification is described in section 6, and distibution of BGP/MPLS traffic Flow Specification is described in section 8. The traffic filtering actions are described in section 7.

5.1. Ordering of Traffic Filtering Rules

With traffic filtering rules, more than one rule may match a particular traffic flow. Thus, it is necessary to define the order at which rules get matched and applied to a particular traffic flow. This ordering function must be such that it must not depend on the arrival order of the Flow Specification's rules and must be consistent in the network.

The relative order of two Flow Specification rules is determined by comparing their respective components. The algorithm starts by comparing the left-most components of the rules. If the types differ, the rule with lowest numeric type value has higher precedence (and thus will match before) than the rule 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 the prefixes overlap, the one with the longer prefix-length has higher precedence. If they do not overlap 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 the ISO C standard. 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.

The code below shows a Python3 implementation of the comparison algorithm. The full code was tested with Python 3.6.3 and can be obtained at https://github.com/stoffi92/flowspec-cmp [1].

```
import itertools
import ipaddress

def flow_rule_cmp(a, b):
    for comp_a, comp_b in itertools.zip_longest(a.components,
                                                b.components):
        # If a component type does not exist in one rule
        # this rule has lower precedence
        if not comp_a:
            return B_HAS_PRECEDENCE
        if not comp_b:
            return A_HAS_PRECEDENCE
        # higher precedence for lower component type
        if comp_a.component_type < comp_b.component_type:
            return A_HAS_PRECEDENCE
        if comp_a.component_type > comp_b.component_type:
            return B_HAS_PRECEDENCE
        # component types are equal -> type specific comparison
        if comp_a.component_type in (IP_DESTINATION, IP_SOURCE):
            # assuming comp_a.value, comp_b.value of type ipaddress
            if comp_a.value.overlaps(comp_b.value):
                # longest prefixlen has precedence
                if comp_a.value.prefixlen > comp_b.value.prefixlen:
                    return A_HAS_PRECEDENCE
                if comp_a.value.prefixlen < comp_b.value.prefixlen:
                    return B_HAS_PRECEDENCE
```

```
                # components equal -> continue with next component
            elif comp_a.value > comp_b.value:
                return B_HAS_PRECEDENCE
            elif comp_a.value < comp_b.value:
                return A_HAS_PRECEDENCE
        else:
            # assuming comp_a.value, comp_b.value of type bytearray
            if len(comp_a.value) == len(comp_b.value):
                if comp_a.value > comp_b.value:
                    return B_HAS_PRECEDENCE
                if comp_a.value < comp_b.value:
                    return A_HAS_PRECEDENCE
                # components equal -> continue with next component
            else:
                common = min(len(comp_a.value), len(comp_b.value))
                if comp_a.value[:common] > comp_b.value[:common]:
                    return B_HAS_PRECEDENCE
                elif comp_a.value[:common] < comp_b.value[:common]:
                    return A_HAS_PRECEDENCE
                # the first common bytes match
                elif len(comp_a.value) > len(comp_b.value):
                    return A_HAS_PRECEDENCE
                else:
                    return B_HAS_PRECEDENCE
    return EQUAL
```

6. Validation Procedure

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 two routes for the same Flow Specification 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 non-feasible. In the context of IP routing information, this step is used to validate that the NEXT_HOP attribute of a given route is resolvable.

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

A Flow Specification NLRI must be validated such that it is considered feasible if and only if:

a) The originator of the Flow Specification matches the originator of the best-match unicast route for the destination prefix embedded in the Flow Specification.

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

By originator of a BGP route, we mean either the BGP originator path attribute, as used by route reflection, or the transport address of the BGP peer, if this path attribute is not present.

BGP implementations MUST also enforce that the AS_PATH attribute of a route received via the External Border Gateway Protocol (eBGP) contains the neighboring AS in the left-most position of the AS_PATH attribute. While this rule is optional in the BGP specification, it becomes necessary to enforce it for security reasons.

The best-match unicast route may change over the time independently of the Flow Specification NLRI. Therefore, a revalidation of the Flow Specification NLRI MUST be performed whenever unicast routes change. Revalidation is defined as retesting that clause a and clause b above are true.

Explanation:

The underlying concept is that the neighboring AS that advertises the best unicast route for a destination is allowed to advertise flow-spec information that conveys a more or equally specific destination prefix. Thus, as long as there are no more specific unicast routes, received from a different neighboring AS, which would be affected by that filtering rule.

