Internet Engineering Task Force	A. Lange
Internet-Draft	Alcatel-Lucent
Intended status: Standards Track	J. Haas
Expires: February 3, 2011	Juniper Networks
	K. Patel
	Cisco
	S. Amante
	Level 3
	August 2, 2010

TOC

Flexible BGP Communities
draft-lange-flexible-bgp-communities-03

Abstract

This document defines a new attribute for BGP called the Flexible Community attribute. Flexible Communities build on the experience and utility of the standard BGP community, and the extended BGP community attributes. This attribute allows operators to associate structured information with a route or set of routes. This information can be then be used to execute routing policy. An enhanced version of communities is necessary to accommodate IPv6, 4-byte ASN's, and introduce a more extensible and flexible policy expression. This document also introduces the concept of Neighbor Classes. A Neighbor Class is applied to a group of BGP neighbors who share certain attributes. For example, the PEER Neighbor Class could be applied to BGP sessions between ASN X and other networks with which ASN X has a non-transit peering relationship.

Status of this Memo

This Internet-Draft will expire on February 3, 2011.

Table of Contents

---

1. Specification of Requirements

TOC

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] (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.).

---

2. Introduction

TOC

This attribute represents the third generation of the BGP community attribute. The first generation is documented in [RFC1997] (Chandrasekeran, R., Traina, P., and T. Li, "BGP Communities Attribute," August 1996.). The second generation, the extended community, is documented in [RFC4360] (Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute," February 2006.).
The Flexible Community Attribute provides a number of important enhancements over the existing BGP Community Attribute and BGP Extended Community Attribute. These enhancements are:

- *Support for IPv6.
- *More efficient encoding of lists of data, for example, a list of ASN's.
- *Clean support for a broad range of future data field structures and interpretations.
- *Support for locally defined community structures, and interpretations.
- *Easy extensibility for a range of future applications.

The continuation and expansion of the structure introduced with Extended Communities allows for policy based on the application where the community is being used. The separation of structure and interpretation allows for easier machine and human parsing of community types which do the same thing with slightly different input.
This attribute continues the use of the Transitivity community bit, first introduced in the Extended Community Attribute.
We also define a set of well-known values for this attribute which can be used to replicate and extend the functionality of the existing well-known community values.
The concept of Neighbor Classes is introduced. A Neighbor Class is defined on a BGP peering session. It allows neighbors that share a set of administrative attributes to be easily grouped together. Policy can

then be defined through Flexible Communities in reference to these groupings.

3. The BGP Flexible Community Attribute

TOC

The Flexible Community Attribute is a transitive optional BGP attribute with a Type Code of TBD. The attribute consists of a list of "flexible communities."
Each Flexible Community is encoded as a variable length quantity. The encoding scheme is:

- *Transitivity Field: 1 bit
- *Structure Field: 7 bits
- *Type Field: 2 octets
- *Length Field: 1 octet
- *Value Field: 0-255 octets

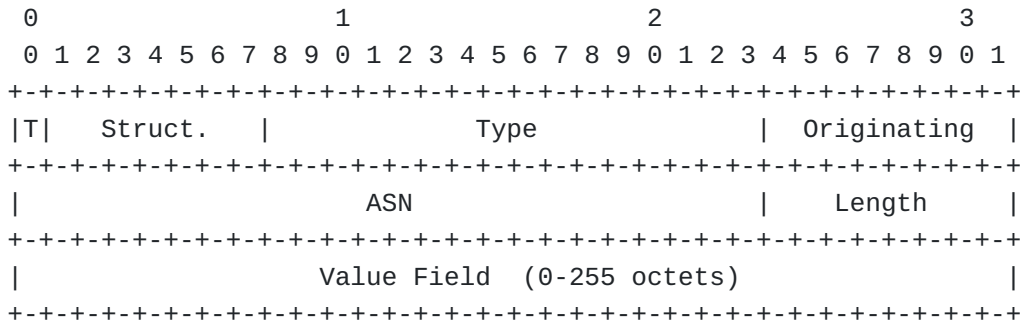

Figure 1: Flexible Community Packet Format

3.1. Transitivity Field

TOC

The Transitivity Field is one bit. It can take the following values:

- *Value 0: The community is transitive across ASes

*Value 1: The community is non-transitive across ASes

It is important to note that the transitivity defined by this field is different from the general transitivity of a BGP attribute. A single Flexible Community Attribute, can contain multiple Flexible Communities, each of which may or may not be transitive. If a route originates in an AS with the transitivity bit set, indicating that the community is non-transitive, then that AS MUST NOT propagate that community to its peers. However, if a community with the transitive bit set is applied on an outbound policy expression (e.g., a route- map), the community will be conveyed to the immediately adjacent peer. That peer, in turn, will NOT propagate the community to its peers. The one exception to this is as-confederations. For the purposes of this attribute, confederation boundaries should be treated the same as IBGP. In other words, non-transitive flexible communities should be propagated to other members of the as- confederation, unless overridden by local policy.
If the community is transitive, then the Value Field MUST contain the originating ASN. This ASN is encoded as a 4-octet value, occupying the first 4 octets of the Value Field. Two-octet ASN numbers are padded out to 4 octets. Any additional information in the Value Field comes after this origin ASN data.

3.2. Structure Field

TOC

The Structure Field's contents modify the Type Field. For example, a Flexible Community which specifies SPECIFIC_NO_EXPORT in its Type Field, can be modified by the contents of the Structure Field to let the receiver know if the list of data on which it must act is a list of 2 octet or 4 octet ASNs. A set of commonly used Structure values is defined later in this document.
The Structure field is the latter 7 bits of the first octet. It is split into two sub-fields.

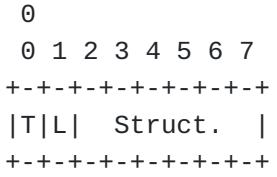

Figure 2: Structure Field

---

*L - Local bit. The Local bit can take two values:

-Value 0: The Structure is Locally Defined.

-Value 1: The Structure is Well Known.

3.3. Type Field

TOC

The Type Field is two octets. This contents of this field are used to define an action for the recipient to take on the route, or to define and attribute that is related to that route. An example of the former would be a Type which requests that a route be ONLY_EXPORTed to a specific set of peers. An example of the latter would be a Type that defines the LINK_BANDWIDTH associated with a certain NLRI.
Like the Structure Field the Type Field is split into two Sub-Fields:

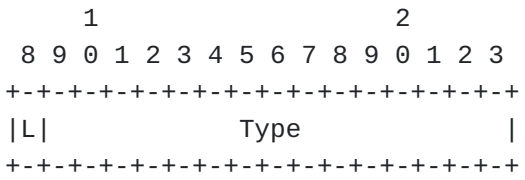

Figure 3: Type Field

The Local bit can take two values:

*Value 0: The Type is Locally Defined.

*Value 1: The Type is Well Known.

An implementation MUST allow an operator to filter out entire Types of Flexible Communities from their peering sessions if they so choose.

3.4. Originating ASN Field

TOC

This field contains the ASN that added the flexible community to the route. It is 4 octets long. In the case of a 2-byte ASN, the first 2 octets are set to zero, as padding.

3.5. Length Field

TOC

The Length Field specifies the length, in octets, of the Value Field.

4. Locally Defined Structures and Types

TOC

The Local bit allows the operator of the network to define Structures and Types that are relevant only within that ASN's boundaries. The definition the term "local" used throughout this document is: "A value used by ad hoc agreement or convention outside the scope of standardization, which has meaning only between the parties using the Flexible Community in question." This typically means that the Local value only has meaning within an AS or set of ASes controlled by a single entity.

A Locally Defined Structure or Type will have a syntax for interpretation defined on the routers that need to interpret it. If a router receives a community with a Locally Defined Structure or Type that it does not recognize, then it should ignore the contents and process the route based on the information in the route that it does understand. This includes obeying the transitivity bit, in the Flexible Community. If the community is set to non-transitive, even if the router does not understand the rest of the Structure or Type of the community, that community should not be forwarded outside the AS.

