IDR

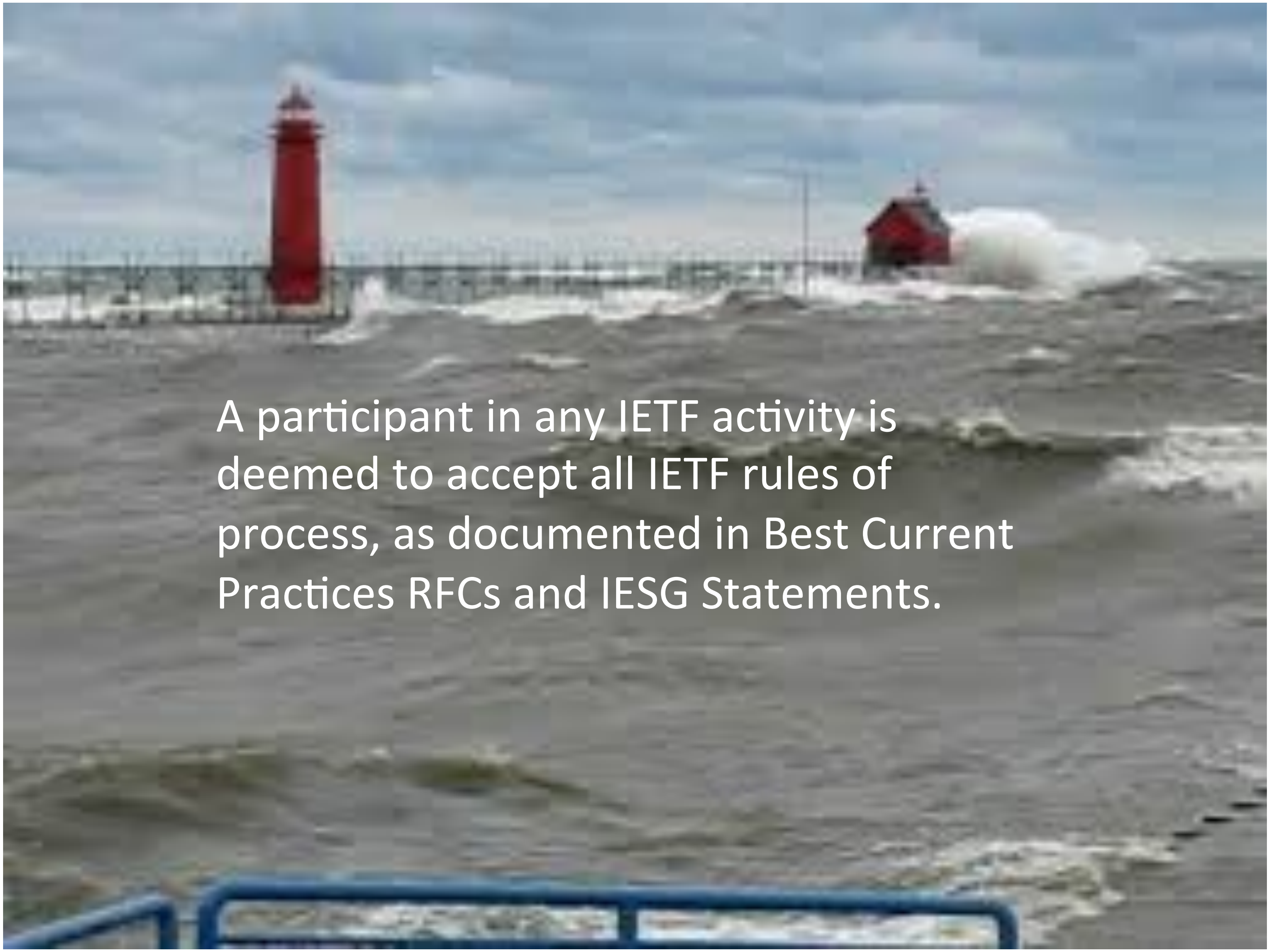
IETF-90, Toronto

July 22, 2014

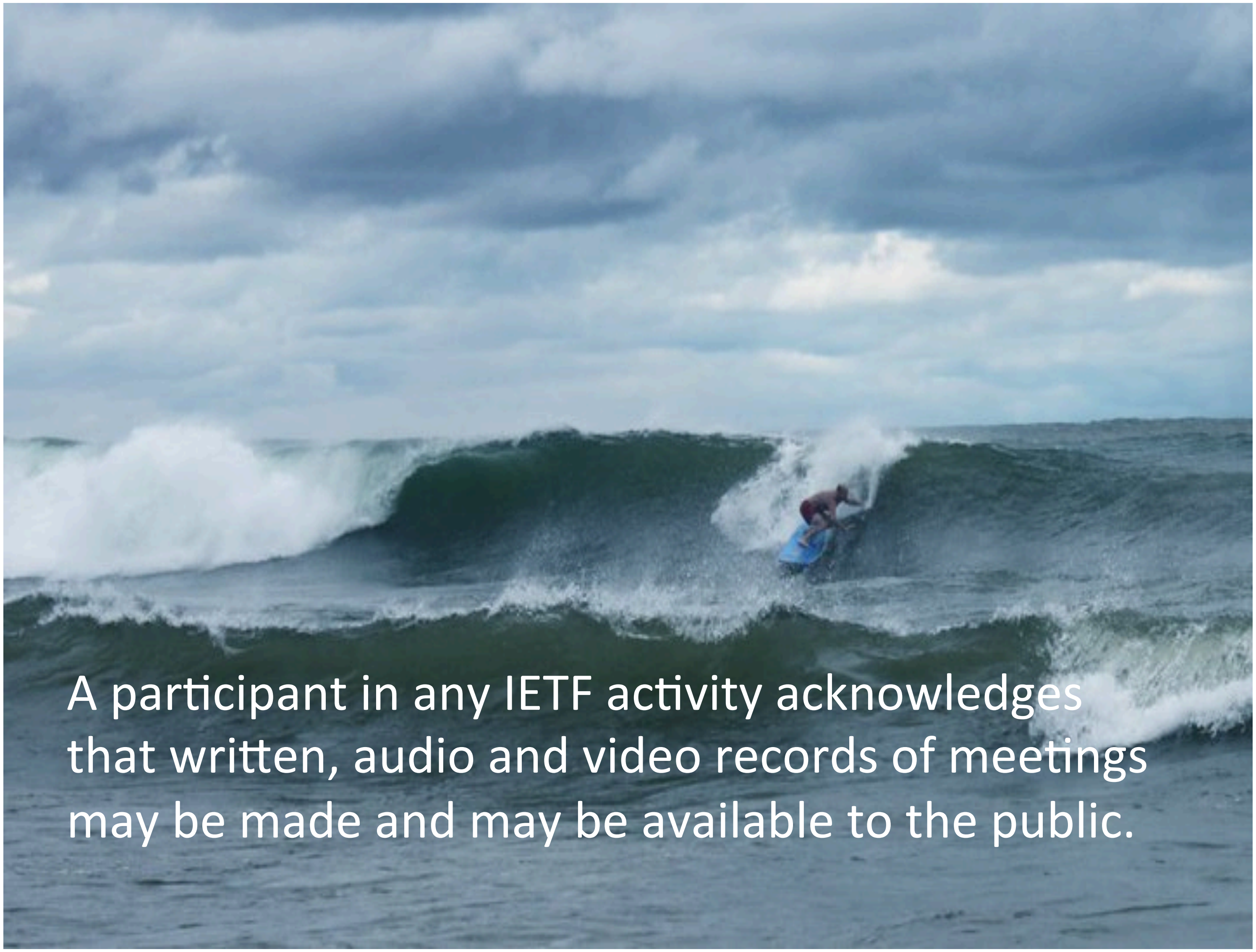
Note Well

- **Note Well**
- Any submission to the IETF intended by the Contributor for publication as all or part of an IETF Internet-Draft or RFC and any statement made within the context of an IETF activity is considered an "IETF Contribution". Such statements include oral statements in IETF sessions, as well as written and electronic communications made at any time or place, which are addressed to:
- The IETF plenary session
- The IESG, or any member thereof on behalf of the IESG
- Any IETF mailing list, including the IETF list itself, any working group or design team list, or any other list functioning under IETF auspices
- Any IETF working group or portion thereof
- Any Birds of a Feather (BOF) session
- The IAB or any member thereof on behalf of the IAB
- The RFC Editor or the Internet-Drafts function

Statements made outside of an IETF session, mailing list or other function, that are clearly not intended to be input to an IETF activity, group or function, are not IETF Contributions in the context of this notice. Please consult RFC 5378 and RFC 3979 for details



A participant in any IETF activity is deemed to accept all IETF rules of process, as documented in Best Current Practices RFCs and IESG Statements.



A participant in any IETF activity acknowledges that written, audio and video records of meetings may be made and may be available to the public.

RFCs Finished

- RFC7153 – IANA registries for BGP extended Communities (draft-ietf-idr-extcomm-iana)
- RFC7196 – Making Route Flap Damping Usable (draft-ietf-idr-rfc-usable)
- RFC7300 – Reservation of Last AS numbers (draft-ietf-idr-last-as-reservation)

At RFC Editors

- Draft-ietf-bgp-route-refresh-10
- Draft-ietf-aigp-18 -

WG Calls coming after IETF

- Draft-ietf-idr-flowspec-redirect-rt-bis (1 week double check for IPR, 13 response)
- Draft-idr-bgp-gr-notification (3 responses, 2 week re-cycle0
- Draft-ietf-idr-error-handling
- Draft-ietf-idr-as-migration-01.txt
- Draft-gredler-idr-ls-distribution

draft-ietf-idr-flowspec-redirect-rt-bis-00

- Flowspec (RFC 5575) specified a "redirect to VRF" extended community action. Basically, it's a route-target.
- That action only permitted 6 bytes of data. The RFC was ambiguous about how to select the VRF type since that's usually part of the 2 bytes that flowspec is using to declare an action.
- That ambiguity lead to interop issues.
- This draft documents the deployed extended community types and puts them in IANA.

Draft authors

- If 2 implementations -→ WG LC and IESG
- If not 2 implement → Can request Be ok by WG and go into Waiting for Implementation state
- If 3 years and no implementations, Sue will be checking in to see if Sue's missed

Looking for Document shepherds

- We are looking for people wanting to be Document Shepherds for IDR

SIDR and BGPSEC

IETF 90

Toronto, CA

Tuesday, 22 July 2014

Sandy Murphy

BGPSEC and SIDR

- The SIDR work has two parts:
  - Origin Validation – basics done and published
    - No changes to BGP protocol
    - Prefix/ASN certificates (RPKI), route origin authorization (ROA)
  - Path Validation – BGPSEC – finishing up
    - Builds on origin validation and RPKI certificates
    - Prevents fiddling with the AS_PATH
      - Prevents adding valid origin to bogus AS_PATH
    - Changes to BGP protocol

IDR Attention Requested

- BGPSEC adds a new attribute to BGP
  - Protecting AS_PATH means dealing with the core of BGP
- This needs a serious look from BGP experts (you)
  - Some IDR experts have participated
  - More eyes would be good
- If you need background
  - Try the architecture (RFC6480), origin operation (RFC7511) and route validation (RFC6483 and RFC6811) and the bgpsec overview draft-ietf-sidr-bgpsec-overview

ERROR-HANDLING -13

John Scudder

July 21, 2014

DRAFT STATUS

WGLC for draft -07 initiated May 7, 2014

Led to much discussion with many productive comments

Six drafts later, we have -13… and it's still not quite ready

Two remaining open issues:

- What's a “valid IP host address”?
- What to do with the Traffic Engineering path attribute?

WHAT’S A “VALID IP HOST ADDRESS”?

RFC 4271 requires that the NEXT_HOP be a “valid IP host address”. (RFC 4760 inherits this.)

- But, it doesn’t say what that means.

Nailing this down has proven surprisingly slippery.

- Clearly unreasonable to define from scratch
- There must be a reference to cite… right?

IANA REGISTRIES

IANA maintains an “IPv4 Special-Purpose Address Registry” and an “IPv6 Special-Purpose Address Registry

- These have various attributes, including whether an address can be a “destination” and whether that address is “forwardable”
- If an address can’t be a destination or isn’t forwardable, that means it’s not a “valid IP host address”… right?

