8 Byte BGP Communities

Finding a practical way forward

Problem space

- AS4 is being deployed and starts to be visible in AS paths.
- Policy needs to be expressed with AS4 elements.
- No simple mechanism to encode AS4:AS4 constructs.
- Wide communities proposal addresses this specific problem and adds much more functionality too, just at what cost?

Standard communities – RFC1997

32 bit community value

Community value is treated as a 32 bit integer

Interpreting it as two 16 bit fields is a local administrative decision.

Widely implemented and deployed.

Extended communities - RFC4360

Type	Subtype	Global part
Local part		

Global part is a 16 bit AS number.

Type	Subtype	Global part
Global part		Local part

Global part is an IPv4 address.

Type	Subtype	Opaque value
Opaque value		

Opaque value is a formatless 48 bit field.

Widely implemented and deployed.

IPv6 based extcomms - RFC5701

<table>
<tr><td>Type</td><td>Subtype</td><td colspan="2">Global part</td></tr>
<tr><td colspan="4">Global part</td></tr>
<tr><td colspan="4">Global part</td></tr>
<tr><td colspan="4">Global part</td></tr>
<tr><td colspan="2">Global part</td><td colspan="2">Local part</td></tr>
</table>

Global part represents an IPv6 address.
Technically this part could be reused to carry
AS4:AS4

This does not seem to be supported by vendors.

4 octet extcomms - RFC5668

<table>
<tr><td>Type</td><td>Subtype</td><td colspan="2">Global part</td></tr>
<tr><td colspan="2">Global part</td><td colspan="2">Local part</td></tr>
</table>

Global part represents an AS4 number.
Local part is still 16 bits only.

It is the same extended community encoding
just with a different interpretation of the fields.

RT, RO, DI, L2VPN ID and others

There are many extcomms used for signalling different aspects of L2VPN and L3VPN operation. All of them reuse RFC4360 definition of AS or IPv4 based extended communities.

These extcomms are used with address families other than IPv4 or IPv6.

AS4 awareness is less of a problem in this context as there exist mechanisms for rewriting/mapping of community values.

EVPN communities - RFC7432

<table>
<tr><td>Type</td><td>Subtype</td><td>Flags</td><td>Reserved</td></tr>
<tr><td colspan="4">4 bytes of value container</td></tr>
</table>

EVPN defines a new format within RFC4360 extcomm - it adds the flags field and defines a context-specific 4 byte container. Overall extcomm length is still 8 bytes.

Wide communities

- It is a new approach, not an extension.
- Adds new functionality – flexible parameters, propagation scope control, target selection.
- Adds much more implementation complexity.
- AS number is a 32 bit value in wide communities.
- Vendors have no firm plans to support it, at least at the moment. This is due to complexity.
- It also adds operational complexity – existing policy constructs likely will need to be changed to accommodate the flexibility of wide extcomms.

Wide communities

<table>
<tr><td colspan="2">Type</td><td>Flags</td><td>Hop Count</td></tr>
<tr><td colspan="2">Length</td><td colspan="2">Reg/Loc Type = 1</td></tr>
<tr><td colspan="2">Reg/Loc Type = 1</td><td colspan="2">Source AS</td></tr>
<tr><td colspan="2">Source AS</td><td colspan="2">Context AS</td></tr>
<tr><td colspan="2">Context AS</td><td>AType = 4</td><td>ALen = 4</td></tr>
<tr><td colspan="4">AVal</td></tr>
</table>

AS4:AS4 extcomm would be encoded in
ContextAS:AVal fields.

Possible workarounds

- Use existing 32 bit encoding and split AS4 into two extcomms based on asdot notation.
  - Separate 16 bit parts are carried in two separate extcomms.
  - This allows to reuse existing encoding – at a cost of complexity of expressing such policy.
  - Possibility of conflicts – might need a third “indicator” extcomm.

- Define a mapping scheme to address target AS4 number via a temporary AS number.

- Reuse extcomms defined for other address families or vendor specific extcomms in a creative way.

Requirements for 64 bit extcomms

- Limit the scope of both the problem and proposed solution.

- Must be able to encode AS4:AS4 policy constructs natively in one path attribute entity.
- Policy configuration and expression should not be different for AS and AS4.
- Simple and practical approach.

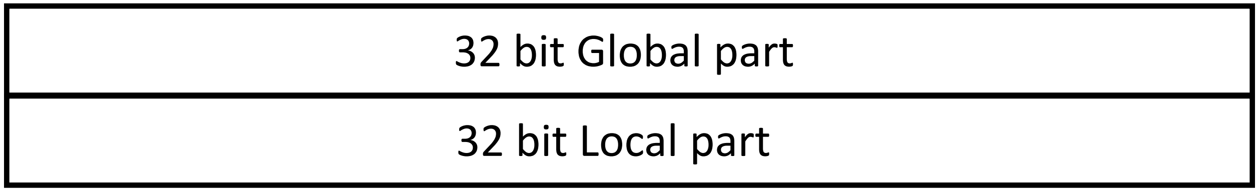
The goal is not to compete with wide communities but to provide a practically deployable solution.

Possible encoding - extendable

<table>
<tr><td>Type</td><td>Subtype</td><td>Flags</td><td>Reserved</td></tr>
<tr><td colspan="4">Global part</td></tr>
<tr><td colspan="4">Local part</td></tr>
<tr><td colspan="4">Reserved, possibly used for scope control</td></tr>
</table>

- New type of extended community path attribute.
- Global and local parts carry AS4 values.
- The third 32 bit field is reserved, could be used for RID or AS number of the target system.
- No use for flags field at the moment, leave as reserved for future extensions.
- May require defining a new capability.

Possible encoding - minimalistic



32 bit Global part

32 bit Local part

- A new path attribute, not an extension to current extcomm encoding.
- Global and local parts carry AS4 values.
- Both values are interpreted as opaque 32 bit integers.
- Operationally it is equivalent to standard communities with two 32 bit fields. Multiple 64 bit communities can be encoded in the same path attribute.
- Applicable to IPv4 and IPv6 address families only.

Discussion

- Is this a problem in your environment?
- How painful is it?
- What other requirements would you see?
- Which encoding option should be used?

- Vendors will need to implement this encoding for it to be practical.