Abstract

The notional premise that different Autonomous Systems belong to different administrative authorities may not always hold. A single administrative authority may instantiate services on and across multiple ASes. A customer accessing those services can reasonably expect that attributes such as LOCAL_PREF that influence routing be applicable even across different ASes. This document describes a mechanism to do so.

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 http://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 July 14, 2014.

Copyright Notice

Copyright (c) 2014 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 (http://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.

This document may contain material from IETF Documents or IETF
Contributions published or made publicly available before November
10, 2008. The person(s) controlling the copyright in some of this
material may not have granted the IETF Trust the right to allow
modifications of such material outside the IETF Standards Process.
Without obtaining an adequate license from the person(s) controlling
the copyright in such materials, this document may not be modified
outside the IETF Standards Process, and derivative works of it may
not be created outside the IETF Standards Process, except to format
it for publication as an RFC or to translate it into languages other
than English.

Table of Contents

1. Introduction .................................................... 2
  1.1. Requirements Language .................................... 3
  1.2. Terminology ................................................. 3
2. Motivation ...................................................... 3
  2.1. One Administrative Domain ................................. 3
3. ATTR_SET_STACK attribute ...................................... 6
4. Example Scenarios ................................................. 8
  4.1. Single provider scenario ................................. 8
  4.2. Dual provider scenario .................................... 11
5. Configuration Management ........................................ 12
6. Operational Considerations ..................................... 13
7. Acknowledgments ................................................ 13
8. IANA Considerations ............................................ 13
9. Security Considerations ........................................ 13
10. Normative References .......................................... 14
Authors’ Addresses ................................................ 14

1. Introduction

One of the basic assumptions of Internet deployment is that different
Autonomous Systems (ASes) belong to different administrative
authorities that use independent policies. Therefore, attributes
such as LOCAL_PREF are not sent across AS boundary. As networks have
evolved, such an assumption may not always hold. A single
administrative authority such as a Service Provider (SP) may own
equipments in multiple ASes and may instantiate services on and
across multiple ASes. As a result, an SP customer’s end-points may
be connected to multiple ASes even though the customer expects the SP
network to behave as a single "domain". For instance, a customer utilizing LOCAL_PREF to influence routing expects that the expressed routing preference be preserved at all of their endpoints whether or not they are connected to same or different ASes. This expectation is reasonable since the ASes, being under the same administrative authority, use consistent policies and LOCAL_PREF set in one AS would be comparable in another AS (when designed to be so). To facilitate such control, this document proposes an approach where non-transitive attributes are tunneled across ASes and are interpreted at traffic ingress points.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2. Terminology

One Administrative Domain (OAD):

A collection of autonomous systems (ASes) that are managed by a single administrative entity. They do not appear any different to ASes that belong to a separate administration.

2. Motivation

2.1. One Administrative Domain

Today a large SP network often consists of multiple ASes, for instance, reflecting the SP's internal management structure. The SP

Figure 1: Typical OAD network
provides services across those ASes to its customers. Some of the sites of a given customer may be connected to one AS whereas some of the other sites of the same customer may be connected to another AS. However, for these customers, the SP network is a single entity. In many instances, the customer desires the routing behavior between two of its sites be uniform whether or not these sites are in the same AS or in different ASes.

Figure 1 provides a typical example of a VPN customer. A customer site with equipment, CE1, is dual-homed to the provider in AS1. A second site of the customer with CE2 is also connected to AS1. A third site of the same customer with CE3 is connected to AS2. CE1 advertises a route. The customer sets different LOCAL_PREF for its two links to the provider network and thereby chooses one of the links as the primary path. CE2 receives the LOCAL_PREFs and correctly uses the preferred link for forwarding. However, CE3 doesn’t receive the LOCAL_PREFs since LOCAL_PREF is not sent across ASes. So CE3 might start to load balance the traffic to CE1 over both links, or might use the non-preferred link solely.