A TWISTY MAZE OF PASSAGES

But what about IPv4-mapped IPv6 addresses? (Used in VPNv6, 6PE)

- OK, make an exception for them.
- And punt on other AFI/SAFI.
- And exceptions should be configurable, of course.

Are we done yet?

- Nope!

… ALL DIFFERENT

Robert Raszuk points out that RFC 5575 (Flow-Spec) makes the next hop optional (sort of)

- So, do we continue to dot i’s, cross t’s, and add codicils to cover every permutation of next hop?

Chris Hall (June 14) does a nice analysis of errors as “venial”, “mortal” and “trivial”

- “all next-hop errors are venial errors. So the draft doesn't, AFAICS, need to get sucked into defining "valid host address"”.

ARE ALL NEXT-HOP ERRORS “VENIAL”?

In MP_REACH, next-hop length defines the beginning of the NLRI section

- So if the next-hop length is invalid, maybe we can’t properly find the NLRI, so can’t do treat-as-withdraw. That’s a “mortal” error (session reset).

This comes back to an underlying assumption of the draft: generally confine ourselves to syntax errors, let purely semantic ones go.

WHEN IN DOUBT, PUNT

Jeff Haas proposes (June 16), radical simplification, rather than trying to dot every i and cross every t:

- "If the next hop field contains a semantically incorrect address within the context of deployed features and address family, treat as withdraw behavior should be used. "

And I would add, if syntactically incorrect in the same context (wrong length), session reset.

PROPOSAL

Adopt something like the language on the preceding slide

Accept that the document will not provide a detailed prescription for every case

- We somehow have muddled through 25 years of BGP anyway

Remove “martians” discussion entirely

TE PATH ATTRIBUTE

The draft tries to cover every extant BGP path attribute (that doesn’t already have compliant spec language and isn’t deprecated)

For most attributes, this was straightforward.

Not so for the TE path attribute (RFC 5543).

When in doubt, punt!

PROPOSAL FOR TE PATH ATTRIBUTE

“an implementation that determines (for whatever reason) that an UPDATE message contains a malformed Traffic Engineering path attribute MUST handle it using the approach of "treat-as-withdraw"”

- Possibly, attribute discard would be OK (if the TE attribute is strictly an optimization)

This language is already in -13

NEXT STEPS

Update -14 with new, reduced “valid IP host address” section

New WGLC

BGP Remote Next-Hop

draft-vandevelde-idr-remote-next-hop

G. Van de Velde, K. Patel, D. Rao, R. Raszuk, R. Bush

90nd IETF - Toronto, USA
22 July 2014

What is BGP Remote Next Hop?

- New generic attribute for encapsulation related signaling for BGP NLRIs
- Each attribute carries one or more tunnel end-points for an NLRI
- Tunnel encapsulation information is included in attribute
- One or more remote-next-hops supported for an NLRI
- Directly attached to NLRI of any address-family

Applicable Use-cases

- Build dynamic overlay infrastructure
- Multi-homing for IPv6 support
- Virtualization and mobility signaling

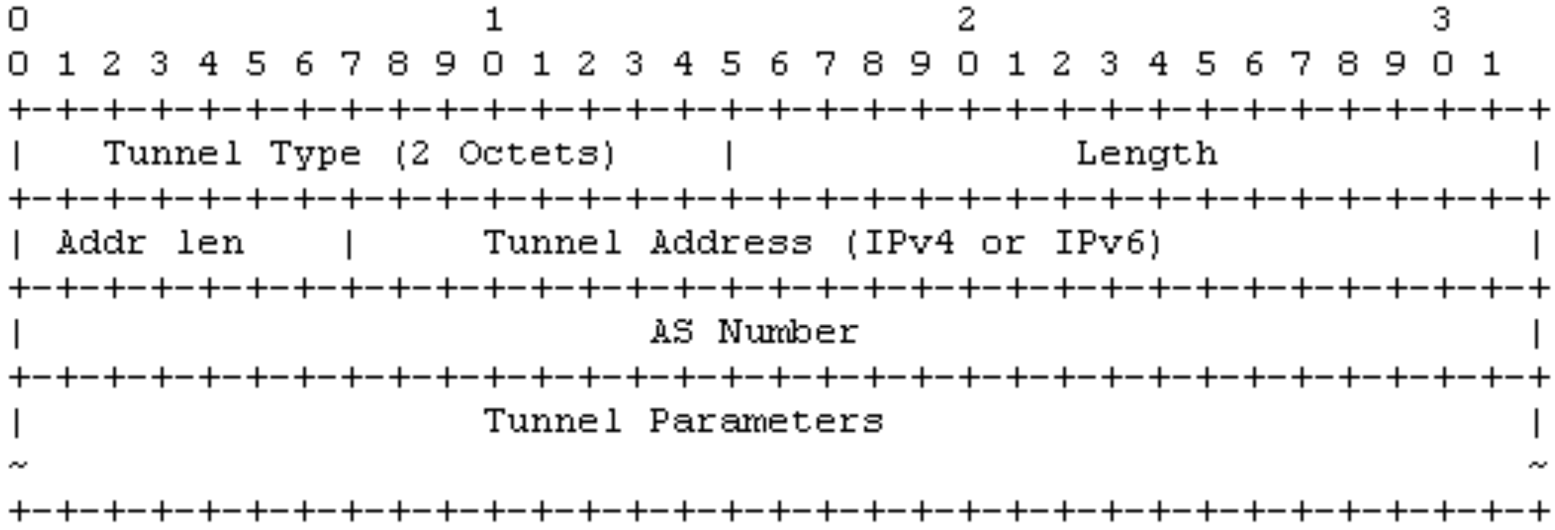
Brief history

- First presented at IETF85, Atlanta, Nov 2012
- Added new use-cases (vEPC, NFV)
- Added new sub-TLVs
- Incorporated additional feedback
- Current version -07

rNH Highlights

- Optional Transitive Attribute
- Composed of a set of TLV encodings
- Supports signaling of multiple tunnel end-points
- Additional data for each end-point
- Re-uses RFC5512 Tunnel Parameters sub-TLV
- Works for iBGP and eBGP
- Graceful global non-flag day insertion supported
  - When rNH not supported, traditional routing will happen
  - When rNH supported, local policy may use associated information
- Optional RPKI validation can be used for security

TLV Format



```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Tunnel Type (2 Octets)    |              Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Addr len    |     Tunnel Address (IPv4 or IPv6)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          AS Number                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Tunnel Parameters                           |
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Tunnel types, sub-TLVs defined for:
  - L2TPv3 over IP, MPLS-in-GRE, IP-in-IP, VxLAN, NVGRE, GTP

Benefits

- Simplifies automatic tunnel signaling for prefixes inside and between AS’s
- Supports per-prefix granularity
- No need to enable new address-family between speakers
- Consistent, extensible method across variety of use-cases
- Applicable across multiple address-families

Next steps

- Add text to reflect recent updates
  - new use-case, comments
- Ready for WG adoption

draft-vandevelde-idr-remote-next-hop

THANK YOU!

NEXTHOP_PATH_RECORD ATTRIBUTE for BGP

draft-zhang-idr-nexthop-path-record-00

Zhenbin Li, Li Zhang, Susan Hares
Huawei Technologies

IETF 90, Toronto, Canada

NEXTHOP_PATH_RECORD ATTRIBUTE

- NEXTHOP_PATH_RECORD ATTRIBUTE

```
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attr. Flags  |Attr. Type Code|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Attr.Flags should be **optional transitive**
Attr.Type Code should be allocated by IANA