The neighboring AS is the immediate destination of the traffic described by the Flow Specification. If it requests these flows to be dropped, that request can be honored without concern that it represents a denial of service in itself. Supposedly, the traffic is being dropped by the downstream autonomous system, and there is no added value in carrying the traffic to it.

7. Traffic Filtering Actions

This specification defines a minimum set of filtering actions that it standardizes as BGP extended community values [RFC4360]. This is not meant to be an inclusive list of all the possible actions, but only a subset that can be interpreted consistently across the network.

Additional actions can be defined as either requiring standards or as vendor specific.

Implementations SHOULD provide mechanisms that map an arbitrary BGP community value (normal or extended) to filtering actions that require different mappings in different systems in the network. For instance, providing packets with a worse-than-best-effort, per-hop behavior is a functionality that is likely to be implemented differently in different systems and for which no standard behavior is currently known. Rather than attempting to define it here, this can be accomplished by mapping a user-defined community value to platform-/network-specific behavior via user configuration.

The default action for a traffic filtering Flow Specification is to accept IP traffic that matches that particular rule.

This document defines the following extended communities values shown in Table 2 in the form 0x8xnn where nn indicates the sub-type. Encodings for these extended communities are described below.

community	action	encoding
0x8006	traffic-rate-bytes	2-byte ASN, 4-byte float
TBD	traffic-rate-packets	2-byte ASN, 4-byte float
0x8007	traffic-action	bitmask
0x8008	rt-redirect AS-2byte	2-octet AS, 4-octet value
0x8108	rt-redirect IPv4	4-octet IPv4 addres, 2-octet value
0x8208	rt-redirect AS-4byte	4-octet AS, 2-octet value
0x8009	traffic-marking	DSCP value

Table 2: Traffic Action Extended Communities

Some traffic action communities may interfere with each other. Section 7.6 of this specification provides rules for handling interference between specific types of traffic actions, and error handling based on [RFC7606]. Any additional definition of a traffic actions specified by additional standards documents or vendor documents MUST specify if the traffic action interacts with an existing traffic actions, and provide error handling per [RFC7606].

Multiple traffic actions may be present for a single NLRI. The traffic actions are processed in ascending order of the sub-type found in the BGP Extended Communities. If not all of them can be processed the filter SHALL NOT be applied at all (for example: if for a given flow there are the action communities rate-limit-bytes and

traffic-marking attached, and the plattform does not support one of them also the other shall not be applied for that flow).

All traffic actions are specified as transitive BGP Extended Communities.

7.1. Traffic Rate in Bytes (traffic-rate-bytes) sub-type 0x06

The traffic-rate-bytes extended community uses the following extended community encoding:

The first two octets carry the 2-octet id, which can be assigned from a 2-byte AS number. When a 4-byte AS number is locally present, the 2 least significant bytes of such an AS number can be used. This value is purely informational and should not be interpreted by the implementation.

The remaining 4 octets carry the maximum rate information in IEEE floating point [IEEE.754.1985] format, units being bytes per second. A traffic-rate of 0 should result on all traffic for the particular flow to be discarded.

Interferes with: No other BGP Flow Specification traffic action in this document.

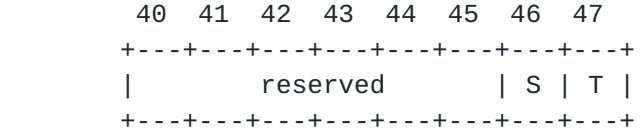
7.2. Traffic Rate in Packets (traffic-rate-packets) sub-type TBD

The traffic-rate-packets extended community uses the same encoding as the traffic-rate-bytes extended community. The floating point value carries the maximum packet rate in packets per second. A traffic-rate-packets of 0 should result in all traffic for the particular flow to be discarded.

Interferes with: No other BGP Flow Specification traffic action in this document.

7.3. Traffic-action (traffic-action) sub-type 0x07

The traffic-action extended community consists of 6 bytes of which only the 2 least significant bits of the 6th byte (from left to right) are currently defined.



where S and T are defined as:

- T: Terminal Action (bit 47): When this bit is set, the traffic filtering engine will apply any subsequent filtering rules (as defined by the ordering procedure). If not set, the evaluation of the traffic filter stops when this rule is applied.

