

IDR Working Group
Internet-Draft
Intended status: Standards Track
Expires: 18 September 2025

R. Raszuk, Ed.
Arrcus
J. Haas, Ed.
Juniper Networks
A. Lange, Ed.
Nokia
B. Decraene, Ed.
Orange
S. Amante
Apple
P. Jakma
Huawei Ireland Research Centre
17 March 2025

BGP Community Container Attribute
draft-ietf-idr-wide-bgp-communities-12

Abstract

Route tagging plays an important role in external BGP [RFC4271] relations, in communicating various routing policies between peers. It is also a common best practice among operators to propagate various additional information about routes intra-domain. The most common tool used today to attach various information about routes is through the use of BGP communities [RFC1997].

This document defines a new encoding that enhances and simplifies what can be accomplished with BGP communities. This specification's most important addition is the ability to specify and advertise an operator's parameters for execution. It also provides an extensible platform for any future community encoding requirements.

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.

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 18 September 2025.

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	3
2. Protocol Summary	4
2.1. BGP Community Container Common Header	4
2.2. Community Containers	4
2.3. BGP Community Container Atoms	4
2.4. Propagation of BGP Community Containers	5
3. BGP Community Container Attribute	5
3.1. BGP Community Container Attribute Common Header	5
4. BGP Community Container, Type 1: BGP Wide Community	7
4.1. Community Value	7
4.2. Source AS Number	8
4.3. Context AS Number	8
4.4. BGP Wide Community TLVs	8
4.4.1. Sub-Type 1, BGP Wide Community Targets TLV	9
4.4.2. Sub-Type 2, BGP Wide Community Exclude Targets TLV	9
4.4.3. Sub-Type 3, BGP Wide Community Parameters TLV	10
5. BGP Community Container Atoms	10
5.1. Atom Type 1, The Autonomous System Number List	11
5.2. Atom Types 2 and 3, The IPv4 and IPv6 Prefix Lists	12
5.3. Atom Type 4, The Integer32 List	12

5.4.	Atom Type 5, The IEEE Floating Point Number List	12
5.5.	Atom Type 6, The Neighbor Class List	12
5.6.	Atom Type 7, The User-defined Class List	13
5.7.	Atom Type 8, the UTF-8 String	13
6.	Well Known Standard BGP Communities	14
7.	Operational Considerations	14
8.	Error Handling	14
8.1.	General Error Handling for BGP Community Containers . . .	14
8.2.	BGP Wide Community Container Error Handling	14
9.	Example	15
9.1.	Example Type 1 Wide Community Definition	15
9.2.	Example Type 1 BGP Wide Community Encoding	15
10.	Security considerations	16
10.1.	BGP Community Container Security Considerations	17
10.2.	BGP Wide Community Security Considerations	17
11.	IANA Considerations	17
11.1.	BGP Community Container Attribute	17
11.2.	BGP Community Container Atoms Types	18
11.3.	BGP Community Container Neighbor Class List Atom Types	18
11.4.	BGP Community Container Values	19
11.5.	Registered Type 1 BGP Wide Communities Container Values	19
11.6.	Registered Type 1 BGP Wide Community TLVs	19
12.	Change History	20
12.1.	Working Group draft	20
12.2.	Individual draft	21
13.	Contributors	22
14.	Acknowledgments	23
15.	References	23
15.1.	Normative References	23
15.2.	Informative References	23
	Authors' Addresses	24

1. Introduction

[RFC1997] defines the BGP Community Attribute. This attribute is used as a tool to carry additional information in BGP routes which may help to automate peering administration. The BGP Communities Attribute consists of a set of one or more four-octet values, where each specifies a different community. Except for two reserved ranges, the encoding of community values mandates that the first two octets are to contain the Autonomous System number, with the next two octets containing some locally defined opaque value.

Since the introduction of [RFC1997], numerous additional mechanisms have been introduced to provide BGP Community-like functionality. Each of these mechanisms introduce a new syntax, typically covered by its encoding with the BGP Path Attribute that defines it, and a semantic space.

The definition of a new BGP Path Attribute, with the ability to contain locally defined parameters enhances the current level of network policies, as well as simplify BGP policy management. The encoding facilitates the delivery of new network services without a need to define a new BGP extension each time.

2. Protocol Summary

This specification defines a new BGP Path Attribute, the BGP Community Container. It carries a series of BGP Community Container types, each prefaced with the BGP Community Container Common Header.