- Next hop path segment

```
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        |   Length                      |  Reserved     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        Next Hop                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Next Hop is the route next hop address

BGP NEXTHOP_PATH_RECORD ATTRIBUTE Description

- NEXTHOP_PATH_RECORD ATTRIBUTE
  - Optional transitive BGP Path Attribute
  - Records Sequence of next hop path segments
- Operation: Path Record Config on and next-hop-self
  - If originate, add next_hop to the next_hop_segment
  - If passing append next_hop to next_hop_segment
- Operation: Path Record Config on and no next-hop-self
  - Don't originate
  - Pass but do not modify

Deployment Considerations

- Customized Best Path Selection
  - The next_hop_record information gathered on an IBGP or EBP route could be used by off-line decision processing to select paths, and re-inserted as policy to affect the decision making via I2RS

Next Steps

- Get feedback on the NEXTHOP_PATH_RECORD ATTRIBUTE extension and application
- Coordinate with other similar drafts to record next-hop path information through BGP extensions.
- The procedure for next hop path segment usage for IPv6 or other extensions will be discussed later

Questions?

Extensions to RT-Constrain for Hierarchical RR Scenario

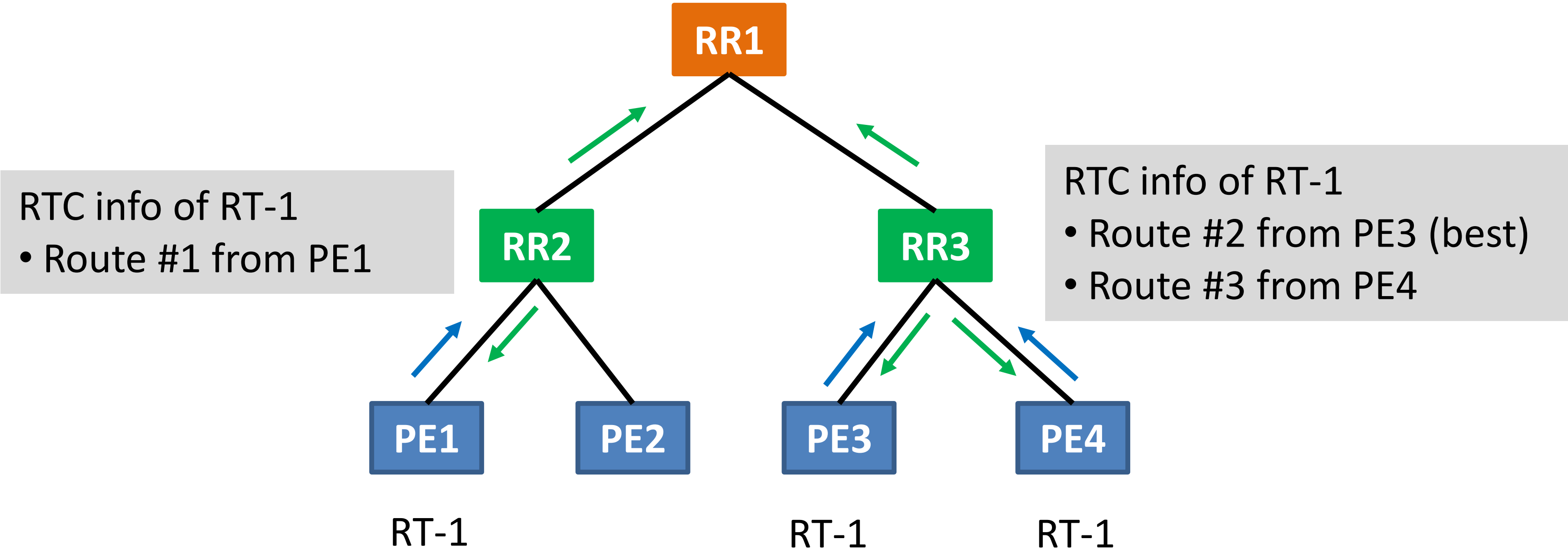
Jie Dong, M. Chen

IETF90 IDR Toronto 2014

Problem Statement

- RFC 4684 specifies rules for RT membership information advertisement
  - To build a VPN route distribution graph
- In hierarchical RR scenario, with current rules the route distribution graph can not be built correctly

Typical Scenario



RTC info of RT-1

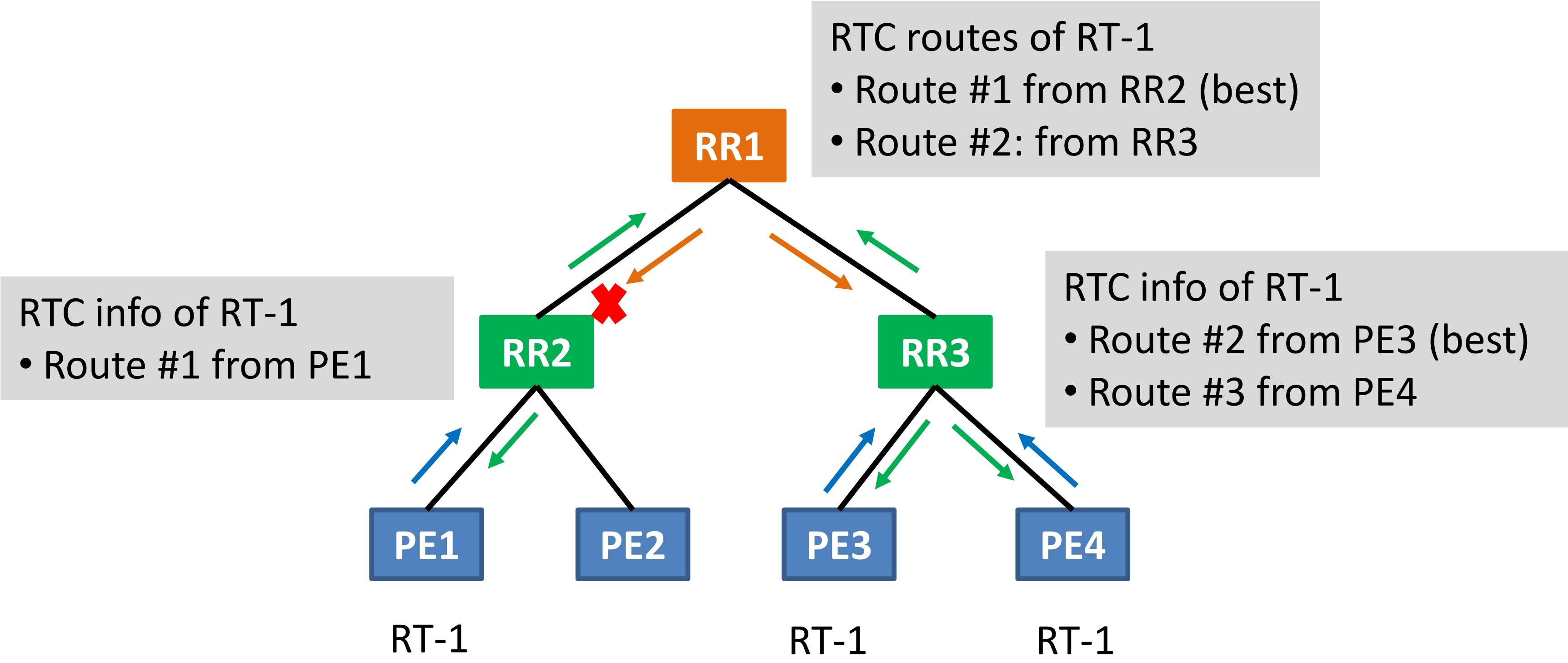
- Route #1 from PE1

RTC info of RT-1

- Route #2 from PE3 (best)
- Route #3 from PE4

- RR2 and RR3 select the best RTC route and advertise it to RR1
  - Create CLUSTER_LIST and insert its own CLUSTER_ID

Typical Scenario



RTC routes of RT-1
- Route #1 from RR2 (best)
- Route #2: from RR3

RTC info of RT-1
- Route #1 from PE1

RTC info of RT-1
- Route #2 from PE3 (best)
- Route #3 from PE4

- RR1 selects the best RTC route (route #1 from RR2) and advertise to RR2 and RR3
- RR2 detects its own CLUSTER_ID in the RTC route, discard it
- RR2 will not advertise VPN routes with RT-1 to RR1

Proposed Solution

- The advertisement rule of RT membership info needs to be modified
  - The objective of RTC is to build a **complete** route distribution graph
  - When advertising an RT membership NLRI to an RR client, if the best path according to RFC 4271 is the path received from this client, and there are alternative paths received from other peers, the **most disjoint alternative path** SHOULD be advertised to this client
  - Most disjoint alternate path:
    - The CLUSTER_LIST and ORIGINATOR_ID attributes are different from those of the best path

Received Comments

- The problem scenario is acknowledged
- The solution space needs more consideration
  - The proposed ‘advertise alternate path’ should be generalized to all iBGP peers, not just to RR clients
  - ‘Add-paths all’ among RRs may be another possible solution
  - Sending default RT from higher RRs to lower RRs
- More discussion about the solution is needed

Next Steps

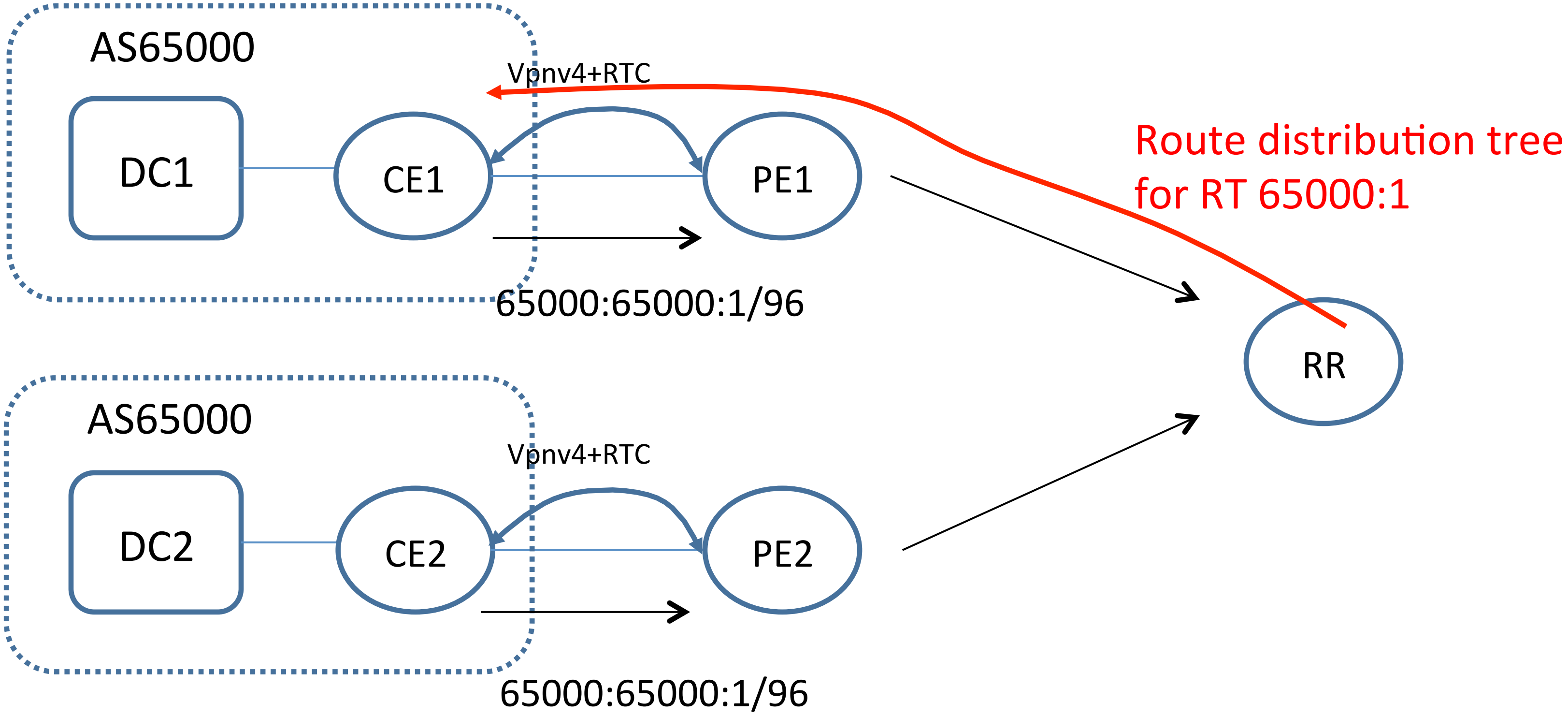
- Continue solution discussion in WG
- Revise the solution section to reflect the consensus
- Then WG adoption?

draft-litkowski-idr-rtc-interas

S. Litkowski, Orange

J. Haas, Juniper

Problem statement



When disjoint ASes setup is used, route distribution tree is wrongly built, preventing communications between sites

Problem statement

- RFC4684 Section 3.2 defines :

  "As indicated above, the inter-AS VPN route distribution graph, for a given route-target, is constructed by creating a directed arc on the inverse direction of received Route Target membership UPDATEs containing an NLRI of the form {origin-as#, route-target}.

  Inside the BGP topology of a given autonomous-system, as far as external RT membership information is concerned (route-targets where the as# is not the local as), it is easy to see that standard BGP route selection and advertisement rules [4] will allow a transit AS to create the necessary flooding state."

- For external RT membership, distribution tree is built over shortest path

Problem statement

- The other rules defined in Section 3.2 of RFC4684 seems to not apply to external informations

> “Route Target membership information that is originated within the autonomous-system, however, requires more careful examination. “

Proposal

- Rules defined in RFC4684 Sec 3.1 & 3.2 are modified
- Path pruning may be disabled by user configuration for :
  - Specific AS numbers (different from local AS)
  - All private ASes

Proposal

```
  ASN 65000                                                   ASN 64000
+-----------+                                              +-------------+
|   ASBR3   | -- (mpebgp vpnv4+rtc) -- ASBR1      PE1 ---- | CE1 --- DC1 |
|     |     |                             \      /         +-------------+
|     |     |                        (mpibgp vpnv4+rtc)
|(vpnv4+rtc)|                               \  /
|     |     |                                RR
|     |     |                               /  \
|     |     |                       (mpibgp vpnv4+rtc)        ASN 64000
|     |     |                            /       \         +-------------+
|   ASBR4   | -- (mpebgp vpnv4+rtc) -- ASBR2      PE2 ---- | CE2 --- DC2 |
+-----------+                                              +-------------+
```

In this situation path pruning may be disabled for AS64000 but enabled for AS65000.

Disabling pruning for all privates Ases, would create unnecessary flooding states in this scenario.

Conclusion & Next steps …

- Basic specification sounds broken for disjoint ASes case (very familiar case in VPN environment)

- WG Feedback on our proposal ?

IDR WG

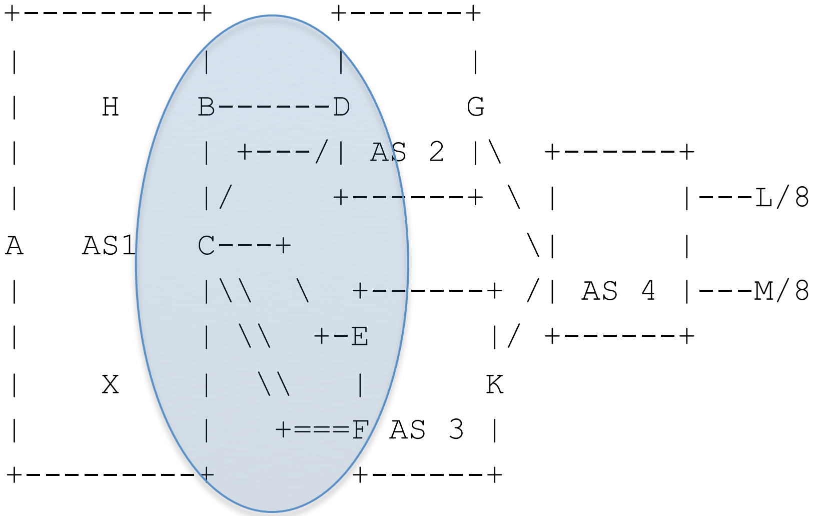
Segment Routing BGPLS Egress Peer Engineering Extensions
draft-previdi-idr-bgpls-segment-routing-epe-00

S. Previdi, C. Filsfils, S. Ray, K. Patel

IETF90 – Toronto, July 2014

Motivations

- Problem statement / use case described in draft-filsfils-spring-segment-routing-central-epe

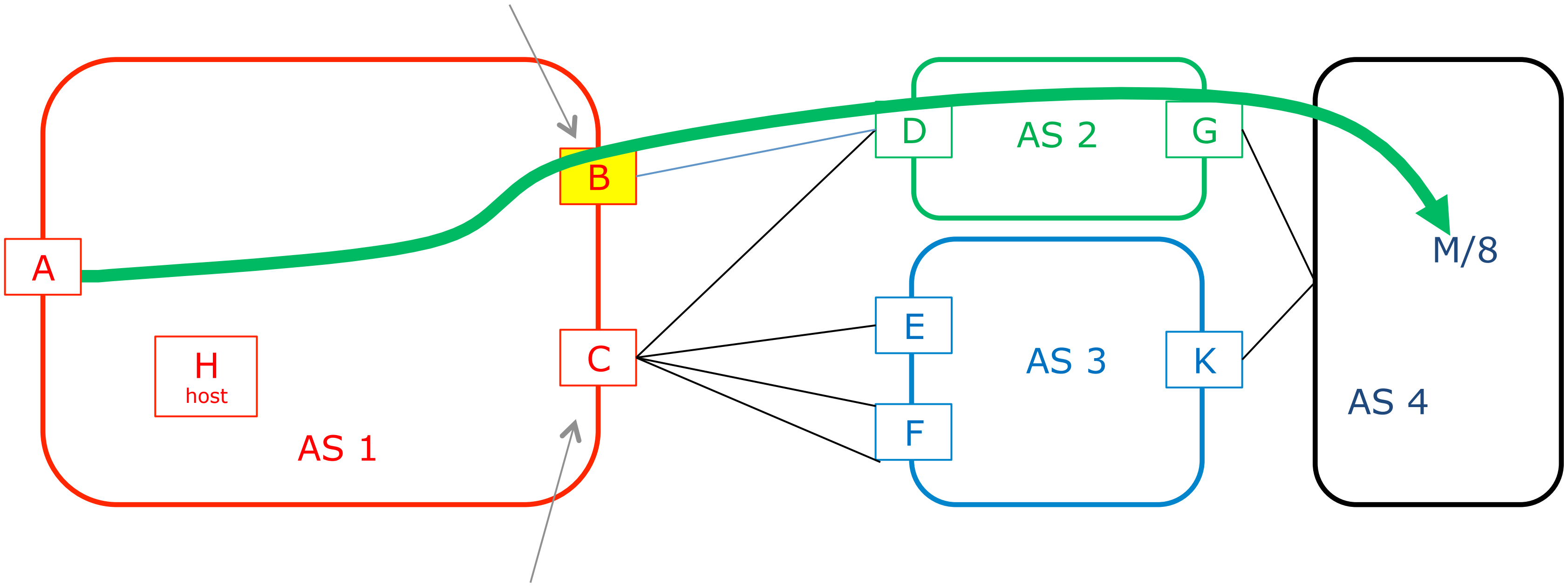