In order to prevent collisions with other operators' Locally Defined values, Flexible Communities containing Locally Defined Structures or Types MUST be non-transitive (have their Transitivity Field set to 1).

5. Neighbor Classes

TOC

A Neighbor Class is a value which represents a certain class, or group, of BGP neighbors. Each BGP peering session can be configured with zero or more Neighbor Classes. This value will allow a general classification of what sort of relationship the BGP session represents. With the sort of session defined, it becomes easier to apply policy to only that class of neighbors. Neighbor Classes make the expression of policy through flexible communities much easier. There are a number of examples in the sections on defined values.

Neighbor Classes can be used in two main ways. First they can be used in a passive manner, where the configuration acts as a policy expression for communities matching it. An example of this would be

configuring a neighbor with the PEER neighbor class, and having a community that, say is set to NO_EXPORT for the NEIGHBOR CLASS PEER. In this configuration, all the neighbors that are configured as PEERs would match and filter out the route carrying the NO_EXPORT, NEIGHBOR CLASS PEER community. This is the standard way of using neighbor classes.

A second, optional, way to use Neighbor Classes would be to allow inbound community tagging. In this usage, routes traversing the session would be automatically tagged with a flexible community with the appropriate neighbor class value. This eases configuration. To extend the example above, our neighbor designated PEER would add a community with NEIGHBOR CLASS PEER to routes traversing the BGP session. This community could then be used further down the line to just announce PEER routes to a particular customer. This usage is configurable on a per-neighbor basis.

A Neighbor Class is encoded as a 2-octet value with 2 parts:

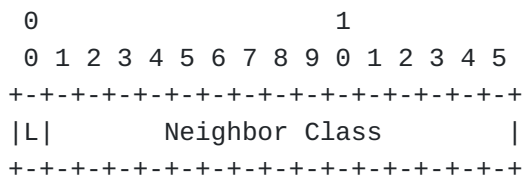

Figure 4: Neighbor Class

The Local bit can take two values:

- *Value 0: The Neighbor Class is Locally Defined.
- *Value 1: The Neighbor Class is Well Known.

5.1. Locally Defined Neighbor Classes

TOC

If a router receives a Flexible Community containing a Neighbor Class that it does not recognize, then it should ignore the contents and process the route based on the information in the route that it does understand. If the Transitivity Field of the Flexible Community with the Locally Defined Structure or Type is set to 1 (the community is non-transitive) then the router MUST NOT forward the Flexible

Community. Similarly, if the Transitivity Field is set to 0 (the community is transitive) the router MUST forward the community along with the NLRI.
Using Locally Defined Neighbor Classes an operator could easily define a set that is locally useful. This set could be used in a flat name space, for example, one could say that "31" would be "Asian Public Peers", and "34" would be European Private Peers.
Also, a given BGP Neighbor can be part of multiple Neighbor Classes. This would allow for a hierarchical or additive name space. For example, a neighbor could be part of both "PEER", and locally defined "ASIAN" and "PUBLIC PEER" Classes. The logical matching functionality available is left implementation-dependent. However, the default in such as case is logical OR functionality for matching this neighbor class. In the case where routes are being tagged inbound, then by default a single community with all configured neighbor classes in a list is added. Implementation dependent knobs are suggested to allow more fine grained control.

5.2. Defined Neighbor Classes

TOC

This document defines Neighbor Class values for common BGP neighbor groupings:

*ALL NEIGHBORS

-This class is the default Neighbor Class for all BGP peers.

-This class is represented by a value of 0 (0x8000).

*PEER

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

-This class is represented by a value of 1 (0x8001).

*CUSTOMER

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

-This class is represented by a value of 2 (0x8002).

*UPSTREAM

-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.

-This class is represented by a value of 3 (0x8003).

*CONFEDERATION PEER

-This class is typically applied to sessions where the remote end of the session is part of a confederation.

-This class is represented by a value of 4 (0x8004).