This specification also defines the BGP Wide Community Container.

2.1. BGP Community Container Common Header

The BGP Community Container Common Header permits Community-like attributes to be grouped under a single BGP Path Attribute. This provides a hierarchy for future Community-like features. It permits implementations without knowledge of a specific Community Container's format to address that Community Container by its code point. It also permits common enforcement of the Community Container's transitivity across AS boundaries without need for the implementation to understand a specific Container's implementation.

2.2. Community Containers

This document defines one Community Container (Type 1 - "BGP Wide Community TLVs") with the encoding defined in Section 4.

2.3. BGP Community Container Atoms

Atoms provide data types that can be used to encode contents of BGP Community Containers. They are in the format of TLVs and are defined later in this document in Section 5.

2.4. Propagation of BGP Community Containers

Propagation of BGP Community Containers is scoped to the administrative boundary. The definition of an administrative boundary consists of arbitrary set of connected ASes, possibly under control of a single entity. How such an administrative boundary is determined is out of scope of this document.

3. BGP Community Container Attribute

This document defines a new BGP Path Attribute, the BGP Community Container. The attribute type code is TBD.

The BGP Community Container attribute is an optional, transitive BGP attribute, and may be present only once in the BGP UPDATE message.

The attribute contains a set of typed containers. Any given container type may appear multiple times, unless that container type's definition specifies otherwise. Each such container can have a unique propagation scope and can be subject to local filtering.

3.1. BGP Community Container Attribute Common Header

Containers always start with the following common header:

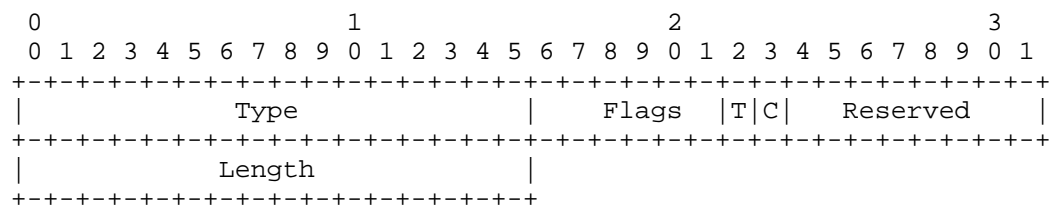


Figure 1: Common container header

The BGP Community Container Common Header contains following encoding:

Container Type:

Container Type 1, BGP Wide Community is defined in this document.

Flags:

Flags control common behavior including the transitivity of the Container.

Length:

Length of the Container header and content.

This document defines container type 1. See the Section 11 for information on additional type registration policies.

Bit	Value	Meaning
T	0	Not Transitive across administrative boundary.
	1	Transitive across AS and administrative boundary.
C	0	Not transitive across confederation boundaries.
	1	Transitive across confederation boundaries.
3..7	-	RESERVED - MUST be zero when originated and SHOULD be ignored upon receipt.

Table 1: Flags

Flags are defined globally and apply to all container types.

Bit 0 (T bit) Transitivity bit:

When not set (value 0), the community in the container is not transitive across administrative boundary.

When set (value 1), the community in the container is transitive across all administrative boundaries.

Bit 1 (C bit) BGP Confederation [RFC3065] bit:

The confederation bit is used to manage the propagation scope of a given container across Member-AS confederation boundary.

When not set (value 0) community is not transitive across confederation Member-AS boundary. When set (value of 1) indicates that community in a given container is transitive across Member-AS confederation boundary.

The Reserved field MUST be set to zero when originated and SHOULD be ignored upon receipt.

The Length field represents the total length (in octets) of a given container including its header.

4. BGP Community Container, Type 1: BGP Wide Community

The Type 1 BGP Community Container, the BGP Wide Community, is of variable size (but minimum length 12). It is composed of a fixed 12-octets - containing the Community Value, the Source AS Number, and the Context AS Number - followed by optional TLVs:

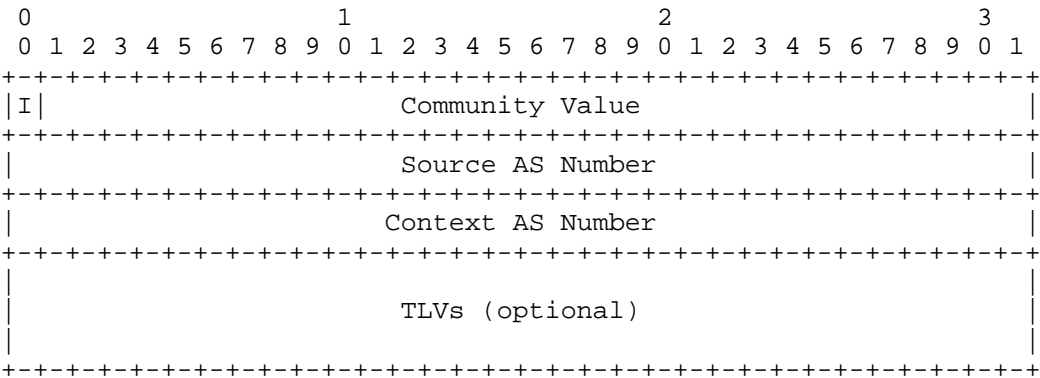


Figure 2: Type 1, BGP Wide Community

4.1. Community Value

Community Value: 4 octets

The Community Value indicates what set of actions a router is requested to take upon reception of a route containing this community. The semantics of this value depend on whether this is a private/local community or IANA registered.

When the high order bit of the Community Value field - I - is set, the value is IANA Registered and has a well defined meaning with underlying semantics. See the documentation for each Registered BGP Wide Community for its semantics and validation requirements.

When the high order bit of the Community Value field is clear, the value is locally defined and has semantics solely within the control of the AS defining that community. The Context AS Number provides the namespace in which this Community Value is interpreted. It is that AS's responsibility to provide the semantics and validation requirements for that BGP Wide Community.

When Community Value is not recognized by the BGP speaker it should ignore it while maintaining propagation rules as specified by T/C bits

See Section 11.5 for code point space partitioning.

4.2. Source AS Number

Source Autonomous System Number: 4 octets

The Autonomous System number indicates the AS originating this BGP Wide Community.

4.3. Context AS Number

Context Autonomous System Number: 4 octets

This identifies the AS that provides the semantics to interpret this Community.

4.4. BGP Wide Community TLVs

Optional type 1 container TLVs are encoded in the following format:

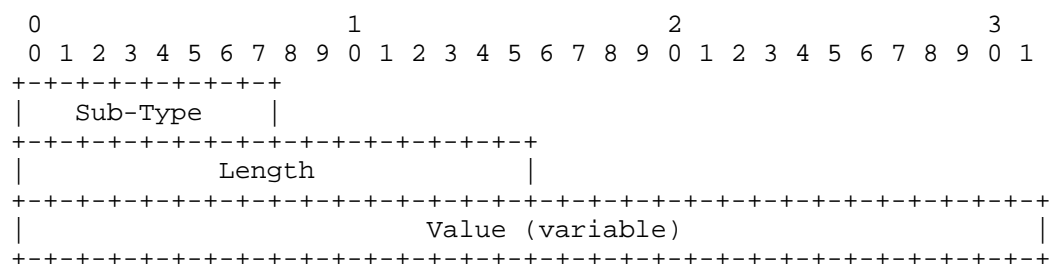


Figure 3: Type 1 Container TLVs

Sub-Type:

The sub-type of the BGP Wide Community TLV. A given Sub-Type MUST NOT appear more than once.

Length:

Length of the "Value" field in octets.

Value:

Specific to the underlying Sub-Type.

4.4.1. Sub-Type 1, BGP Wide Community Targets TLV

The value field of the Wide Community Target(s) TLV (Sub-Type 1) is a series of Atom TLVs. The semantics of any given Atom TLV MUST be part of the definition of a given Wide Community.

BGP Wide Community Targets define the matching criteria for the community. A given Wide Community may have a number of targets that it applies to. The semantics of these targets will vary on a per community basis. Depending on the definition of the community, targets may be optional.

BGP Wide Community Targets consist of a series of Atoms that have "match any" semantics. Thus, if any given target matches per the semantics of that Atom for the community, the community is considered to match and the action defined by the community should be executed.

When no Target(s) TLV is specified, it is considered "match all".

If the semantics of a given Atom is undefined or unrecognized this Atom MUST be ignored.

When no targets are required by the definition of a given Wide Community, the Wide Community Target(s) TLV SHOULD NOT be encoded in the community. Implementations MUST be prepared to accept a Wide Community Target(s) TLV with an empty value field.

4.4.2. Sub-Type 2, BGP Wide Community Exclude Targets TLV

The BGP Wide Community Exclude Target(s) TLV (Sub-Type 2) contains a series of a Atoms.