```
+---------+      +------+
|         |      |      |
|    H    B------D      G
|         | +---/| AS 2 |\  +------+
|         |/     +------+ \ |      |---L/8
A   AS1   C---+            \|      |
|         |\\  \  +------+ /| AS 4 |---M/8
|         | \\  +-E      |/ +------+
|    X    |  \\   |      K
|         |   +===F AS 3 |
+---------+       +------+
```

- Section 1.2 Problem Statement

  A **centralized controller** should be able to instruct an ingress PE or a content source within the domain to use a specific egress PE and a specific external interface to reach a particular destination.

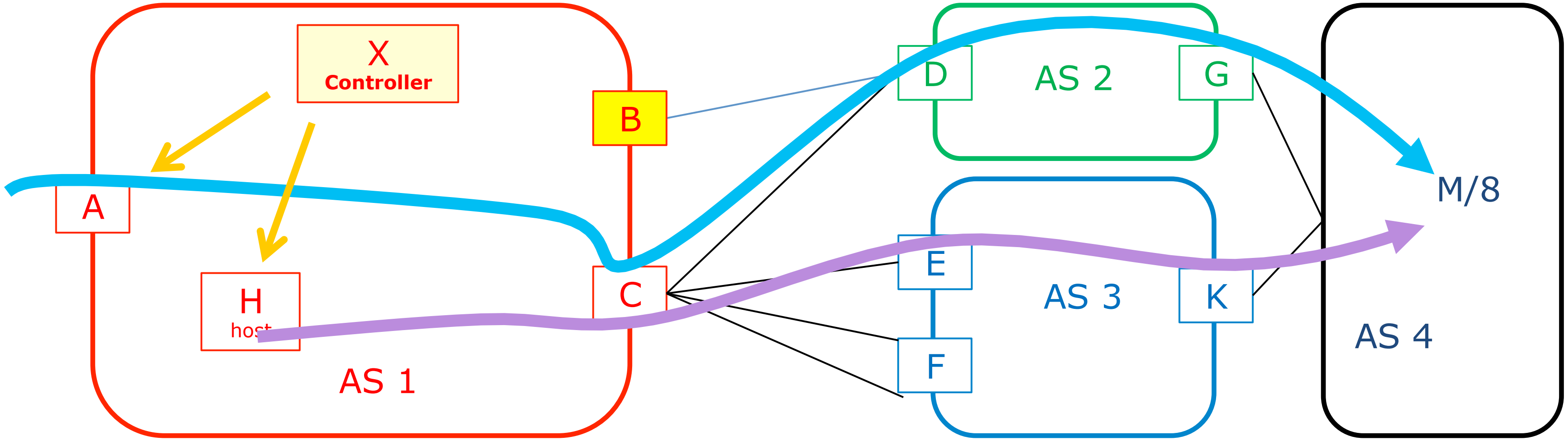
Reference Diagram



AS1's best-path to M/8: B

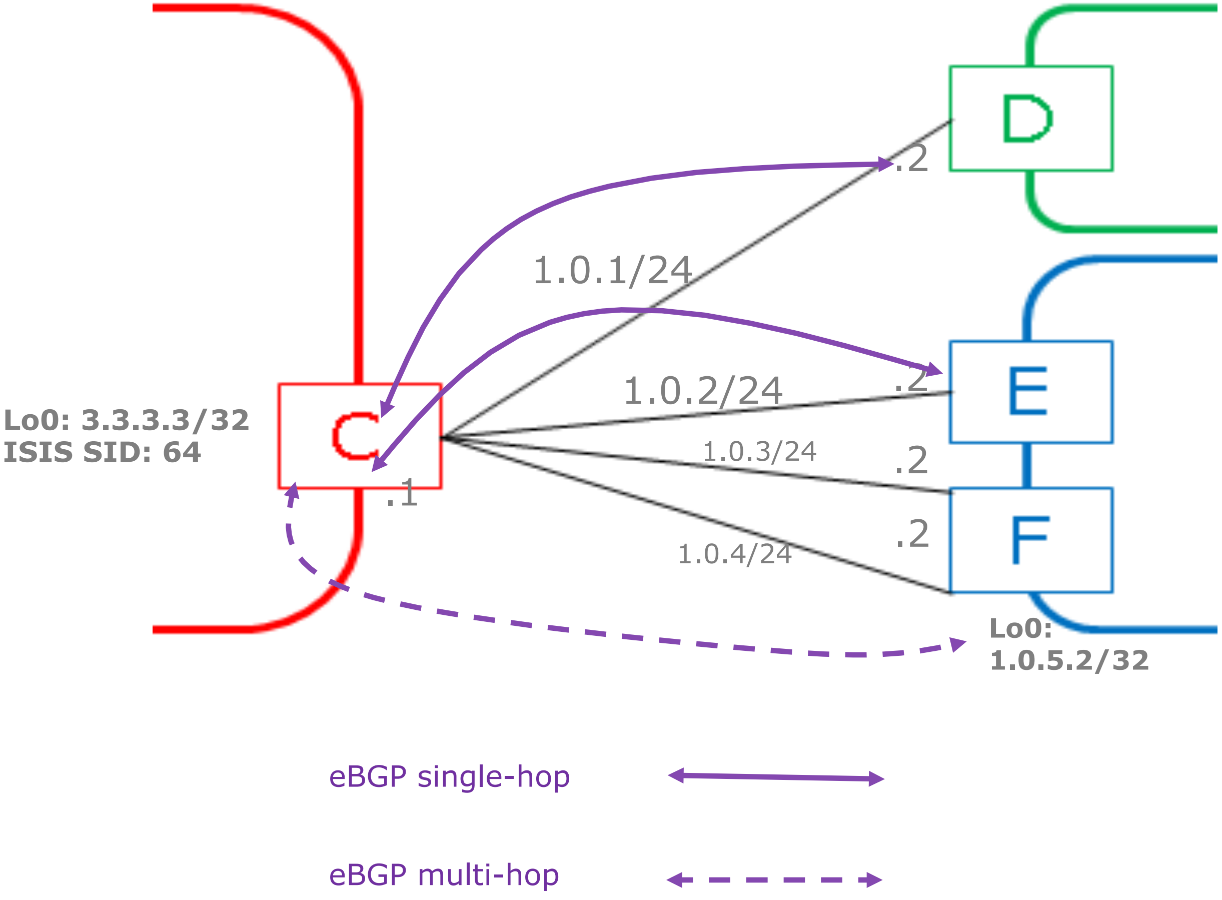
C's Node SID = 64

Objective: centralized egress peer engineering



- Per-Flow TE state only at the source node
  - Ingress router or directly at the source host

eBGP Peering Topology
BGP Peering Segments

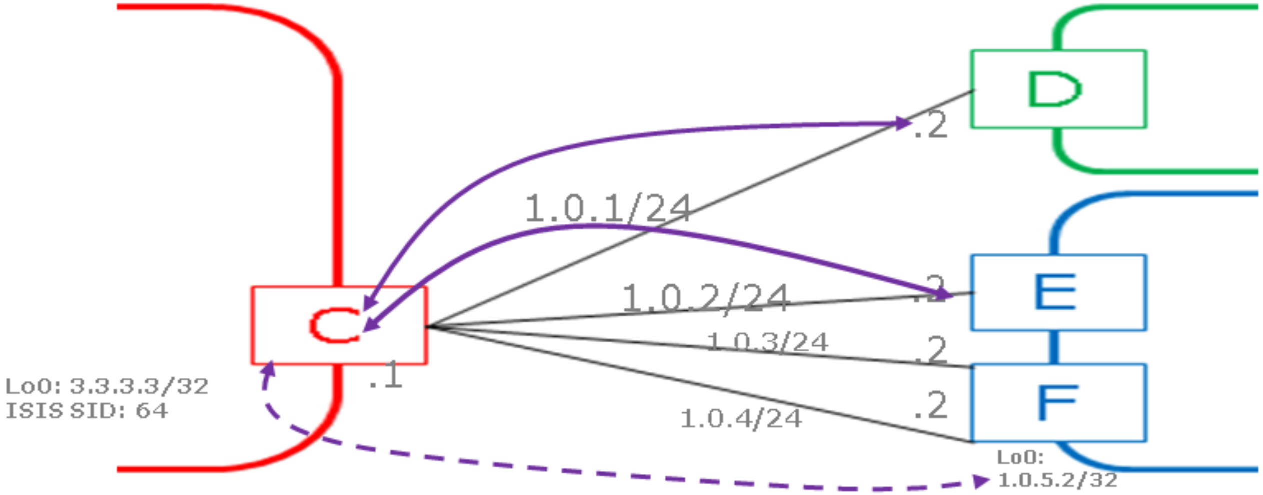


Automated BGP Peering SID allocation

BGP Peering SID's in C's MPLS Dataplane

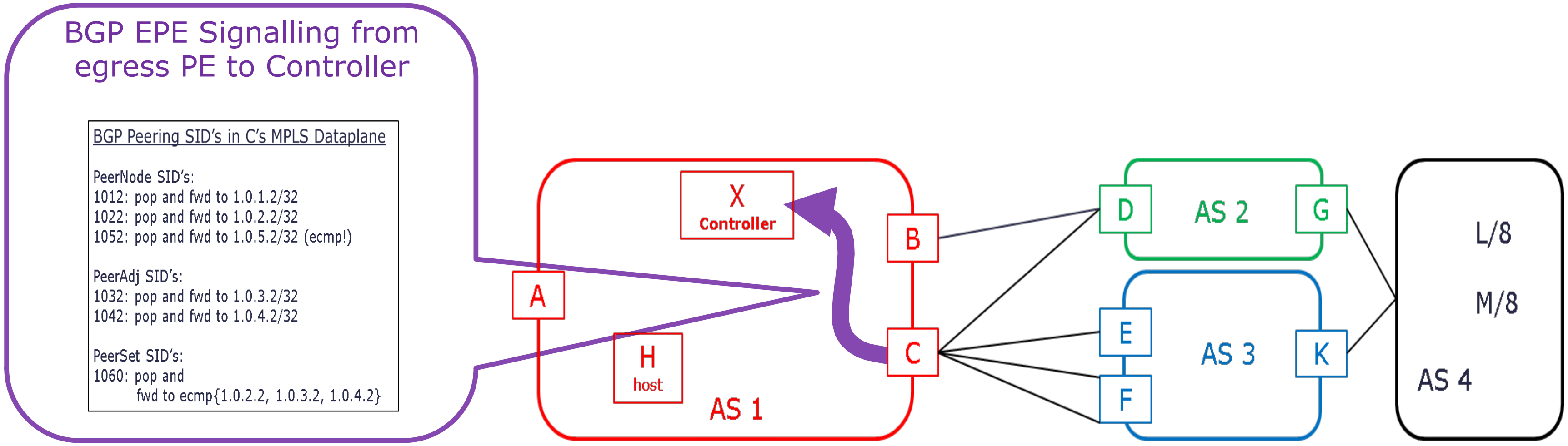
PeerNode SID's:
1012: pop and fwd to 1.0.1.2/32 (D)
1022: pop and fwd to 1.0.2.2/32 (E)
1052: pop and fwd to 1.0.5.2/32 (ecmp to F)

PeerAdj SID's:
1032: pop and fwd to 1.0.3.2/32 (upper link to F)
1042: pop and fwd to 1.0.4.2/32 (lower link to F)



BGP EPE Routes
draft-previdi-idr-bgpls-segment-routing-epe

- The controller learns the BGP Peering SID's and the external topology of the egress border router via BGP-LS EPE routes



BGP EPE Signalling from egress PE to Controller

BGP Peering SID's in C's MPLS Dataplane

PeerNode SID's:
1012: pop and fwd to 1.0.1.2/32
1022: pop and fwd to 1.0.2.2/32
1052: pop and fwd to 1.0.5.2/32 (ecmp!)

PeerAdj SID's:
1032: pop and fwd to 1.0.3.2/32
1042: pop and fwd to 1.0.4.2/32

PeerSet SID's:
1060: pop and
fwd to ecmp{1.0.2.2, 1.0.3.2, 1.0.4.2}

Controller – Decision

- Collects valid internet routes from peers
- Collect performance information across peers
  - EPE solution allows to target probes across probed peer
- Based on business policy and performance information, decides to engineer a flow via an explicit peer different than the best-path
- Outside the scope of the IETF drafts

Peer NLRI Type

- New NLRI Type (TBA, suggested value 5)
  - Peer NLRI-Type
  - Describes the connectivity of a BGP Egress router