In this scenario, the two ASes are contiguous and under the same administrative domain. So it is desirable that the SP customer be able to use the simple mechanism of setting LOCAL_PREF to influence routing decisions irrespective of the internal design of the provider network. In other words, it is desirable to make the OAD behave essentially as one AS.

The SP may be able to solve the issue by mapping LOCAL_PREF to a community in AS1, allowing the community to go across the AS boundary and finally reverse mapping the community to LOCAL_PREF in AS2. However, an approach like that is narrow in scope and is difficult to manage in a large network.
Multiple ASes under the same administrative authority may not always be contiguous. Figure 2 shows a scenario where two ASes, AS1 and AS2, that belong to the same provider, are separated by an AS that is owned by a third party. Such a scenario may arise due to merger of two SPs. While the mechanism proposed in this draft would work in the same way, caution must be exercised in exposing internal parameters of the provider network to a 3rd party transit AS.

We acknowledge that one can consider fixing the problem described here by merging the ASes into one AS (i.e., by renumbering them to one ASN). However, in many cases that is not a viable option. Instead, the solution described here allows an OAD consisting of multiple ASes to essentially behave as a single AS.
3. ATTR_SET_STACK attribute

```
+------------------------------+  |
| Attr Flags (O|T) Code = TBD   |  |
|------------------------------+  |
| Length                       |  |
+------------------------------+  |
| Attr Flags (O|T) Code = 128    |  |
|------------------------------+  |
| Length (for the outer attrs) |  |
+------------------------------+  |
| Attr Flags (O|T) Code = 128    |  |
|------------------------------+  |
| Length (for inner attributes)|  |
+------------------------------+  |
| Origin AS (provider network) |  |
|------------------------------+  |
| Path Attributes (variable)  |  |
+------------------------------+  |
| Attr Flags (O|T) Code = 128    |  |
|------------------------------+  |
| Length (for inner attributes)|  |
+------------------------------+  |
| Origin AS (customer network)|  |
|------------------------------+  |
| Path Attributes (variable)  |  |
+------------------------------+  |
| Attr Flags (O|T) Code = 128    |  |
|------------------------------+  |
| Length (for inner attributes)|  |
+------------------------------+  |
| Origin AS (customer network)|  |
|------------------------------+  |
| Path Attributes (variable)  |  |
+------------------------------+  |
| Attr Flags (O|T) Code = 128    |  |
|------------------------------+  |
| Length (for inner attributes)|  |
+------------------------------+  |
| Origin AS (customer network)|  |
|------------------------------+  |
| Path Attributes (variable)  |  |
+------------------------------+  |
```

Figure 3: ATTR_SET_STACK

The problem described in Section 2 arises because non-transitive attributes that crucially influence routing decisions are dropped at AS boundaries. The key idea is to carry these non-transitive attributes to the traffic ingress point. BGP already supports attribute tunneling by using the ATTR_SET attribute that transparently carries multiple attributes that need to be preserved across AS boundaries ([RFC6368]). However, ATTR_SET can carry only one set of attributes. As shown in the examples later on, a solution for the present problem needs to carry two sets of attributes, (i) the attribute set for the edge (PE to CE connection, to address the
Towards this end, this document generalizes the attribute tunneling mechanism by introducing a new attribute called ATTR_SET_STACK that carries multiple ATTR_SETs by stacking them. This approach allows adding multiple ATTR_SETs as well as preserves the sequence in which they must be used. The attribute is defined as shown in Figure 3.

The 'Length' field of ATTR_SET_STACK includes the cumulative length, in octet, of all the ATTR_SET attributes.

In this document we define the rules for stacking two ATTR_SET attributes, which are sufficient for the purpose of OAD. We keep the rules open to future additions to support applications that may require more than two ATTR_SET attributes.