- S: Sample (bit 46): Enables traffic sampling and logging for this Flow Specification.

- reserved: should always be set to 0 by the originator and not be evaluated by the receiving BGP speaker.

The use of the Terminal Action (bit 47) may result in more than one filter-rule matching a particular flow. All the flow actions from these rules shall be collected and applied. If interfering actions have been collected only the first occurence SHALL be applied. However, if a single rule contains interfering actions this rule SHALL still be handled as described in Section 7.6.

Interferes with: No other BGP Flow Specification traffic action in this document.

7.4. RT Redirect (rt-redirect) sub-type 0x08

The redirect extended community allows the traffic to be redirected to a VRF routing instance that lists the specified route-target in its import policy. If several local instances match this criteria, the choice between them is a local matter (for example, the instance with the lowest Route Distinguisher value can be elected). This extended community allows 3 different encodings formats for the route-target (type 0x80, 0x81, 0x82). Is uses the same encoding as the Route Target extended community [RFC4360].

It should be noted that the low-order nibble of the Redirect's Type field corresponds to the Route Target Extended Community format field (Type). (See Sections 3.1, 3.2, and 4 of [RFC4360] plus Section 2 of [RFC5668].) The low-order octet (Sub-Type) of the Redirect Extended Community remains 0x08 for all three encodings of the BGP Extended Communities (AS 2-byte, AS 4-byte, and IPv4 address).

Interferes with: All other redirect functions. All redirect functions are mutually exclusive. If this redirect function exists, then no other redirect functions can be processed.

7.5. Traffic Marking (traffic-marking) sub-type 0x09

The traffic marking extended community instructs a system to modify the DSCP bits of a transiting IP packet to the corresponding value. This extended community is encoded as a sequence of 5 zero bytes

followed by the DSCP value encoded in the 6 least significant bits of 6th byte.

Interferes with: No other BGP Flow Specification traffic action in this document.

7.6. Rules on Traffic Action Interference

Traffic actions may interfere with each other. If interfering traffic actions are present for a single Flow Specification NLRI the entire Flow Specification (irrespective if there are any other non conflicting actions associated with the same Flow Specification) SHALL be treated as BGP WITHDRAW.

This document defines 7 traffic actions which are interfering in the following way:

1. Redirect-action-communities (0x8008, 0x8108, 0x8208):

   The three redirect-communities are mutually exclusive. Only a single redirect community may be associated with a Flow Specification otherwise they are interfering.

2. All traffic-action communities (including redirect-actions):

   Multiple occurences of the same (sub-type and type) traffic-action associated with a Flow Specification are always interfering.

When a traffic action is defined in a standards document the handling of interaction with other/same traffic actions MUST be defined as well. Invalid interactions between actions SHOULD NOT trigger a BGP NOTIFICATION. All error handling for error conditions based on [RFC7606].

7.6.1. Examples

(rt-redirect vrf-a, rt-redirect vrf-b, traffic-rate-bytes 1Mbit/s)

RT-redirect vrf-a and rt-redirect vrf-b are interfering: The BGP UPDATE is treated as WITHDRAW.

(rt-redirect vrf-a, traffic-rate-bytes 1Mbit/s, traffic-rate-bytes 2Mbit/s)

Duplicate traffic-rate-bytes are interfering: The BGP UPDATE is treated as WITHDRAW.

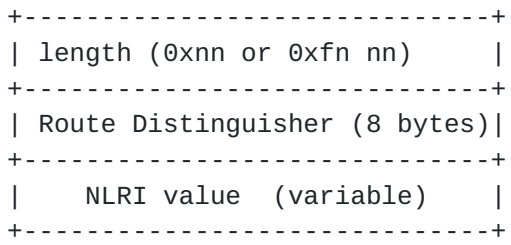
(rt-redirect vrf-a, traffic-rate-bytes 1Mbit/s, traffic-rate-packets 1000)

No interfering action communities: The BGP UPDATE is subject to further processing.

8. Dissemination of Traffic Filtering in BGP/MPLS VPN Networks

Provider-based Layer 3 VPN networks, such as the ones using a BGP/MPLS IP VPN [RFC4364] control plane, may have different traffic filtering requirements than Internet service providers. But also Internet service providers may use those VPNs for scenarios like having the Internet routing table in a VRF, resulting in the same traffic filtering requirements as defined for the global routing table environment within this document. This document proposes an additional BGP NLRI type (AFI=1, SAFI=134) value, which can be used to propagate traffic filtering information in a BGP/MPLS VPN environment.