```
 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
+-+-+-+-+-+-+-+-+
|  Protocol-ID  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Identifier                          |
|                            (64 bits)                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//               Local Node Descriptors (variable)             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//               Peer Descriptors (variable)                   //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//               Link Descriptors (variable)                   //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Local Node Descriptors: as defined in draft-ietf-idr-ls-distribution Section 3.2.1.2.
- Link Descriptors: as defined in draft-ietf-idr-ls-distribution Section 3.2.2.

Peer Descriptors

- Peer Descriptors (Peer Descriptors Sub-TLVs are defined in draft-ietf-idr-ls-distribution

Sub-TLV	Description	Length
Code Point		
512	Peer Autonomous System	4
513	BGP-LS Identifier	4

Peer Attributes

- The Peer Attributes Sub-TLVs codepoints (defined in draft-idr-ls-distribution):

TLV Code Point	Description	Length	IS-IS SR TLV /sub-TLV
1099	Adjacency Segment Identifier (Adj-SID)	variable	31 (section 2.3.1)
1100	LAN Adjacency Segment Identifier (Adj-LAN SID)	variable	32 (section 2.3.2)
TBA	Peer Set SID	variable	31 (section 2.3.1)

Peering Segments

- See draft-filsfils-spring-segment-routing-central-epe for description of use cases

Questions?

Thanks!

BGP Extensions

for Inter-AS TE Link Distribution

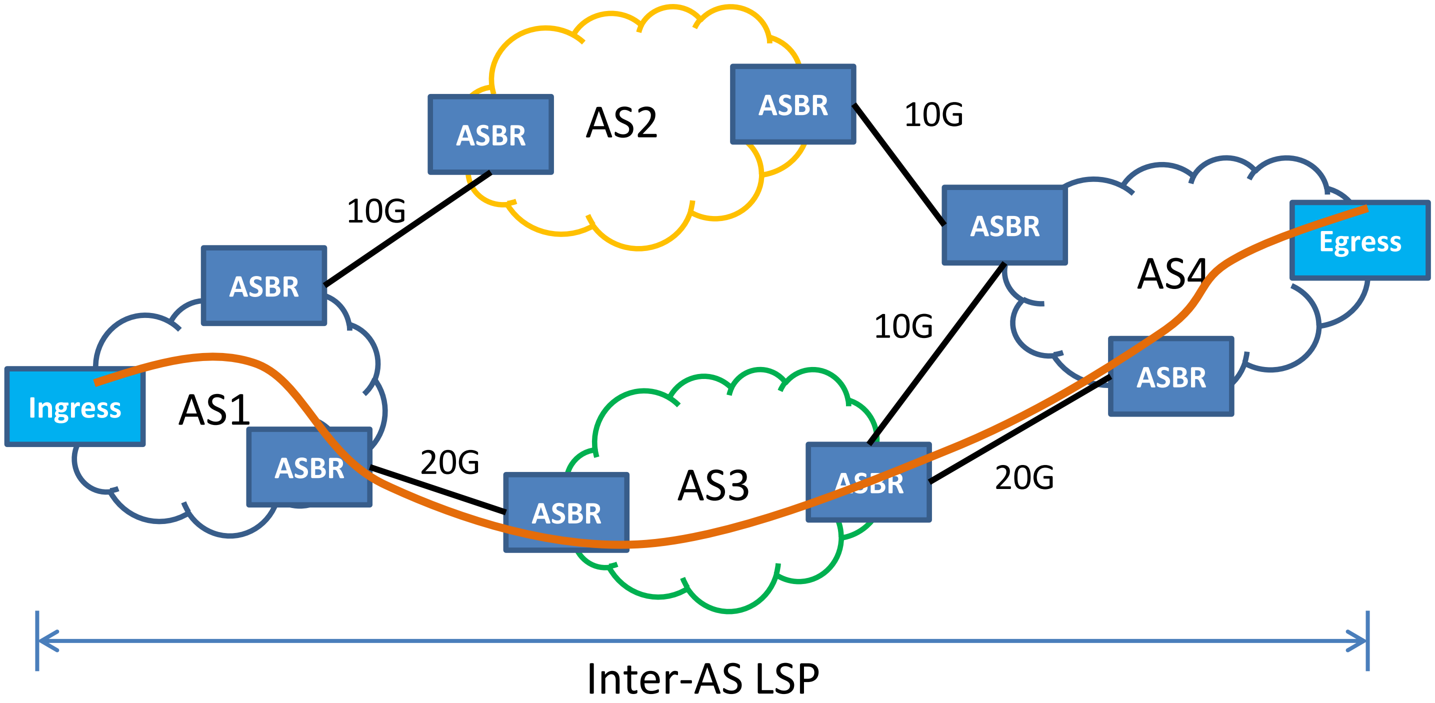
Jie Dong, M. Chen

IETF90 IDR Toronto 2014

Background

- RFC 5316 and 5392 extend IGPs for Inter-AS TE link information flooding
  - Some information are manually configured
    - Remote ASN, remote ASBR TE ID
  - ‘Proxy’ LSA/LSP for two-way link check
    - Can not specify the Inter-AS TE link attributes accurately
    - Additional complexity and processing overhead

Typical Scenario



- Inter-AS TE LSP requires accurate info of Inter-AS TE links for optimal end-to-end path calculation
- Dynamic exchange of Inter-AS TE link info is needed

Proposed Solution

- Dynamic exchange of Inter-AS TE link info between the adjacent ASes
  - Local/remote ASN
  - Local/remote BGP ID
  - Peering addresses
  - TE link attributes
- By default SHOULD NOT be advertised to other ASes
- Can also be used for north-bound distribution of Inter-AS TE link info under policy control

BGP Extensions

- Link-State NLRI in draft-idr-ls-distribution is extended
  - New protocol ID: Inter-AS
  - New Node Descriptor Sub-TLV: BGP Identifier
- Link NLRI with protocol ID ‘Inter-AS’:
  - Contains Local ASBR ID, Remote ASBR ID, peering link ID
- TE attributes of the Inter-AS link are carried in BGP-LS attribute
  - Bandwidth
  - SRLG
  - …

Next Steps

- Collect feedbacks from WG
- Revise the draft

BGP Extensions for Service-Oriented MPLS Path Programming (MPP)

draft-li-idr-mpls-path-programming-00

Zhenbin Li , Shunwan Zhuang (Presenter)
Huawei Technologies

IETF 90, Toronto, Canada

Introduction

- Service-oriented MPLS programming proposed by [I-D.li-spring-mpls-path-programming] is to provide customized service process based on flexible label combinations.

- BGP will play an important role for MPLS path programming to allocate MPLS segment, download programmed MPLS path and the mapping of the service path to the transport path.

- This document defines BGP extensions to support service-oriented MPLS path programming.

Use Cases for Unicast Service MPLS Path Programming

- Use cases for unicast service MPLS path programming is shown as follows:

```
+----------+----------+----------+----------+----------+
|  Entropy | Steering |VPN Prefix|    VPN   |  Source  | ---> Transport
|   Label  |   Label  |   Label  |   Label  |   Label  |        Tunnel
+----------+----------+----------+----------+----------+
```

✓ VPN Prefix Label : Basic reachability. It is defined in [RFC4364].

✓ VPN Label: Identification of VPN. It is defined in [I-D.zhang-l3vpn-label-sharing].

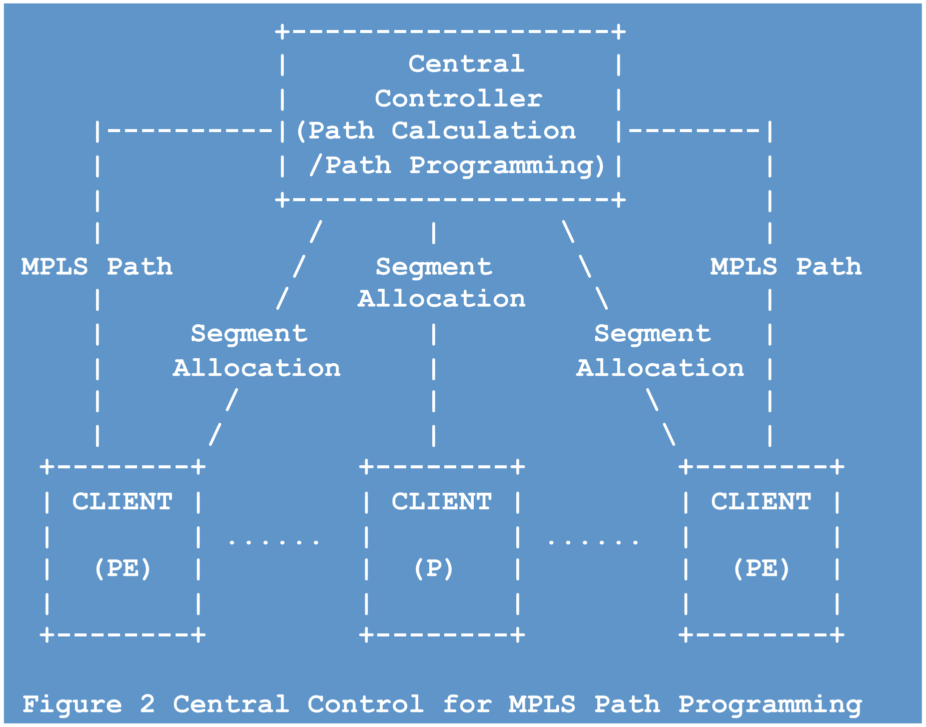
✓ Entropy Label: Identification of ECMP. It is defined in [RFC6790].

✓ Source Label: Identification of source PE which can be used for OAM. It is defined in [I-D.chen-mpls-source-label].

✓ Steering Label: [I-D.filsfils-spring-segment-routing-central-epe] illustrates the application of steering label for the Egress Peer Engineering (EPE).