---

6. Defined Flexible BGP Community Structures

TOC

This section defines a number of Structure values which different Type values can inherit.
Summary of the Defined Values:

*opaque/variable (0x40 or 0xC0)

*list of ASN's (0x41 or 0xC1)

*list of IPv4 addresses (0x42 or 0xC2)

*list of IPv6 addresses (0x43 or OxC3)

*list of neighbor_classes (0x44 or 0xC4)

Defined Structure can be transitive or non-transitive, they are well known.

Opaque/Variable Structure

> This sort of structure defers interpretation of the community and value field to the Type value. Typically this structure value will be used when the Type value does not have a lot of variations, but rather one structure for the Value Field.
>
> This structure is represented by the value 0x00.

List of ASN's

> This structure value means that the Type Field's action is qualified by a list of ASN's, contained in the Value Field. In

the case of 2 byte ASN's the value is padded to 4 bytes by inserting 2 octets worth of zeros in the leftmost portion of the value.

This structure is represented by the value 0x01.

List of IPv4 Addresses

This structure value means that the Type Field's action is qualified by a list of IPv4 addresses, contained in the Value Field.

This structure is represented by the value 0x02.

List of IPv6 Addresses

This structure value means that the Type Field's action is qualified by a list of IPv6 addresses, contained in the Value Field.

This structure is represented by the value 0x03.

List of Neighbor Classes

This structure value means that the Type Field's action is qualified by a list of neighbor classes, contained in the Value Field.

This structure is represented by the value 0x04.

---

7. Base BGP Flexible Community Type (Opaque Type)

TOC

This section defines the base BGP community specification. Since the root value of communities is the ability to tag a route with arbitrary information, and then create a system to give that information meaning, the base flexible community type (type 0), is very simple. This base type could also be described as the "opaque type." All actions can be replicated with a well-thought out, provider dependent, implementation of a scheme using this opaque type.
Like all flexible communities, this type can be transitive or non-transitive. This is a well known type.

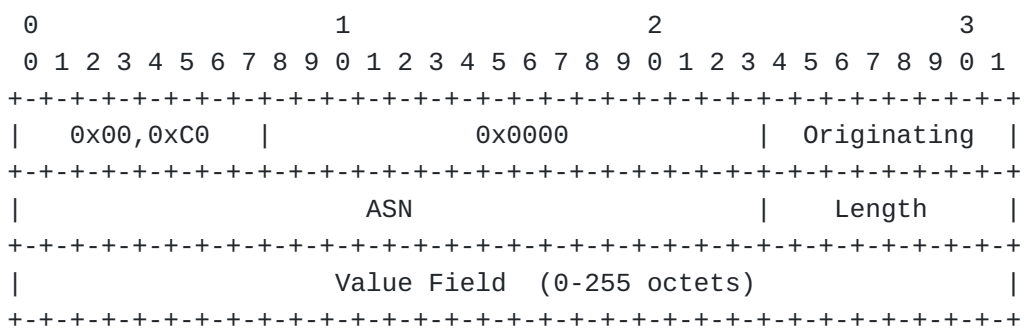

Figure 5: Opaque Type

8. Defined Flexible BGP Community Types

TOC

The Type Field specifies the subgroup that a set of communities belongs to. Typically this subgroup represents an action to be taken on the data. A variety of well-known Type Values follow.

8.1. NO_EXPORT

TOC

This grouping of well-known communities specify a list of ASNs, Peer IPs, or Neighbor Classes NOT to announce a route to.

Name: NO_EXPORT

Type Code: 0x0001

Can Take Structures:

*Ox01 (ASN)

*0x02 (IPv4)

*0x03 (IPv6)

*0x04 (neighbor-class)

Transitive: Non-Transitive

Min Length of Value Field: 2 octets

Max Length of Value Field: 254 octets

Behavior:

> All routes received with this community MUST NOT be advertised to the list of ASNs, Peer IPs, or Neighbor Classes contained in the Value Field.

Notes:

GLOBAL_NO_EXPORT is accomplished by sending a NO_EXPORT Flexible Community with the Neighbor Class of 0x00 (ALL NEIGHBORS).