The NLRI format for this address family consists of a fixed-length Route Distinguisher field (8 bytes) followed by a Flow Specification, following the encoding defined above in Section 4.2 of this document. The NLRI length field shall include both the 8 bytes of the Route Distinguisher as well as the subsequent Flow Specification.



Flow-spec NLRI for MPLS

Propagation of this NLRI is controlled by matching Route Target extended communities associated with the BGP path advertisement with the VRF import policy, using the same mechanism as described in "BGP/MPLS IP VPNs" [RFC4364].

Flow Specification rules received via this NLRI apply only to traffic that belongs to the VRF(s) in which it is imported. By default, traffic received from a remote PE is switched via an MPLS forwarding decision and is not subject to filtering.

Contrary to the behavior specified for the non-VPN NLRI, flow rules are accepted by default, when received from remote PE routers.

8.1. Validation Procedures for BGP/MPLS VPNs

The validation procedures are the same as for IPv4.

8.2. Traffic Actions Rules

The traffic action rules are the same as for IPv4.

9. Limitations of Previous Traffic Filtering Efforts

9.1. Limitations in Previous DDoS Traffic Filtering Efforts

The popularity of traffic-based, denial-of-service (DoS) attacks, which often requires the network operator to be able to use traffic filters for detection and mitigation, brings with it requirements that are not fully satisfied by existing tools.

Increasingly, DoS mitigation requires coordination among several service providers in order to be able to identify traffic source(s) and because the volumes of traffic may be such that they will otherwise significantly affect the performance of the network.

Several techniques are currently used to control traffic filtering of DoS attacks. Among those, one of the most common is to inject unicast route advertisements corresponding to a destination prefix being attacked (commonly known as remote triggered blackhole RTBH). One variant of this technique marks such route advertisements with a community that gets translated into a discard Next-Hop by the receiving router. Other variants attract traffic to a particular node that serves as a deterministic drop point.

Using unicast routing advertisements to distribute traffic filtering information has the advantage of using the existing infrastructure and inter-AS communication channels. This can allow, for instance, a service provider to accept filtering requests from customers for address space they own.

There are several drawbacks, however. An issue that is immediately apparent is the granularity of filtering control: only destination prefixes may be specified. Another area of concern is the fact that filtering information is intermingled with routing information.

The mechanism defined in this document is designed to address these limitations. We use the Flow Specification NLRI defined above to convey information about traffic filtering rules for traffic that is subject to modified forwarding behavior (actions). The actions are defined as extended communities and include (but are not limited to)

rate-limiting (including discard), traffic redirection, packet rewriting.

9.2. Limitations in Previous BGP/MPLS Traffic Filtering

Provider-based Layer 3 VPN networks, such as the ones using a BGP/MPLS IP VPN [RFC4364] control plane, may have different traffic filtering requirements than Internet service providers.

In these environments, the VPN customer network often has traffic filtering capabilities towards their external network connections (e.g., firewall facing public network connection). Less common is the presence of traffic filtering capabilities between different VPN attachment sites. In an any-to-any connectivity model, which is the default, this means that site-to-site traffic is unfiltered.

In circumstances where a security threat does get propagated inside the VPN customer network, there may not be readily available mechanisms to provide mitigation via traffic filter.

But also Internet service providers may use those VPNs for scenarios like having the Internet routing table in a VRF. Therefore, limitations described in Section 9.1 also apply to this section.

The BGP Flow Specification version 1 addresses these limitations.

10. Traffic Monitoring

Traffic filtering applications require monitoring and traffic statistics facilities. While this is an implementation-specific choice, implementations SHOULD provide:

- A mechanism to log the packet header of filtered traffic.
- A mechanism to count the number of matches for a given flow specification rule.

11. IANA Considerations

This section complies with [RFC7153].