Architecture of MPLS Path Programming

- Central control plays an important role in MPLS path programming. It can extend the MPLS path programming capability easily. There are two important functionalities for the central control:
  - Central controlled MPLS label allocation: Label can be allocated centrally for special usage other than reachability. These labels can be used to compose MPLS path. We call it as MPLS Segment.
  - Central controlled MPLS path programming: Central controller can calculate path in a global network view and implement the MPLS path programming based on the collected information of MPLS segments to satisfy different requirements of services.



```
                +--------------------+
                |      Central       |
                |     Controller     |
     |----------|(Path Calculation   |--------|
     |          | /Path Programming) |        |
     |          +--------------------+        |
     |            /       |       \           |
MPLS Path        /     Segment     \      MPLS Path
     |          /     Allocation    \         |
     |     Segment        |      Segment      |
     |    Allocation      |     Allocation    |
     |        /           |           \       |
     |       /            |            \      |
  +---------+        +---------+        +---------+
  | CLIENT  |        | CLIENT  |        | CLIENT  |
  |         | ...... |         | ...... |         |
  |  (PE)   |        |   (P)   |        |  (PE)   |
  |         |        |         |        |         |
  +---------+        +---------+        +---------+
```

Figure 2 Central Control for MPLS Path Programming

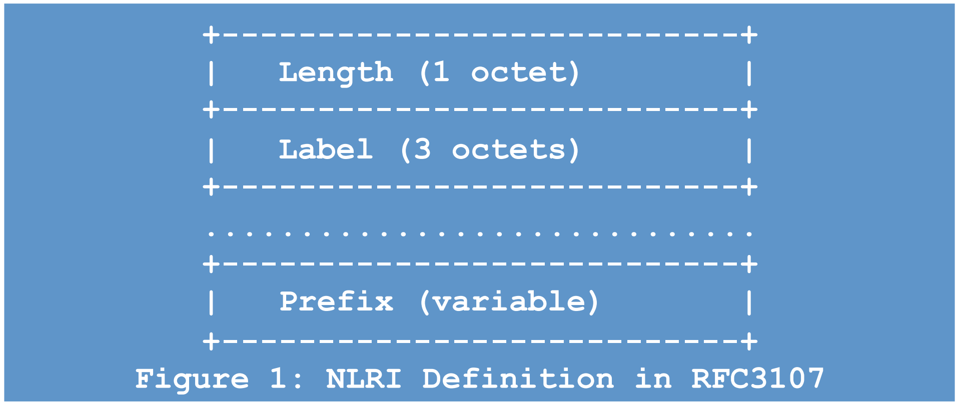
BGP Extensions Requirements for Service-Oriented MPLS Path Programming

- BGP

1. REQ 01: BGP extensions SHOULD be introduced to distribute local label mapping for specific process.
2. REQ 02: BGP extensions SHOULD be introduced to distribute global label mapping for specific process.
3. REQ 03: BGP extensions SHOULD be introduced to download label stack for service-oriented MPLS path.
4. REQ 04: BGP extensions SHOULD be introduced to carry the identifier of the transport MPLS path with service MPLS path to implement the mapping.

Download of MPLS Path

- According to the service requirements, the central controller can combine MPLS segments flexibly. Then it can download the service label combination for specific prefix related with the service. BGP extensions are necessary to advertise label stacks for prefix in NLRI field.



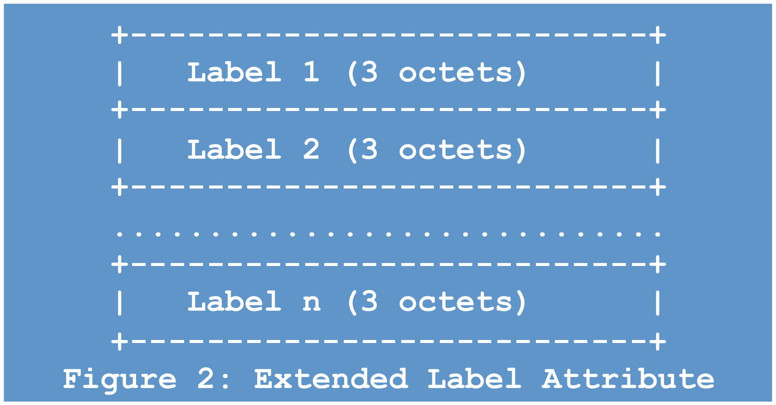
```
+---------------------------+
|   Length  (1 octet)       |
+---------------------------+
|   Label  (3 octets)       |
+---------------------------+
.............................
+---------------------------+
|   Prefix  (variable)      |
+---------------------------+
```

Figure 1: NLRI Definition in RFC3107

- [RFC3107] defines above NLRI to advertise label binding for specific prefix. The label field can carry one or more labels.
- But for other AFI/SAFIs using label binding such as VPNv4, VPNv6, EVPN, MVPN, etc., it dose not support the capability to carry more labels for the specific prefix.
- Moreover for the AFI/SAFIs which do not support label binding capability originally, but may possibly adopt MPLS path programming now, there is no label field in the NLRI.

Download of MPLS Path (Cont.)

- In order to support flexible MPLS path programming, this document defines and uses a new BGP attribute called the "Extended Label attribute".



```
+---------------------------+
|   Label 1  (3 octets)     |
+---------------------------+
|   Label 2  (3 octets)     |
+---------------------------+
 ...........................
+---------------------------+
|   Label n  (3 octets)     |
+---------------------------+
```

Figure 2: Extended Label Attribute

- The Label field carries one or more labels (that corresponds to the stack of labels [[RFC3032]]).
- The Central Controller for MPLS path programming could build a route with Extended Label attribute and send it to the ingress routers.

Download of MPLS Path (Cont.)

- Upon receiving such a route from the MPP Controller, the ingress router SHOULD select such a route as the best path.
- If a packet comes into the ingress router and uses such a path, the ingress router will encapsulate the stack of labels which gets from the Extended Label Attribute of the route into the packet and forward the packet along the path.

- The "Extended Label attribute" can be used for various BGP address families.
- Before using this attribute, firstly, it is necessary to negotiate the capability between two nodes to support MPLS path programming for a specific BGP address family.

Download of Mapping of Service Path to Transport Path

- Since the transport path is also to satisfy the service bearing the requirement, it can also be programmed according to traffic engineering requirements of service. Or the transport path can be set up according to general traffic engineering requirements. Then there needs to be implements the mapping of the service path to the transport path.

- [RFC6514] defines the "P-Multicast Service Interface Tunnel (PMSI Tunnel) attribute". The attribute can not be applied to all possible use cases of service-oriented MPLS path programming.

- This document accordingly defines two new types of BGP attribute for both usage of unicast service path and the multicast service path:
  - ✓ Extended Unicast Tunnel Attribute
  - ✓ Extended PMSI Tunnel Attribute

Extended Unicast Tunnel Attribute

- This document defines and uses a new BGP attribute called the "Extended Unicast Tunnel attribute".

```
+-----------------------------------------------+
| Flags (1 octet)                               |
+-----------------------------------------------+
| Tunnel Type (1 octets)                        |
+-----------------------------------------------+
| Tunnel Identifier (variable)                  |
+-----------------------------------------------+
| Tunnel Specific Attributes (Variable)(Optional)|
+-----------------------------------------------+
```

- The Tunnel Type identifies the type of the tunneling technology used for the unicast service path. The type determines the syntax and semantics of the Tunnel Identifier field. This document defines the following Tunnel Types:

  + 0 - No tunnel information present

  + 1 - RSVP-TE LSP

  + 2 - LDP LSP

  + 3 - GRE Tunnel

  + 4 - MPLS-based Segment Routing Best-effort Path

  + 5 - MPLS-based Segment Routing Traffic Engineering Path

Extended PMSI Tunnel Attribute

- This document defines and uses a new BGP attribute called the "Extended PMSI Tunnel attribute".

```
+------------------------------------------------+
| Flags (1 octet)                                |
+------------------------------------------------+
| Tunnel Type (1 octets)                         |
+------------------------------------------------+
| Tunnel Identifier (variable)                   |
+------------------------------------------------+
| Tunnel Specific Attributes (Variable)(Optional)|
+------------------------------------------------+
```

- The Tunnel Type identifies the type of the tunneling technology used for the multicast service path. The type determines the syntax and semantics of the Tunnel Identifier field. This document defines the following Tunnel Types:

  + 0 - No tunnel information present

  + 1 - RSVP-TE P2MP LSP

  + 2 - mLDP P2MP LSP

  + 3 - PIM-SSM Tree

  + 4 - PIM-SM Tree

  + 5 - BIDIR-PIM Tree

  + 6 - Ingress Replication

  + 7 - mLDP MP2MP LSP

Next Step

- Seek comments and feedbacks
- Revise the draft

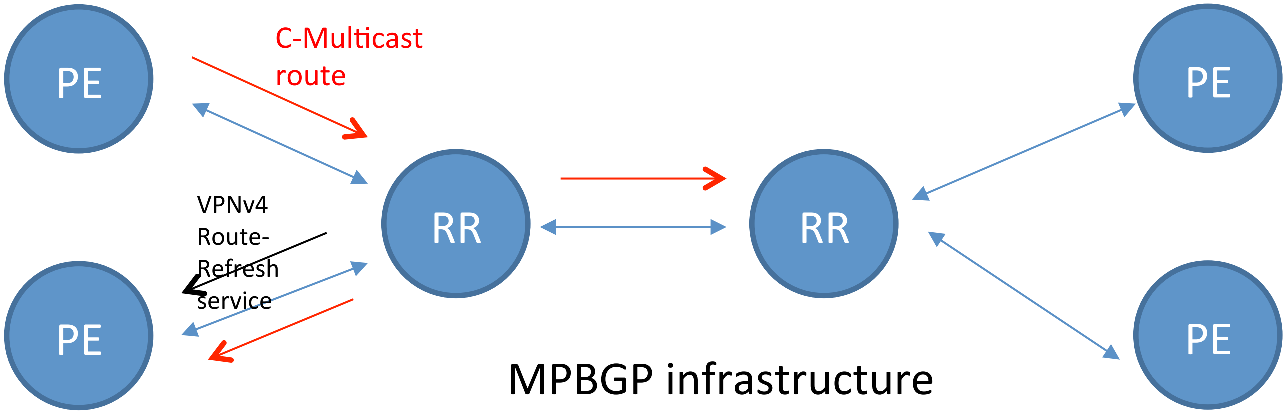
draft-litkowski-idr-bgp-timestamp

S. Litkowski, Orange

K. Patel, Cisco

J. Haas, Juniper

Problem statement



MPBGP infrastructure used for : VPNv4, VPNv6, 6PE, IPv4+label, mvpn

BGP infrastructure is used to transport more and more services.