Rules:

- When an AS border router (ASBR) advertises a route that doesn’t have an ATTR_SET_STACK attribute to another AS, if allowed by the policy, the ASBR
  * Creates an ATTR_SET_STACK attribute,
  * "Pushes" any existing ATTR_SET attribute in the ATTR_SET_STACK attribute.
  * Encodes the current attributes in an ATTR_SET and "pushes" this ATTR_SET in the ATTR_SET_STACK attribute.

Thus, when there are edge attributes to tunnel, the ASBR creates an ATTR_SET_STACK attribute with two ATTR_SET attributes in it with the ATTR_SET for the edge attributes at the bottom. When only core attributes are to be tunneled, it creates an ATTR_SET_STACK attribute with one ATTR_SET attribute in it carrying the core (set by PE) attributes.

- An ingress PE that imports the route "pops" the top ATTR_SET attributes from the ATTR_SET_STACK. If permitted by the local policy, it uses the attributes from it in its best path selection process.
When an ingress PE advertises an imported route to a CE, only the bottom ATTR_SET element is advertised to it (without any ATTR_SET_STACK attribute wrapper).

If a router receives a route with an ATTR_SET_STACK attribute, and it propagates that route to one of its peers, then if the peer is trusted, the peer receives the route with the same ATTR_SET_STACK attribute; otherwise the ATTR_SET_STACK is removed from the route.

Note that the creation of ATTR_SET_STACK is controlled by local policy (discussed later) and SHOULD be done only for trusted peer ASes.

4. Example Scenarios

In this section, we provide some examples of customer accessing VPN service from a provider to illustrate the difference between the existing behavior and the OAD behavior.

4.1. Single provider scenario

This is a simpler case of a customer connected to only one provider network and there is no edge attribute set.

---

Provider OAD
-------------

AS1 ---- AS2

(RD1)A/B, Lbl1

+------ PE1

+-------- PE2

(RD2)A/B, Lbl2

Figure 4: Option C Network (existing behavior)
As shown in Figure 4, the provider network consisting of two ASes connected by option C technique ([RFC4364]). The customer site with CE1 is dual-homed and advertises prefix A/B to PE1 and PE2. Customer prefers the PE1-CE1 link. This preference is expressed by setting LOCAL_PREF to 200 on the route advertised by PE1 whereas PE2 sets LOCAL_PREF to 150. The second customer site with CE2 is connected to PE3 in AS2. Each PE uses a unique RD. So PE3 receives two prefixes: (RD1)A/B and (RD2)A/B, and imports them into (RD3)A/B. Therefore, the prefix (RD3)A/B has two paths. The first path is with nexthop PE1 (in option C, the nexthops remain unchanged), and the second path is with nexthop PE2.

Existing behavior:
When RR1 sends the routes to RR2, since they are in different ASes, RR1 does not send LOCAL_PREFs to RR2. So when RR2 sends the routes to PE3, it sends default LOCAL_PREF (shown as 100). I.e., PE3 loses the route preferences that were set in AS1.

OAD behavior:
When OAD behavior is turned on on RR1 (and RR2 is added as a trusted peer), when RR1 sends the routes to RR2, it creates an ATTR_SET_STACK attribute with one ATTR_SET in it that contains the LOCAL_PREF of the route. When PE3 imports the routes into (RD3)A/B, it extracts the LOCAL_PREFs from the ATTR_SET_STACK (which contains only one ATTR_SET attribute). Therefore, PE3 has both the LOCAL_PREF set by PE1 and PE2 (coming from the ATTR_SET_STACK) and the (default) LOCAL_PREF set by RR2. As
per the policy set on PE2, the LOCAL_PREFs coming from AS1 can be used by PE2 for computing best path and hence honor the routing preferences set by the customer. This behavior is depicted in Figure 5.