11.1. AFI/SAFI Definitions

IANA maintains a registry entitled "SAFI Values". For the purpose of this work, IANA updated the registry and allocated two additional SAFIs:

Value	Name	Reference
133	IPv4 dissemination of Flow Specification rules	[this document]
134	VPNv4 dissemination of Flow Specification rules	[this document]

Table 3: Registry: SAFI Values

11.2. Flow Component Definitions

A Flow Specification consists of a sequence of flow components, which are identified by a an 8-bit component type. IANA has created and maintains a registry entitled "Flow Spec Component Types". This document defines the following Component Type Codes:

Value	Name	Reference
1	Destination Prefix	[this document]
2	Source Prefix	[this document]
3	IP Protocol	[this document]
4	Port	[this document]
5	Destination port	[this document]
6	Source port	[this document]
7	ICMP type	[this document]
8	ICMP code	[this document]
9	TCP flags	[this document]
10	Packet length	[this document]
11	DSCP	[this document]
12	Fragment	[this document]

Table 4: Registry: Flow Spec Component Types

In order to manage the limited number space and accommodate several usages, the following policies defined by [RFC5226] used:

Range	Policy
0	Invalid value
[1 .. 12]	Defined by this specification
[13 .. 127]	Specification required
[128 .. 255]	First Come First Served

Table 5: Flow Spec Component Types Policies

The specification of a particular "Flow Spec Component Type" must clearly identify what the criteria used to match packets forwarded by the router is. This criteria should be meaningful across router hops and not depend on values that change hop-by-hop such as TTL or Layer 2 encapsulation.

11.3. Extended Community Flow Specification Actions

The Extended Community Flow Specification Action types defined in this document consist of two parts:

Type (BGP Transitive Extended Community Type)

Sub-Type

For the type-part, IANA maintains a registry entitled "BGP Transitive Extended Community Types". For the purpose of this work (Section 7), IANA updated the registry to contain the values listed below:

Sub-Type Value	Name	Reference
0x80	Generic Transitive Experimental Use Extended Community (Sub-Types are defined in the "Generic Transitive Experimental Use Extended Community Sub-Types" registry)	[RFC7153]
0x81	Generic Transitive Experimental Use Extended Community Part 2 (Sub-Types are defined in the "Generic Transitive Experimental Use Extended Community Part 2 Sub-Types" Registry)	[this document] [See Note-1]
0x82	Generic Transitive Experimental Use Extended Community Part 3 (Sub-Types are defined in the "Generic Transitive Experimental Use Extended Community Part 3 Sub-Types" Registry)	[this document] [See Note-1]

Table 6: Registry: Generic Transitive Experimental Use Extended Community Types

Note-1: This document replaces [RFC7674].

For the sub-type part of the extended community actions IANA maintains and updated the following registries:

Sub-Type Value	Name	Reference
0x06	Flow spec traffic-rate-bytes	[this document]
TBD	Flow spec traffic-rate-packets	[this document]
0x07	Flow spec traffic-action (Use of the "Value" field is defined in the "Traffic Action Fields" registry)	[this document] [See Note-2]
0x08	Flow spec rt-redirect AS-2byte format	[this document]
0x09	Flow spec traffic-remarking	[this document]

Table 7: Registry: Generic Transitive Experimental Use Extended Community Sub-Types

Note-2: This document replaces both [RFC7674] and [RFC5575].

Sub-Type Value	Name	Reference
0x08	Flow spec rt-redirect IPv4 format	[this document] [See Note-3]

Table 8: Registry: Generic Transitive Experimental Use Extended Community Part 2 Sub-Types

Sub-Type Value	Name	Reference
0x08	Flow spec rt-redirect AS-4byte format	[this document] [See Note-3]

Table 9: Registry: Generic Transitive Experimental Use Extended Community Part 3 Sub-Types

Note-3: This document replaces [RFC7674], and becomes the only reference for this table.

The "traffic-action" extended community (Section 7.3) defined in this document has 46 unused bits, which can be used to convey additional meaning. IANA created and maintains a new registry entitled: "Traffic Action Fields". These values should be assigned via IETF Review rules only. The following traffic-action fields have been allocated:

Bit	Name	Reference
47	Terminal Action	[this document]
46	Sample	[this document]

Table 10: Registry: Traffic Action Fields

12. Security Considerations

Inter-provider routing is based on a web of trust. Neighboring autonomous systems are trusted to advertise valid reachability information. If this trust model is violated, a neighboring

autonomous system may cause a denial-of-service attack by advertising reachability information for a given prefix for which it does not provide service.