Service Provider has to ensure that each AFI/SAFI requires guaranteed SLAs (path propagation time)

SLAs have to be monitored

Problem statement

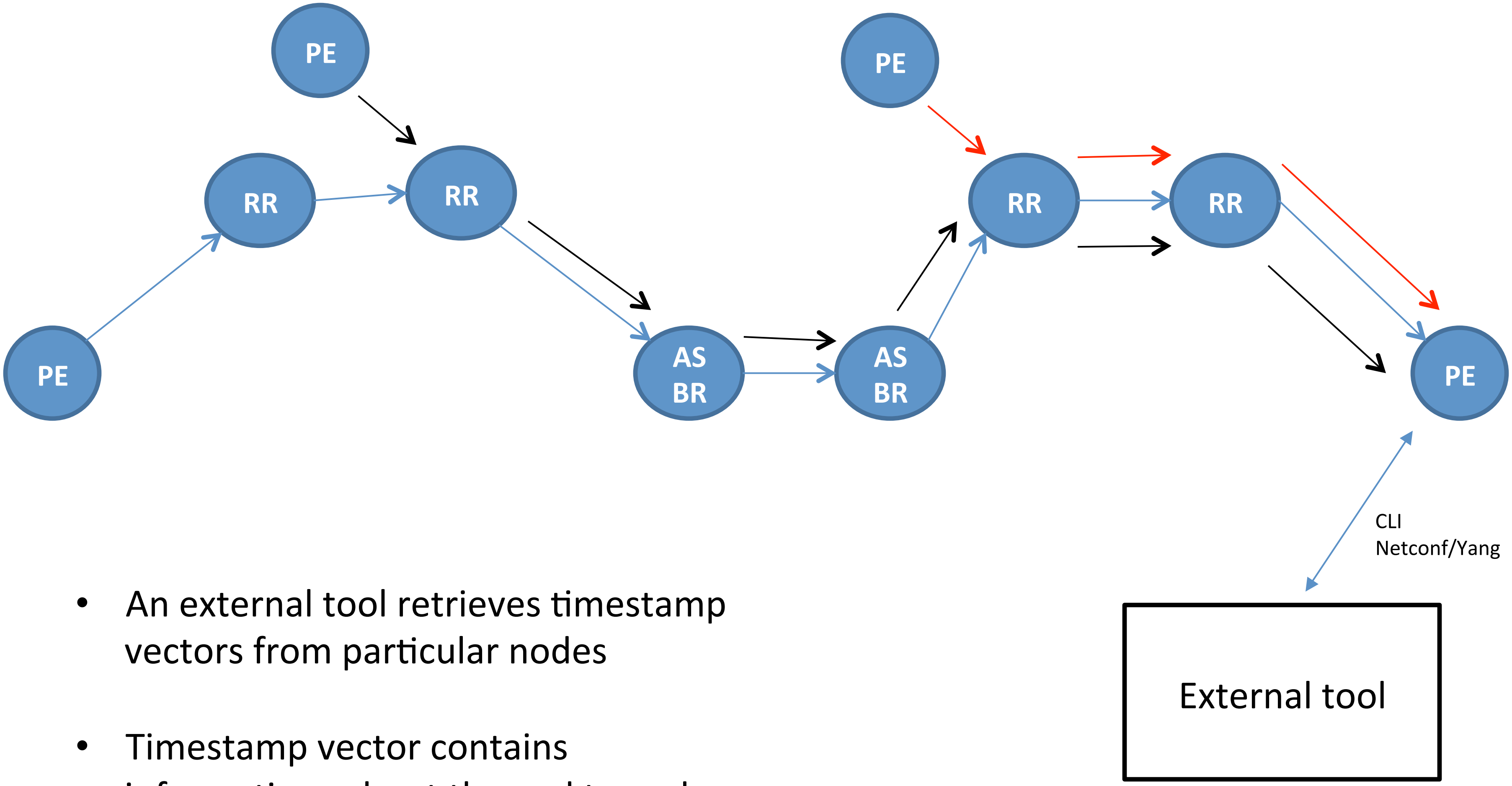
- Service Providers require a solution for monitoring BGP path propagation time :
  - Single point of listening
  - No or very limited correlation need on collector
  - Bottlenecks identification
  - Accuracy of and between measurements (synchronization required)

Proposal

- Add a timestamp vector to BGP path to monitor propagation delay and track bottlenecks

```
    BGP Update      BGP Update       BGP Update      BGP Update
    10.0.0.0/8      10.0.0.0/8       10.0.0.0/8      10.0.0.0/8
    Timestamp:      Timestamp:       Timestamp:      Timestamp:
    R1:T1           R1:T1            R1:T1           R1:T1
                    R2:T2            R2:T2           R2:T2
                                     R3:T3           R3:T3
                                                     R4:T4
R1 ------------> R2 ------------> R3 -----------> R4 ------------> R5
```

Architecture for propagation time measurement



- An external tool retrieves timestamp vectors from particular nodes
- Timestamp vector contains informations about the end to end propagation

Proposal

- Timestamp attribute (Optional transitive)

Type of originator (AS or peer)

Ordered list of timestamp entries

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| OType         |    Originator (variable)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Timestamp #1  (variable)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Timestamp #2  (variable)                       |
...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Timestamp #n  (variable)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Timestamp entry :

Flags :
A : AS type (AS2 or AS4)
P : Peer type (IPv4, IPv6)
S : Summary
T : Synchronized

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Receive Timestamp #x                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|A|P|S|T| Rsvd  |   SyncType    |         AS#x (variable)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Peer#x (variable)                        |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Proposal

- Inspection list :
  - Timestamp are added only to a subset of BGP path matched by the inspection list (filter)
  - By default, do not timestamp

- Receiving a BGP path :
  - If BGP speaker supports BGP TS and path matches inspection list, timestamp is done :
    - If BGP-TS attr does not exist, it is created
    - If BGP-TS attr exists, new timestamp entry is added
    - Timestamp added is receive timestamp
  - If BGP speaker does not support BGP TS, it follows RFC4271 (transitive attribute)

Proposal

- Sending BGP path :
  - We suggest to send BGP TS attribute to only peers configured locally with a « send timestamp » option

Inter AS

- Service provider may not want to expose its timestamp information to external peers

- Three options available :
  - Propagate : propagate TS vector as for internal peers (all details provided)
  - Drop : strip BGP-TS attribute
  - Summary : modify TS vector by aggregating local AS entries into a single summary AS entry (Use S bit)

Compared to BMP

- BMP does not mandate timestamps
- To retrieve timestamp vector, a BMP session would be required to each node including correlation in the external tool
- BMP does not provide information about synchronization state of the peer (is it free run or NTP ?)
- BMP basically dumps all received updates, strong filtering is required to catch interresting NLRIs

Next steps …

- We <u>need a solution to monitor BGP update</u> propagation time

- First shot proposal and we have issues to solve with current draft :
  - NLRI with multiple originators and best path change over time (may lead to advertise stale timestamps)
  - Churn introduced by adding a new « variable » attribute

- WG feedback ?

draft-litkowski-idr-flowspec-interfaceset
IETF 90 - Toronto

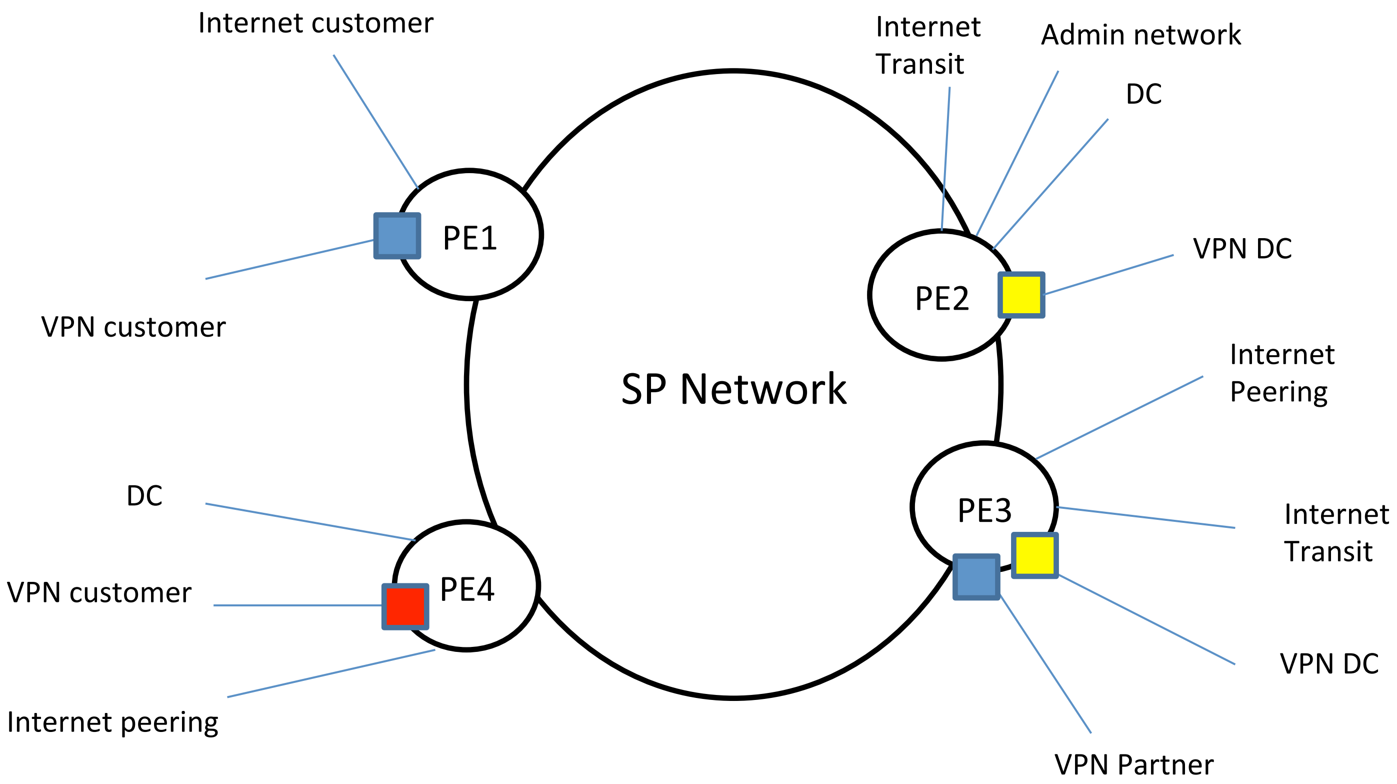
S. Litkowski, Orange

A. Simpson, ALU

K. Patel, Cisco

J. Haas, Juniper

Problem statement



Multiple outside connections in the network

How to deploy <u>specific Flowspec rules</u> on a <u>specific set of interfaces ?</u>

Use cases

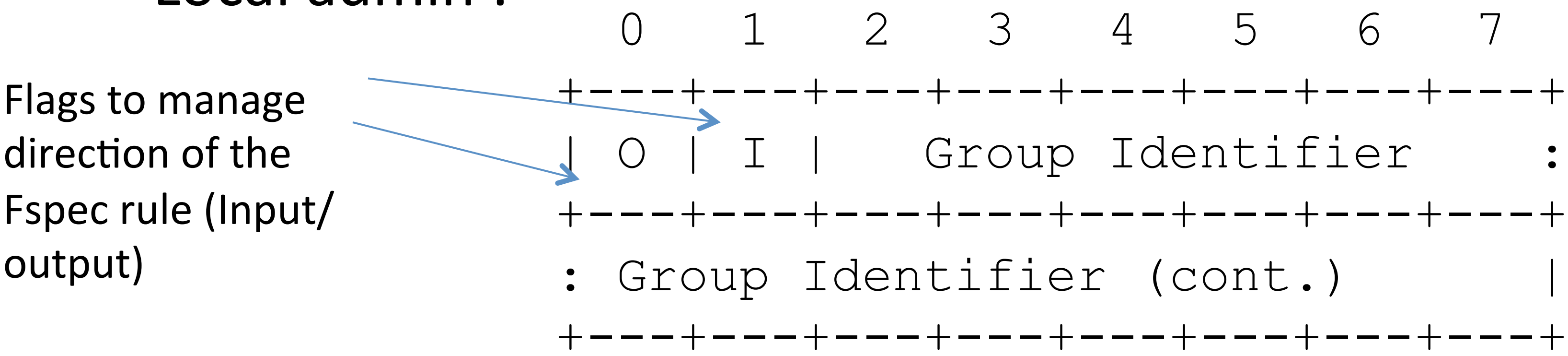
- Specific filtering for DDoS prevention :
  - Maintain rate-limiting rules (NTP, DNS …) on ISP connections based on interface BW

- Infrastructure ACL management
  - Complete management of infra ACL : all the ACL is maintained through a list of Fspec rules. Each interface type has its own Fspec rule set.