GLOBAL_NO_EXPORT's NO_EXPORT behavior is defined as: All routes received with this community MUST NOT be advertised outside a BGP confederation boundary (a stand-alone autonomous system that is not part of a confederation should be considered a confederation itself.)[RFC1997] (Chandrasekeran, R., Traina, P., and T. Li, “BGP Communities Attribute,” August 1996.)

This is analogous to the NO_EXPORT community defined in [RFC1997] (Chandrasekeran, R., Traina, P., and T. Li, “BGP Communities Attribute,” August 1996.).

8.2. ONLY_EXPORT

TOC

This grouping of well-known communities specify a list of ASNs, Peer IPs, or Neighbor Classes to announce a route to.

Name: ONLY_EXPORT

Type Code: 0x0002

Can Take Structures:

- *Ox01 (ASN)
- *0x02 (IPv4)
- *0x03 (IPv6)
- *0x04 (neighbor-class)

Transitive: Non-Transitive

Min Length of Value Field: 0 octets

Max Length of Value Field: 254 octets

Behavior:

> The Value Field contains a list of ASNs, neighbor IP addresses, or Neighbor Classes to which the route should be advertised.
>
> The default behavior of a route carrying this community is the same as the GLOBAL_NO_EXPORT behavior, except for the ASNs, IPs, or Neighbor Classes listed in the Value Field.

Notes:

> This community can be used to replicate the NO_ADVERTISE functionality from [RFC1997] (Chandrasekeran, R., Traina, P., and T. Li, “BGP Communities Attribute,” August 1996.). To do so, simply announce ONLY_EXPORT with a Structure of 0x03 or 0x04 (one of the IP address Structures), but with no IP address in the list. This will tell the receiving router that you wish to ONLY_EXPORT this route to NO peer IPs.
>
> This community can also be use to replicate the NO_EXPORT_SUBCONFED functionality from [RFC1997] (Chandrasekeran, R., Traina, P., and T. Li, “BGP Communities Attribute,” August 1996.). To do so, simply announce ONLY_EXPORT with a Neighbor Class of CONFEDERATION PEER (4, 0x8004). This will tell the receiving router that you wish to ONLY_EXPORT this route to Confederation Peers.

8.3. ANNOUNCE_WITH

TOC

This group of well-known communities allows a network to announce a community to an ASN beyond those that it directly peers with, assuming its direct peers allow it to transit the community value. This community group has a lot of flexibility, and could be used to nest another ANNOUNCE_WITH community to gain reach greater than 2 ASN-hops away. If this is a good idea or not is unknown and is left to further study. The only theoretical restriction to the amount of nesting is that the community cannot exceed the maximum size for the Value Field. Since true transitivity can be obtained by simply setting a bit, this community is mainly useful for propagating NO_EXPORT or ONLY_EXPORT (which are non-transitive) to your neighbor's neighbors.
In effect, this community, if allowed by the BGP neighbors in the chain, can be used for an originating network to very specifically control the distribution of its routes. This community type does

contain a LOT of rope, and should be used with care. In the end, though, a mistake should only effect the person originating the route.

> Name: ANNOUNCE_WITH
>
> Type Code: 0x0003

Can Take Structures:

- *Ox01 (ASN)
- *0x02 (IPv4)
- *0x03 (IPv6)
- *0x04 (neighbor-class)

> Transitive: Non-Transitive
>
> Min Length of Value Field: 8 octets
>
> Max Length of Value Field: 254 octets
>
> Behavior: This community's Value Field is split into two sections.
>
> - *The first section is a variable length field that contains the full community value that you wish to announce.
> - *The second section is a variable length field that contains the list of, ASN's, IP addresses, or neighbor_classes you wish to propagate this community to. If you wish to propagate to all peers, use the ALL NEIGHBORS neighbor class.

When a router receives this community value, it should strip the ANNOUNCE_WITH community and announce the underlying community value to its neighbor.

8.4. PREPEND

TOC

This community can be used to ask a BGP peer to prepend its own ASN to its peers.