BGP Wide Community Exclude Targets define criteria by which the community is considered to not match. Depending on the semantics of the BGP Wide Community, Exclude Target(s) may be optional.

The semantic of the BGP Wide Community Exclude Target(s) is to match all targets specified in BGP Wide Community Target(s) TLV with the exception of those listed in the BGP Wide Community Exclude Target(s) TLV.

The value field of the BGP Wide Community Exclude Target(s) TLV is a series of BGP Wide Community Atom TLVs. The semantics of any given Atom TLV MUST be part of the definition of a given Wide Community.

If the semantics of a given Atom is undefined for the community in question, this Atom MUST be ignored.

If the BGP Wide Community Target(s) TLV and the BGP Wide Community Exclude Target(s) TLV result in conflicting actions, priority MUST be given to the Wide Community Exclude Target(s) TLV.

When no exclude targets are required by the definition of a given BGP Wide Community, the BGP Wide Community Exclude Target(s) TLV SHOULD NOT be encoded in the community. Implementations MUST be prepared to accept a BGP Wide Community Exclude Target(s) TLV with an empty value field.

4.4.3. Sub-Type 3, BGP Wide Community Parameters TLV

The BGP Wide Community Parameter(s) TLV (Sub-Type 3) contains an ordered set of Atoms sub-TLVs.

A given BGP Wide Community may have parameters that are used as inputs for executing actions defined for that community. These parameters, and any constraints implied by the parameters, MUST be defined by the wide community definition. Parameters consist of an ordered set of Atom sub-TLVs. The semantics of any specific positional instance of an Atom MUST be defined by the wide community.

Care must be taken when using Atoms with list semantics. If the desired behavior is a single or limited number of instances of that type, this MUST be documented as part of the use case of that BGP Wide Community.

If it is the case that a parameter for a given community is of an unexpected type or length, the BGP Wide Community MUST be ignored.

If it is the case that there are too many or too few parameters for a given community, the BGP Wide Community MUST be ignored.

When no parameters are required by the definition of a given Wide Community, the Wide Community Parameters TLV SHOULD NOT be encoded in the community. Implementations MUST be prepared to accept a Wide Community Parameter TLV with an empty value field.

5. BGP Community Container Atoms

Some types of BGP Community Containers, for example BGP Wide Communities, will act on and hence need to encode some distinct Atoms of data. Use of Atoms is solely subject to definition of the specific BGP Container type. Atoms are encoded as TLVs, where each TLV has the following format:

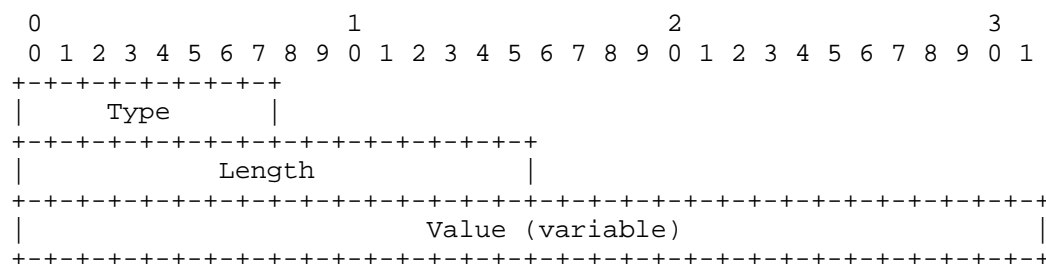


Figure 4: Atoms TLVs

The Type field contains a value of 1-254. The values 0 and 255 are reserved for future use.

The Length represents the length of the "Value" field in octets.

The Value field contains the TLV value.

Supported formats of the TLVs defined in this document are:

- ```
Type 1: Autonomous System Number List.
Type 2: IPv4 Prefix List.
Type 3: IPv6 Prefix List.
Type 4: Integer32 List.
Type 5: IEEE Floating Point Number List.
Type 6: Neighbor Class List.
Type 7: User-defined Class List.
Type 8: UTF-8 String.
```

The semantics of a given Atom will depend upon the context in which it is used, as defined by the containing wide community.

The following sections define the Atoms and their validation rules for contained data within provided Length. If the Length of the Atom does not match the rules for that Atom, it SHALL be considered malformed. (See Section 8.)