**Provider OAD**

```
<table>
<thead>
<tr>
<th>AS1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/B</td>
</tr>
<tr>
<td>(RD1, Lbl1)</td>
</tr>
</tbody>
</table>

+------ PE1
| /     |
| /     |
| / LP=200\ |
| CE1               |
| RR1            |
| RR2 --- PE3 ----- CE2 |
| \       |
| \ LP=150/  |
| \       |
| \ ASBR1 ... ASBR2 |
| +------ PE2 |
| A/B        (RD2, Lbl2) |

Figure 6: Option B Network (existing behavior)
```

Figure 6 shows the same provider network when its two ASes are connected by option B ([RFC4364]). Similar to the option C case, on PE3, the prefix (RD3)A/B has two paths, but both with next-hop ASBR2.

**Figure 7: Option B Network (OAD behavior)**

```
<table>
<thead>
<tr>
<th>AS1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/B</td>
</tr>
<tr>
<td>(RD1, Lbl1)</td>
</tr>
</tbody>
</table>

+------ PE1
| /     |
| /     |
| / LP=200\ |
| CE1               |
| RR1            |
| RR2 --- PE3 ----- CE2 |
| \       |
| \ LP=150/  |
| \       |
| \ ASBR1 ... ASBR2 |
| +------ PE2 |
| A/B        (RD2, Lbl2) |
| \       |
| \ LP=150/  |
| \       |
| \ ASBR1 ... ASBR2 |
| +------ PE2 |
| A/B        (RD2, Lbl2) |

Figure 7: Option B Network (OAD behavior)
```

```
The VPN label of each route is changed by ASBR2, which allows the packet to ultimately reach PE1 or PE2.

Existing behavior:
Similar to option C, ASBR1 does not send LOCAL_PREFs to ASBR2. So PE3 loses the route preferences that were set in AS1.

OAD behavior:
When OAD behavior is turned on on ASBR1 (and ASBR2 is added as a trusted peer), when ASBR1 sends the routes to ASBR2, it creates an ATTR_SET_STACK attribute with one ATTR_SET in it that contains the LOCAL_PREF of the route. This way PE3 receives both the LOCAL_PREF set by PE1 and PE2 (coming from the ATTR_SET_STACK) and the (default) LOCAL_PREF set by ASBR2. Therefore PE2 can honor the routing preferences set by the customer.

4.2. Dual provider scenario

```
    Provider 1  
        AS1     AS2
       +-------+     +-------+
      | A/B   |    | A/B   |
      | LP=100|    | LP=100|
      +-------+    +-------+  
      | PE1(Lbl1) |
      /           /           
     +-----+     +-----+     +-----+  
    | A/B |     | A/B |     | A/B |
    | LP=150|     | LP=150|     | LP=150|
   