> Name: PREPEND
>
> Type Code: 0x0004

Can Take Structures:

*0x01 (ASN)

*0x02 (IPv4)

*0x03 (IPv6)

*0x04 (neighbor-class)

Transitive: Non-Transitive

Min Length of Value Field: 3 octets

Max Length of Value Field: 254 octets

Behavior: This community has 2 sections:

*The first section: Is a one-octet value which specifies the number of times that the ASN should prepend its ASN. It is recommended that operators constrain this value to no more than 3. Implementations MUST offer the ability for an operator to set a maximum bound for this field. The suggested default is also 3.

*The second section: Contains a list of ASNs, peer IPs, or Neighbor Classes to which the originator of this community wishes its peer to prepend its ASN.

8.5. The BGP VPN Communities

TOC

These communities are used mostly for BGP MPLS VPN's. Please see [RFC2547] (Rosen, E. and Y. Rekhter, “BGP/MPLS VPNs,” March 1999.) for more detail on how these VPNs are constructed.

Name: ROUTE_TARGET

Type Code: 0x0005

Can Take Structures:

*0x02 (IPv4)

*0x03 (IPv6)

Transitive: Transitive or Non-Transitive

Min Length of Value Field: 4 octets

Max Length of Value Field: 254 octets

Behavior: The Value Field of this community represents a list of the IP addresses where this route is to be announced.

Name: ROUTE_ORIGIN

Type Code: 0x0006

Can Take Structures:

*0x02 (IPv4)

*0x03 (IPv6)

Transitive: Transitive or Non-Transitive

Min Length of Value Field: 4 octets

Max Length of Value Field: 16 octets

Behavior: The Value Field of this community represents a list of the IP address where this route is originated. This community can only contain one IP address.

Name: LINK_BANDWIDTH

Type Code: 0x0007

Can Take Structures:

*0x00 (opaque)

*Ox01 (ASN)

Transitive: Transitive or Non-Transitive

Min Length of Value Field: 4 octets

Max Length of Value Field: 6 octets

Behavior: This community consists of two parts:

*The first part represents the bandwidth of the link in bits-per- second, encoded in IEEE floating point format. This part is 4 octets long.

*The second part consists of a list of ASNs of the peer whose link bandwidth you wish to propagate.

9. New Capability Code for Flexible Communities

TOC

To ensure compatibility between implementations that may or may not implement this protocol extension, this document defines a new capability.

> Capability Code: TBD
>
> Capability Length: 1
>
> Capability Value:
>
> *0x00 for unsupported
>
> *0x01 for supported

Capability negotiation is especially important for this attribute because we are creating a transitivity action within an optional, transitive attribute. If an implementation sends a flexible community with the non-transitive bit set within the community to a router that does not support flexible communities, that router will send the community on to its peers when it should not do so.

10. Aggregation

TOC

Aggregation behaves the same as with other community types.
By default if a range of routes is to be aggregated and the resultant aggregates path attributes do not carry the ATOMIC_AGGREGATE attribute, then the resulting aggregate should have an Flexible Communities path attribute which contains the set union of all the Flexible Communities from all of the aggregated routes. The default behavior could be overridden via local configuration, in which case handling the Flexible Communities attribute in the presence of route aggregation becomes a matter of the local policy of the BGP speaker that performs the aggregation.

11. Operations

TOC

Flexible Communities are handled operationally in a manner very similar to other community values.