In general, Atoms of List type have the semantics of sets. The presence of duplicate entries have no additional semantics, and they MAY be removed by a BGP Speaker propagating the Lists.

### 5.1. Atom Type 1, The Autonomous System Number List

This Atom represents a list of four octet Autonomous System numbers. The minimum Length of this Atom is 4 octets. The Length MUST be a multiple of 4.

When used in an Autonomous System Number List Atom, the following values have special meaning::

0x00000000 - to indicate "No Autonomous Systems".  
 0xFFFFFFFF - to indicate "All Autonomous Systems".

## 5.2. Atom Types 2 and 3, The IPv4 and IPv6 Prefix Lists

These Atoms represents a list of IPv4 or IPv6 prefixes. IPv4 and IPv6 Prefix List Atoms values are encoded as specified for the BGP NLRI in Section 4.3 of [RFC4271].

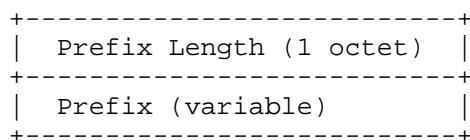


Figure 5: IP prefix in atoms

The Container's Length field must be able to accommodate the list of prefixes according to the encoding rules. If the Container's Length cannot fully accommodate the required number of octets to encode the Atom's Prefix Length and the actual Prefix, the Atom is considered malformed.

## 5.3. Atom Type 4, The Integer32 List

This Atom represents a list of four-octet Integers. These Integers are stored in network byte order.

The minimum Length of the Integer32 list Atom is 4 octets. The Length MUST be a multiple of 4.

## 5.4. Atom Type 5, The IEEE Floating Point Number List

This Atom represents a list of floating point numbers. Floating point numbers are a fixed Length of 4 octets and are stored in [IEEE.754.1985] format.

The minimum Length of the Floating Point Number list Atom is 4 octets. The Length MUST be a multiple of 4.

## 5.5. Atom Type 6, The Neighbor Class List

The Neighbor Class list Atom represents a list of Neighbor classes, each 4 octets in size. Neighbor class currently can contain three values:

**Peer (1):**

This class is typically applied to sessions where a transit-free relationship exists between two providers.

**Customer (2):**

This class is typically applied to sessions where the remote end of the session is operated by a customer.

**Upstream (3):**

This class is typically applied to sessions where the remote end of the session is operated by a network from which you receive transit routes.

The minimum Length of the Neighbor Class list Atom is 4 octets. The Length MUST be a multiple of 4.

**5.6. Atom Type 7, The User-defined Class List**

The User-defined Class list Atom represents a list of user-defined class, each 4 octets in size. The exact property definition is up to the semantics of the defining Autonomous System. The semantics governing a given User-defined Class list are defined by the Context AS Number and the Community Value.

Examples of User-defined Class properties include geography (East, West), continent (North America, Asia, Europe), etc. Similar to the [RFC1997] BGP Communities, it is necessary that the Context AS provide a registry of the value and the semantics of a given community - how this is done is out of scope.

The minimum Length of the User-defined Class list Atom is 4 octets. The Length of this Atom MUST be a multiple of 4.

**5.7. Atom Type 8, the UTF-8 String**

The UTF-8 String Atom represents an arbitrary Unicode string in UTF-8 [RFC3629] format. The Length is required to be of sufficient size to carry the UTF-8 string in the Value field.

Implementations MUST be prepared for truncated/improperly formed UTF-8 strings. When detecting such a string, the implementation should remove trailing octets of a multi-octet sequence in order to have a well-formed string.

Implementations MUST be prepared to receive empty (zero-Length) UTF-8 String Atoms as they may be used as Parameters.

## 6. Well Known Standard BGP Communities

According to RFC 1997, as well as IANA's BGP Well-Known BGP Communities (<https://www.iana.org/assignments/bgp-well-known-communities/bgp-well-known-communities.xhtml>) registry, number of BGP communities are defined to have global significance.

A number of Well-Known BGP Communities are documented by IANA BGP-WKC (<https://www.iana.org/assignments/bgp-well-known-communities/bgp-well-known-communities.xhtml>). It is recommended to continue using the BGP Community Attribute [RFC1997] for the propagation of these Well-Known Communities.

## 7. Operational Considerations

Multiple mechanisms exist for an operator to propagate actions into the network. Besides BGP Wide Communities, an operator can use the BGP Community Attribute [RFC1997], the BGP Extended Communities Attribute [RFC4360], or the BGP Large Communities Attribute [RFC8092] to achieve similar objectives. Care should be taken when using more than one of these tools. The interaction between the different community attributes is out of the scope of this document.