     +-----+     +-----+     +-----+  
    | CE1  |     | CE2  |     | ----------+
    \       \     \       \     +--------------------------+
      +-----+     +-----+     +--------------------------+
     / 
    | A/B |     | A/B |
    | LP=90|     |   |
   +-----+     +-----+  
```

Figure 8: OAD Network in Dual Provider Setup

This example considers the scenario when there is both an edge ATTR_SET and a core ATTR_SET. The scenario is shown in Figure 8 where a customer utilizes enterprise VPN service from both Provider 1 and Provider 2. Provider 1 runs an OAD consisting of two ASes, AS1...
and AS2, connected by interAS Option B or Option C techniques. To Provider 1, the customer connects one site at AS1 via CE1 and another site at AS2 via CE2. At AS1, CE1 is dual-homed connecting to PE1 and PE2 as IBGP ([RFC6368]) and prefers PE1.

CE1 originates a route, A/B, that it advertises to CE2 via both Provider 1 and Provider 2. CE1 prefers Provider 1 by setting the LOCAL_PREF attribute to 100 towards Provider 1 and to 90 towards Provider 2. Within Provider 1, since PE1 is preferred by the customer, PE1 advertises A/B to RR1 with LOCAL_PREF 200 (and label Lbl1) and PE2 advertises A/B with LOCAL_PREF 150 (and label Lbl2). RR1 preserves both routes since PE1 and PE2 uses different route-distinguishers for the customer VPN route.

In Provider 1’s OAD, PE3 receives two routes for A/B: the first one with label Lbl1' and a next-hop that takes the packet to PE1, and the second one with label Lbl2' and a next-hop that takes the packet to PE2.

CE2 receives one route each from Provider 1 (at AS2) and Provider 2. By using the mechanism described in [RFC6368], CE2 sees the LOCAL_PREF attributes set by CE1 and chooses Provider 1’s path and sends traffic to PE3.

Existing behavior:
PE3 does not have any visibility into the LOCAL_PREFs that PE1 or PE2 has set (as LOCAL_PREF is non-transitive attribute) and may choose the path with Lbl2’ as its bestpath and send traffic to PE2 violating the intent of the customer to receive traffic via PE1.

OAD behavior:
When OAD is turned on, PE3 receives the ATTR_SET_STACK attribute containing two ATTR_SETS: (i) the top ATTR_SET containing the core attributes (set by PE1 or PE2), (ii) the bottom ATTR_SET containing the edge attributes that comes from the CE. PE3 extracts the top ATTR_SET for its own best path computation and sends the bottom ATTR_SET to CE2. This way PE3 is able to honor the preferences set in AS1.

5. Configuration Management

An implementation MUST allow the operator to identify the neighbors that belong to the same OAD, and/or are trusted.

An implementation MUST allow the operator to specify whether the attributes from the ATTR_SET (within an ATTR_SET_STACK) are to be used for best path computation. Note that attributes MUST not be
mixed; i.e., either only the attributes from an ATTR_SET are used, or no attribute from an ATTR_SET are used.

6. Operational Considerations

When non-transitive attributes such as LOCAL_PREF are tunneled across AS boundary, the values used for these attributes must be consistent across different ASes in an OAD.

When the originator sends an ATTR_SET_STACK attribute to a 3rd party peer AS, even if the peer AS is a transit AS with respect to the provider network, the peer AS may extract the ATTR_SETs and use them for its own calculations (e.g., if the customer also has a site connected to the 3rd party AS). If the routing policies of the 3rd party AS is not consistent with the originator AS, routing inconsistencies may occur. Therefore, ATTR_SET_STACK attribute may be sent to a peer AS only if the peer AS is trusted. In this context, a trusted AS is either in the same OAD, or it is contractually bound to treat the ATTR_SET_STACK attribute as an opaque attribute, or its routing policy is consistent with the originator AS.

A route carrying an ATTR_SET attribute potentially has two sets of non-transitive attributes for possible use: (i) those in the ATTR_SET, and (ii) those carried by the route. The non-transitive attributes are given a "global" scope when those in the ATTR_SET are used. Sometimes, however, a "local" scope may be preferred in some ASes in a given OAD, in which case the non-transitive attributes carried by the route are used. Local policy must govern which set of attributes should be used.

7. Acknowledgments

8. IANA Considerations

IANA shall assign a value from the "BGP Path Attributes" registry, to be called "ATTR_SET_STACK", with this document as the reference.

9. Security Considerations

The proposed mechanism allows non-transitive attributes to be sent across AS boundary. Sending the non-transitive attributes to non-trusted peers can create routing inconsistencies and other vulnerabilities and MUST not be done.

Procedures and protocol extensions defined in this document do not otherwise affect the BGP security model.
10. Normative References


Authors’ Addresses

James Uttaro
AT&T
200 S. Laurel Avenue
Middletown, NJ  07748
USA

Email: uttaro@att.com

Saikat Ray
Cisco Systems
170 W. Tasman Drive
San Jose, CA  95134
USA

Email: sairay@cisco.com

Pradosh Mohapatra
Cumulus Networks
140C S. Whisman Rd
Mountain View, CA  94041
USA

Email: pmohapat@cumulusnetworks.com