A BGP speaker may use the Flexible Communities attribute to control which routing information it accepts or distributes to its peers.
The Flexible Community attribute MUST NOT be used to modify the BGP best path selection algorithm in a way that leads to forwarding loops.
A BGP speaker receiving a route that doesn't have the Flexible Communities attribute MAY append this attribute to the route when propagating it to its peers.
A BGP speaker receiving a route with the Flexible Communities attribute MAY modify this attribute according to the local policy. If a route has a non-transitivity flexible community, then before advertising the route across the Autonomous system boundary the community SHOULD be removed from the route. However, the community SHOULD NOT be removed when advertising the route across the BGP Confederation boundary.
A route may carry the BGP Communities attribute as defined in [RFC1997] (Chandrasekeran, R., Traina, P., and T. Li, “BGP Communities Attribute,” August 1996.), the Extended BGP Communities attribute as defined in [RFC4360] (Sangli, S., Tappan, D., and Y. Rekhter, “BGP Extended Communities Attribute,” February 2006.), and the Flexible Communities attribute. In this case the BGP Communities attribute is handled as specified in [RFC1997] (Chandrasekeran, R., Traina, P., and T. Li, “BGP Communities Attribute,” August 1996.), the Extended BGP Communities attribute is handled as specified in [RFC4360] (Sangli, S., Tappan, D., and Y. Rekhter, “BGP Extended Communities Attribute,” February 2006.) and the Flexible Communities attribute is handled as specified in this document.
If older community types are present in addition to the flexible community there is the possibility that the information contained may be redundant. Although effort should be made to avoid this situation, if it does occur the BGP Speaker shall prefer the flexible community first, then the extended community second and then the base community.

---

12. Security Considerations

TOC

This extension to BGP does not change the underlying security issues.

---

13. IANA Considerations

TOC

The values for the Transitivity Field (1 or 0) are completely defined in this document.
The assignment policy for the Structure Field is:

*The "L" bit's usage is completely defined in this document.

*Values of the Structure Field where the "L" bit is "0" are to be assigned in accordance with the Private Use policy outlined in [RFC2434] (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” October 1998.).

*Values of the Structure Field where the "L" bit is "1" defined in this document are: 0-4 (0x40-0x44 and 0xC1-0xC4). Remaining values in this range are to be assigned using the IETF Consensus policy outlined in [RFC2434] (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” October 1998.).

The assignment policy for the Type Field is:

*The "L" bit's usage is completely defined in this document.

*Values of the Type Field where the "L" bit is "0" are to be assigned in accordance with the Private Use policy outlined in [RFC2434] (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” October 1998.).

*Values of the Type Field where the "L" but is "1" defined in this document are: 0-7 (0x0000-0x0007). Remaining values in this range are to be assigned using the IETF Consensus policy outlined in [RFC2434] (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” October 1998.).

The assignment policy for Neighbor Classes is:

*The "L" bit's usage is completely defined in this document.

*Values of the Type Field where the "L" bit is "0" are to be assigned in accordance with the Private Use policy outlined in [RFC2434] (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” October 1998.).

*Values of the Type Field where the "L" but is "1" defined in this document are: 0-4 (0x8000-0x8004). Remaining values in this range are to be assigned using the IETF Consensus policy outlined in [RFC2434] (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” October 1998.).

TOC

14. Acknowledgements

I would like to thank Tom Barron, Dave Ward and especially Hal Peterson for their valuable comments and feedback. I would also like to thank Brian Haberman.

---

15. Normative References

TOC

[RFC1771]	Rekhter, Y. and T. Li, “A Border Gateway Protocol 4 (BGP-4),” RFC 1771, March 1995 (TXT).
[RFC1997]	Chandrasekeran, R., Traina, P., and T. Li, “BGP Communities Attribute,” RFC 1997, August 1996 (TXT).
[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC2434]	Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” BCP 26, RFC 2434, October 1998 (TXT, HTML, XML).
[RFC2547]	Rosen, E. and Y. Rekhter, “BGP/MPLS VPNs,” RFC 2547, March 1999 (TXT).
[RFC2842]	Chandra, R. and J. Scudder, “Capabilities Advertisement with BGP-4,” RFC 2842, May 2000 (TXT).
[RFC4360]	Sangli, S., Tappan, D., and Y. Rekhter, “BGP Extended Communities Attribute,” RFC 4360, February 2006 (TXT).

Authors' Addresses

TOC

	Andrew Lange
	Alcatel-Lucent
Email:	andrew.lange@alcatel-lucent.com
	Jeffrey Haas
	Juniper Networks
Email:	jhaas@pfrc.org
	Keyur Patel
	Cisco
Email:	keyupate@cisco.com
	Shane Amante
	Level 3
Email:	shane@castlepoint.net