## 8. Error Handling

### 8.1. General Error Handling for BGP Community Containers

The "treat as withdraw" behavior [RFC7606] MUST be executed for any malformed Community Containers, including their contents or presence of the BGP Community Container Attribute or given community TLV more than once in the BGP Update Message.

Each Community Container type may have additional validation rules, including permitted length of Atoms. Failure to conform to those additional rules MUST also be treated as a malformed Community Container.

### 8.2. BGP Wide Community Container Error Handling

If any Atom in a BGP Wide Community container's Exclude Targets TLV is unrecognized, that Wide Community MUST NOT be considered a match and no actions for that community should be processed. While the Targets TLV is meant to be inclusive, the Exclude Targets TLV is meant to be proscriptive of applying the action.

## 9. Example

### 9.1. Example Type 1 Wide Community Definition

An operator of an AS 64496, wishes to locally define a Wide Community with the semantics of permitting AS\_PATH prepending with targets that include AS numbers of peer ASes and peers who have been marked with a set of enumerated city locations. AS 64496 has selected Community Value 1 to represent this functionality.

AS 64496 has established a set of values to use for its User-defined Class:

- 100 - Amsterdam
- 101 - New York
- 102 - San Francisco
- 103 - Tokyo
- 104 - Moscow

Target semantics:

The Autonomous System Number list Atom refers to the target peer AS Numbers.

The User-defined Class for AS 64496 has been defined elsewhere and the values 100..104 may be used for this locally defined Wide Community.

The Targets TLV must contain at least one entry.

The Exclude Targets TLV may contain entries of the above supported Atoms.

The semantics of all other Atoms are undefined for this community.

Parameter semantics:

The parameter TLV shall consist of exactly one Integer32 Atom value that is constrained to have a value of 2..8.

### 9.2. Example Type 1 BGP Wide Community Encoding

AS\_PATH prepend 4 TIMES TO AS 2424, AS 8888, to peers marked as Amsterdam (100) or to peers marked Moscow (104), but not to peers in New York (101).