As long as traffic filtering rules are restricted to match the corresponding unicast routing paths for the relevant prefixes, the security characteristics of this proposal are equivalent to the existing security properties of BGP unicast routing.

Where it is not the case, this would open the door to further denial-of-service attacks.

Enabling firewall-like capabilities in routers without centralized management could make certain failures harder to diagnose. For example, it is possible to allow TCP packets to pass between a pair of addresses but not ICMP packets. It is also possible to permit packets smaller than 900 or greater than 1000 bytes to pass between a pair of addresses, but not packets whose length is in the range 900-1000. Such behavior may be confusing and these capabilities should be used with care whether manually configured or coordinated through the protocol extensions described in this document.

13. Original authors

Barry Greene, MuPedro Marques, Jared Mauch, Danny McPherson, and Nischal Sheth were authors on [RFC5575], and therefore are contributing authors on this document.

14. Acknowledgements

The authors would like to thank Yakov Rekhter, Dennis Ferguson, Chris Morrow, Charlie Kaufman, and David Smith for their comments for the comments on the original [RFC5575]. Chaitanya Kodeboyina helped design the flow validation procedure; and Steven Lin and Jim Washburn ironed out all the details necessary to produce a working implementation in the original [RFC5575].

Additional the authors would like to thank Alexander Mayrhofer, Nicolas Fevrier and Job Snijders for their comments and review.

15. References

15.1. Normative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[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>.

[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>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[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>.

[RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <https://www.rfc-editor.org/info/rfc4761>.

[RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <https://www.rfc-editor.org/info/rfc4762>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <https://www.rfc-editor.org/info/rfc5575>.

[RFC5668] Rekhter, Y., Sangli, S., and D. Tappan, "4-Octet AS Specific BGP Extended Community", RFC 5668, DOI 10.17487/RFC5668, October 2009, <https://www.rfc-editor.org/info/rfc5668>.

[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>.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, DOI 10.17487/RFC6482, February 2012, <https://www.rfc-editor.org/info/rfc6482>.

[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>.

[RFC7674] Haas, J., Ed., "Clarification of the Flowspec Redirect Extended Community", RFC 7674, DOI 10.17487/RFC7674, October 2015, <https://www.rfc-editor.org/info/rfc7674>.

15.2. Informative References

[I-D.ietf-idr-flow-spec-v6] McPherson, D., Raszuk, R., Pithawala, B., akarch@cisco.com, a., and S. Hares, "Dissemination of Flow Specification Rules for IPv6", draft-ietf-idr-flow-spec-v6-08 (work in progress), March 2017.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

15.3. URIs

[1] https://github.com/stoffi92/flowspec-cmp

Appendix A. Comparison with RFC 5575

This document includes numerous editorial changes to [RFC5575]. It is recommended to read the entire document. The authors, however want to point out the following technical changes to [RFC5575]:

Section 4.2.3 defines a numeric operator and comparison bit combinations. In [RFC5575] the meaning of those bit combination was not explicitly defined and left open to the reader.

Section 4.2.3 - Section 4.2.8, Section 4.2.10, Section 4.2.11 make use of the above numeric operator. The allowed length of the comparison value was not consistently defined in [RFC5575].

Section 7 defines all traffic action extended communities as transitive extended communities. [RFC5575] defined the traffic-rate action to be non-transitive and did not define the transitivity of the other action communities at all.

Section 7.2 introduces a new traffic filtering action (traffic-rate-packets). This action did not exist in [RFC5575].

Section 7.4 contains the same redirect actions already defined in [RFC5575] however, these actions have been renamed to "rt-redirect" to make it clearer that the redirection is based on route-target.

Section 7.6 introduces rules how updates of Flow Specifications shall be handled in case they contain interfering actions. Section 7.3 also cross-references this section. [RFC5575] did not define this.

Authors' Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Email: shares@ndzh.com

Christoph Loibl
Next Layer Communications
Mariahilfer Guertel 37/7
Vienna 1150
AT

Phone: +43 664 1176414
Email: cl@tix.at

Robert Raszuk
Bloomberg LP
731 Lexington Ave
New York City, NY 10022
USA

Email: robert@raszuk.net

Danny McPherson
Verisign
USA

Email: dmcpherson@verisign.com

Martin Bacher
T-Mobile Austria
Rennweg 97-99
Vienna 1030
AT

Email: mb.ietf@gmail.com