The T Flag (transitive) is unset to prevent propagation of this community.

```

 0 1 2 3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type (1, Wide) | Flags |0|1| Reserved(0) |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Length: 63 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Community: LOCAL PREPEND ACTION CATEGORY 1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Source AS 64496 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Context AS 64496 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Target TLV (1) | Length: 22 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ASN List (1) | Length: 8 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Target ASN# 2424 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Target ASN# 8888 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| User List(7) | Length: 8 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Amsterdam (100) |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Moscow (104) |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ExcTargetTLV(2) | Length: 7 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| User List(7) | Length: 4 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| New York (101) |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Param TLV (3) | Length: 7 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Integer32 (4) | Length: 4 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Prepend # 4 |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 6: Example 1

## 10. Security considerations

### 10.1. BGP Community Container Security Considerations

The security considerations for BGP Communities [RFC1997] or BGP Extended Communities [RFC4360] apply to BGP Community Containers.

Transitive BGP Community Container communities could unintentionally spread far from their origin. If a router receives many routes from multiple sources on the Internet with different communities, it could cause significant processing burden or memory usage. To prevent this, it is RECOMMENDED that routers should be configured to strip unexpected communities from received routes.

This document uses UTF-8 encoding for the UTF-8 String Atom. There are a number of security issues with Unicode. Implementers and operators are advised to review Unicode Technical Report #36 (<https://unicode.org/reports/tr36/>) to learn about these issues. UTF-8 "Shortest Form" encoding is REQUIRED to guard against the technical issues outlined in [UTR36] (<https://unicode.org/reports/tr36/>).

### 10.2. BGP Wide Community Security Considerations

For BGP Wide Communities, the Community Value and the Source AS Number may provide sufficient context to remove unwanted or unexpected communities.

Given the flexibility and power offered by BGP Wide communities, it is important to consider the additional possibilities allowed by their definition. In particular, for locally defined BGP Wide Communities, it is RECOMMENDED to restrict the range of parameters. For registered BGP Wide Communities, the security considerations of the document defining them MUST address issues specific to those newly defined Communities.

## 11. IANA Considerations

All Registration Policies contained in this document are based on [RFC8126].

### 11.1. BGP Community Container Attribute

This document defines a new BGP Path Attribute called BGP Community Container Attribute. For this new type IANA is requested to allocate a new value in the corresponding registry:

Registry Name: BGP Path Attributes

This document makes the following assignments for the optional, transitive BGP Community Container Attribute:

| Name                              | Type  | Value           |
|-----------------------------------|-------|-----------------|
| ----                              | ----- |                 |
| BGP Community Container Attribute | TBD   | [this document] |

### 11.2. BGP Community Container Atoms Types

This document requests IANA to define and maintain a new registry named: "BGP Community Container Atom Types". The pool of 0x00-0xFF has been defined for its allocations.

Registration procedures:

0x00: Reserved.  
 0x01-0x08: Defined in this document.  
 0x09-0xFE: IETF Consensus.  
 0xFF: Reserved.

This document makes the following assignments for the BGP Community Container Atom Type values registry:

| Name                            | Type  | Value           |
|---------------------------------|-------|-----------------|
| ----                            | ----- |                 |
| Autonomous System Number List   | 0x01  | [this document] |
| IPv4 Prefix list                | 0x02  | [this document] |
| IPv6 Prefix list                | 0x03  | [this document] |
| Integer32 list                  | 0x04  | [this document] |
| IEEE Floating Point Number list | 0x05  | [this document] |
| Neighbor Class list             | 0x06  | [this document] |
| User-defined Class list         | 0x07  | [this document] |
| UTF-8 string                    | 0x08  | [this document] |

### 11.3. BGP Community Container Neighbor Class List Atom Types

This document requests IANA to define and maintain a new registry named: "BGP Community Container Neighbor Class List Atom Types". The pool of 0x00000000-0xFFFFFFFF has been defined for its allocations.

Registration procedures:

0x00000000 : Reserved.  
 0x00000001-0x00000003 : Defined in this document.  
 0x00000004-0xFFFFFFFF : IETF Consensus.  
 0xFFFFFFFF : Reserved.

This document makes the following assignments for the BGP Community Container Neighbor Class List Atom Types registry:

| Name     | Type | Value           |
|----------|------|-----------------|
| ----     |      | -----           |
| Peer     | 1    | [this document] |
| Customer | 2    | [this document] |
| Upstream | 3    | [this document] |

#### 11.4. BGP Community Container Values

This document requests IANA to define and maintain a new registry named: "BGP Community Container Types".

The pool of: 0x0000..0xFFFF has been defined for its allocations.

Registration procedures:

```

0x0000 : Reserved.
0x0001-0x00FF : IETF Review.
0x0100-0xFF00 : First Come First Served.
0xFF01-0xFFFE : Experimental Use.
0xFFFF : Reserved.

```

Values:

```

0x0001 : BGP Wide Community [this document]

```

#### 11.5. Registered Type 1 BGP Wide Communities Container Values

This document requests IANA to define and maintain a new registry named: "Registered Type 1 BGP Wide Community Container Values". The pool of 0x00000000..0xFFFFFFFF has been defined for its allocation.

Registration procedures:

```

0x00000000 : Reserved.
0x00000001-0x7FFFFFFF : Private use.
0x80000000 : Reserved.
0x80000001-0xFFFFFFFF : First Come First Served
0xFFFFFFFF00-0xFFFFFFFFFE : Experimental.
0xFFFFFFFFFF : Reserved.

```

#### 11.6. Registered Type 1 BGP Wide Community TLVs

This document requests IANA to define and maintain a new registry named: "Registered Type 1 BGP Wide Community Optional Sub-Types". The pool of 0x00..0xFF has been defined for its allocation.

Registration procedures:

0 : Reserved.  
1..3 : Defined in this document.  
4..254 : IETF Consensus.  
255 : Reserved.

This document makes the following assignments for the Registered Type  
1 BGP Wide Community Optional Sub-Types registry:

| Name            | Type  | Value           |
|-----------------|-------|-----------------|
| ----            | ----- |                 |
| Targets         | 1     | [this document] |
| Exclude Targets | 2     | [this document] |
| Parameters      | 3     | [this document] |

## 12. Change History

### 12.1. Working Group draft

Changes from -03 to -04:

Many editorial changes.

Restored the structure of the common header to accommodate prior implementations from Huawei. However, do not keep the Hop count per prior IDR and author discussion.

Adopt the name BGP Community Container for the general feature and common header after discussion on IDR regarding Large BGP Communities. Wide communities now specifically refer to the Type 1 container.

Updated the Common Container Header's definition of Length to only cover the length of the contents, and not the header.

Hide the Type 2 (4:4), Type 3 (Nx4), Type 4 (16+Nx4) containers for now.

Outstanding issues addresses and section removed.

Type 1 container renamed from "Wide community" to "Wide community TLVs".

Rename Integer Atom to Integer32.

Example changed, following previous specification change.

Changes from -02 to -03:

Many editorial change.

Introduction of new type of containers: Type 2 (4:4), Type 3 (Nx4), Type 4 (16+Nx4)

Common container header: Type length changed from 2-octets to 1 octet, "Hop Count" removed, "Context AS number" moved from type 1 to the generic header.

Remove community "AS-4 List Generic Wide BGP Community"

Changes from -00 to -02: no change

00: no change

## 12.2. Individual draft

Changes from -03 via -04 to -05:

Update the Introduction.

Substantial re-work of Atom types removing proposed Group container and moving Atoms to be lists.

Added the Exclude Targets TLV to the Wide Community container.

Added a section on error handling.

Updated the example.

Changes from -02 to -03:

Removed C and R named bit fields originally from -00.

Rename Target AS field to Context AS.

Make Integer Atom a fixed 4 octets in length.

Add Neighbor Class Atom

Rename TTL to Hop Count

Changes from -01 to -02:

The Type field has been expanded to 2 octets.

The Length field has been moved to the common header.

Changed format to use TLVs.

Added Atom TLV to define well defined syntactic items.

Added TLVs to distinguish targets from parameters.

Various editorial changes to language.

### 13. Contributors

The following people contributed significantly to the content of the document:

Shintaro Kojima  
OTEMACHI 1st. SQUARE EAST TOWER, 3F  
1-5-1, Otemachi,  
Chiyoda-ku, Tokyo 100-0004  
Japan  
Email: koji@mfeed.ad.jp

Juan Alcaide  
Cisco Systems  
Research Triangle Park, NC  
United States  
Email: jalcaide@cisco.com

Burjiz Pithawala  
Cisco Systems  
170 West Tasman Dr  
San Jose, CA  
United States  
Email: bpithaw@cisco.com

Saku Ytti  
TDC Oy  
Mechelininkatu 1a  
00094 TDC  
Finland  
Email: ytti@tdc.net

## 14. Acknowledgments

This document owes draft-lange-flexible-bgp-communities a debt for the inspiration of many features contained herein.

The authors would like to thank Alvaro Retana, Enke Chen, Pedro Marques, Alton Lo, Igor Gashinsky and Job Snijders for their valuable input.

## 15. References

### 15.1. Normative References

- [IEEE.754.1985]  
Institute of Electrical and Electronics Engineers,  
"Standard for Binary Floating-Point Arithmetic",  
IEEE Standard 754, August 1985.
- [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>>.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO  
10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November  
2003, <<https://www.rfc-editor.org/info/rfc3629>>.
- [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>>.
- [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>>.
- [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>>.

### 15.2. Informative References

- [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>>.
- [RFC3065] Traina, P., McPherson, D., and J. Scudder, "Autonomous System Confederations for BGP", RFC 3065, DOI 10.17487/RFC3065, February 2001, <<https://www.rfc-editor.org/info/rfc3065>>.
- [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>>.
- [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/info/rfc8092>>.

#### Authors' Addresses

Robert Raszuk (editor)  
Arrcus  
2077 Gateway Place  
San Jose, CA 95110  
United States of America  
Email: [robert@raszuk.net](mailto:robert@raszuk.net)

Jeffrey Haas (editor)  
Juniper Networks  
1194 N.Mathilda Ave  
Sunnyvale, CA 94089  
United States of America  
Email: [jhaas@juniper.net](mailto:jhaas@juniper.net)

Andrew Lange (editor)  
Nokia  
777 E. Middlefield Road  
Mountain View, CA 94043  
United States of America  
Email: [andrew.lange@nokia.com](mailto:andrew.lange@nokia.com)

Bruno Decraene (editor)  
Orange  
Email: [bruno.decraene@orange.com](mailto:bruno.decraene@orange.com)

Shane Amante  
Apple  
One Apple Park Way, 3016-DEF  
Cupertino, CA 95014  
United States of America  
Email: amante@apple.com

Paul Jakma  
Huawei Ireland Research Centre  
Georges Court  
Dublin  
D02 R156  
Ireland  
Email: paul@jakma.org