  - Quick update of CLI based ACLs (security alert …)

interface-set extended community

- Transitive 4-B AS-specific extended community
  - Global admin : ASN of the originating router
  - Local admin :

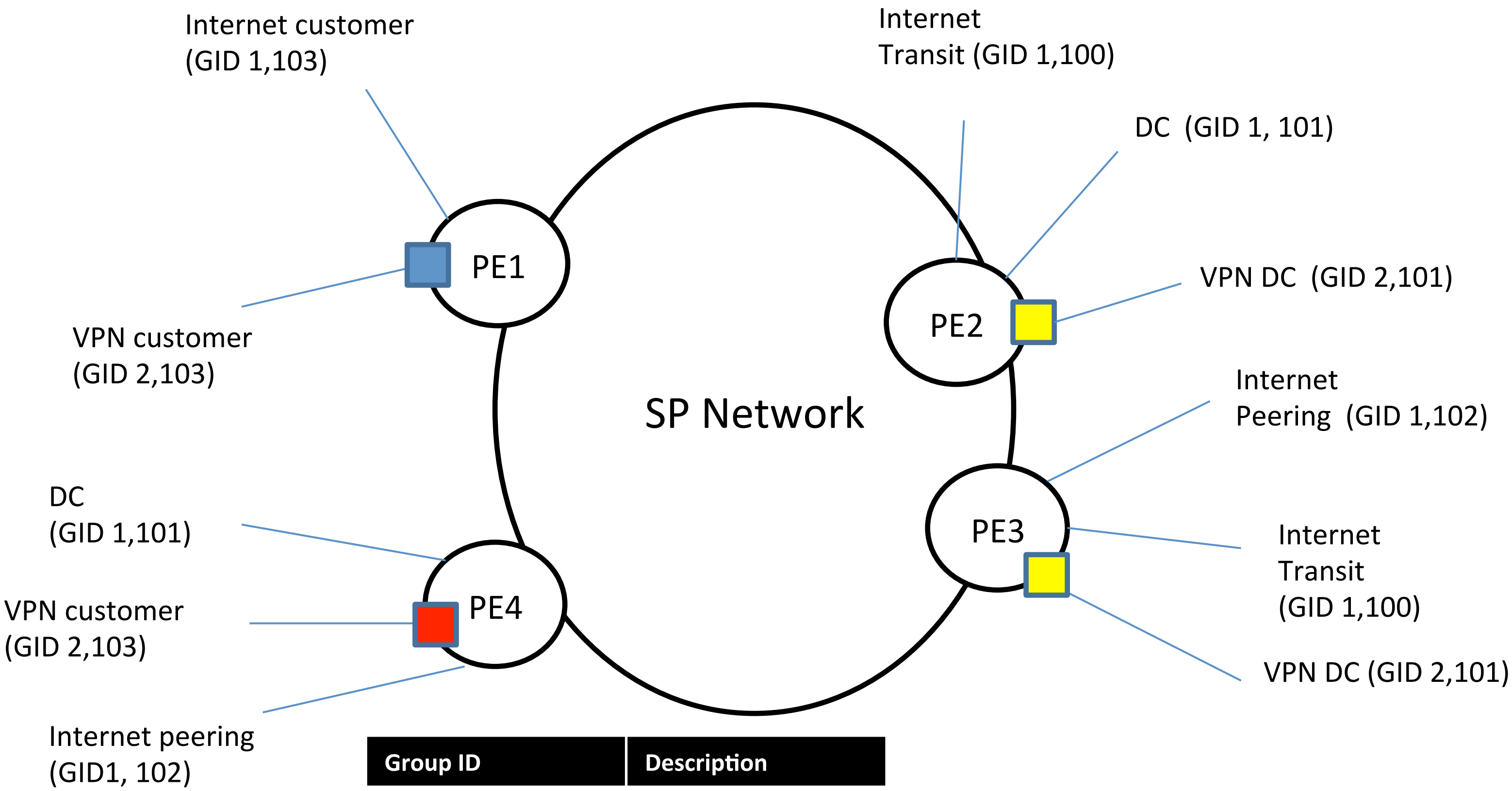


Flags to manage direction of the Fspec rule (Input/ output)

```
  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
| O | I |   Group Identifier    :
+---+---+---+---+---+---+---+---+
: Group Identifier (cont.)      |
+---+---+---+---+---+---+---+---+
```

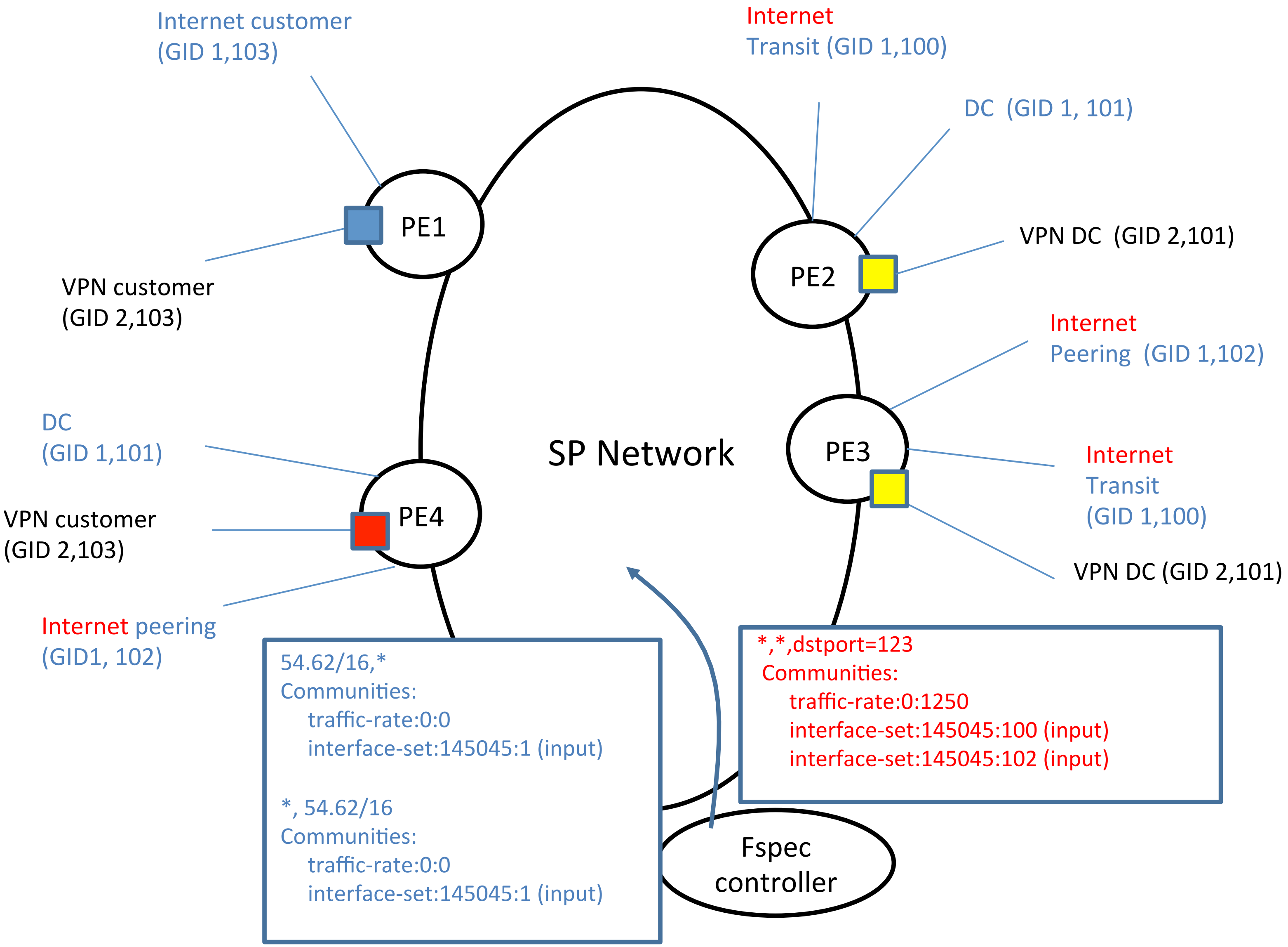
- Multiple interface-set on the same Fspec NLRI means « match-any »

Example



Group ID	Description
1	Internet connection
2	VPN connection
100	Transit connection
101	DC connection
102	Peering connection
103	Customer

Example



Internet customer
(GID 1,103)

Internet
Transit (GID 1,100)

DC (GID 1, 101)

PE1

PE2

VPN DC (GID 2,101)

VPN customer
(GID 2,103)

Internet
Peering (GID 1,102)

DC
(GID 1,101)

SP Network

PE3

Internet
Transit
(GID 1,100)

VPN customer
(GID 2,103)

PE4

VPN DC (GID 2,101)

Internet peering
(GID1, 102)

54.62/16,*
Communities:
traffic-rate:0:0
interface-set:145045:1 (input)

*, 54.62/16
Communities:
traffic-rate:0:0
interface-set:145045:1 (input)

,,dstport=123
Communities:
traffic-rate:0:1250
interface-set:145045:100 (input)
interface-set:145045:102 (input)

Fspec
controller

Security considerations

- Managing infra ACLs using Fspec may be dangerous as Filters are ephemeral (linked to life of the BGP path)

- An attacker may break Fspec BGP session and open all the network doors (probability low …)

- LLGR for FSpec AFI/SAFIs would help to make filters more persistent

Discussions outcomes

- Some text to fix :
  - Community format error handling (both flags set to 0)
  - Logical operation to clarify when having multiple interface-set for a single NLRI
- Encoding :
  - Using wide-communities would help (more flexibility in encoding and group logic)
  - Using wide-community is a good idea but would slow down availability of the use case (wide-comm specification is not yet finalized)
  - Authors would prefer to use existing communities for now. Wide communities could be used in addition when available to bring more flexibility.

Discussions outcomes

- Define assigned interface-sets ??
  - Idea is that system automatically binds some group ID to an interface

Registry Name: Assigned Flow spec interface-set

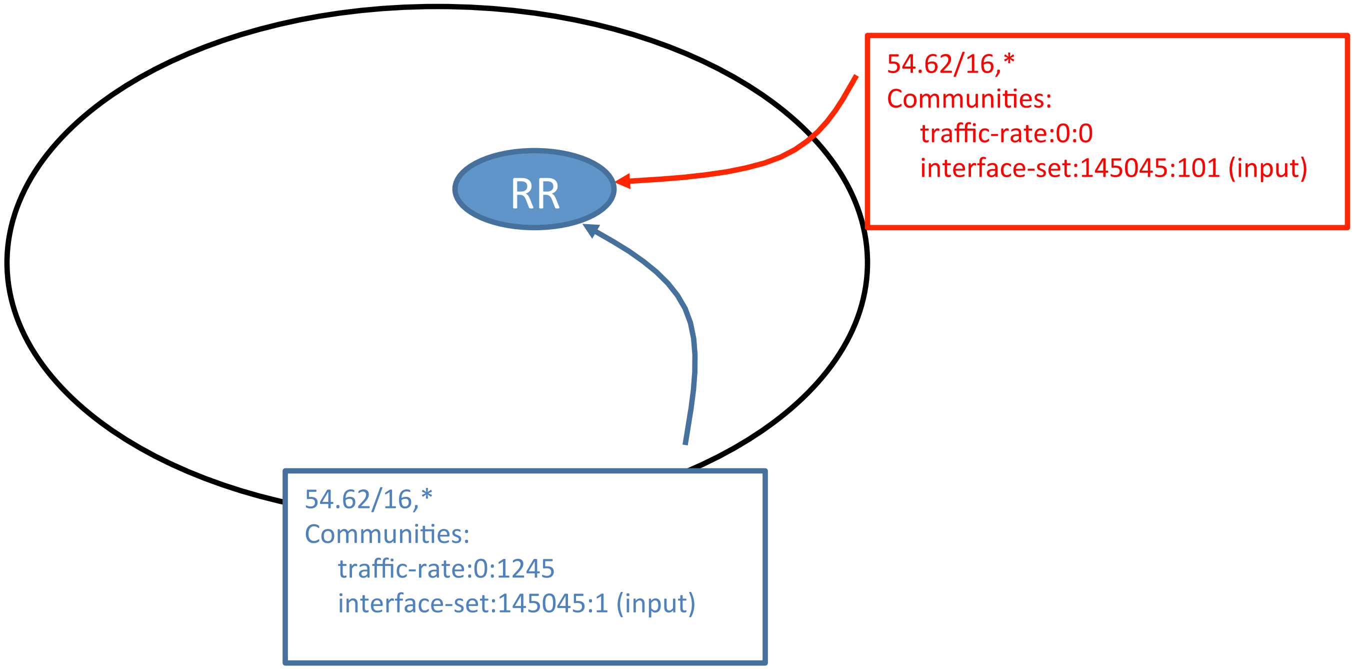
Range	Registration Procedures
0x0000-3EFF	Reserved for private use.
0x3F00-3FFF	Standards Action/Early IANA Allocation.

The IANA is requested to update the registry "Assigned Flow spec interface-set" as follows:

0x3F00: IGP interface
0x3F01: non IGP interface
0x3F02: eBGP interface
0x3F03: non eBGP interface
0x3F04: VRF interface
0x3F05: non VRF interface

Issue to solve

- What about multi-originator scenario ?
  - The issue is not linked to interface-set
  - Already present with basic RFC5575



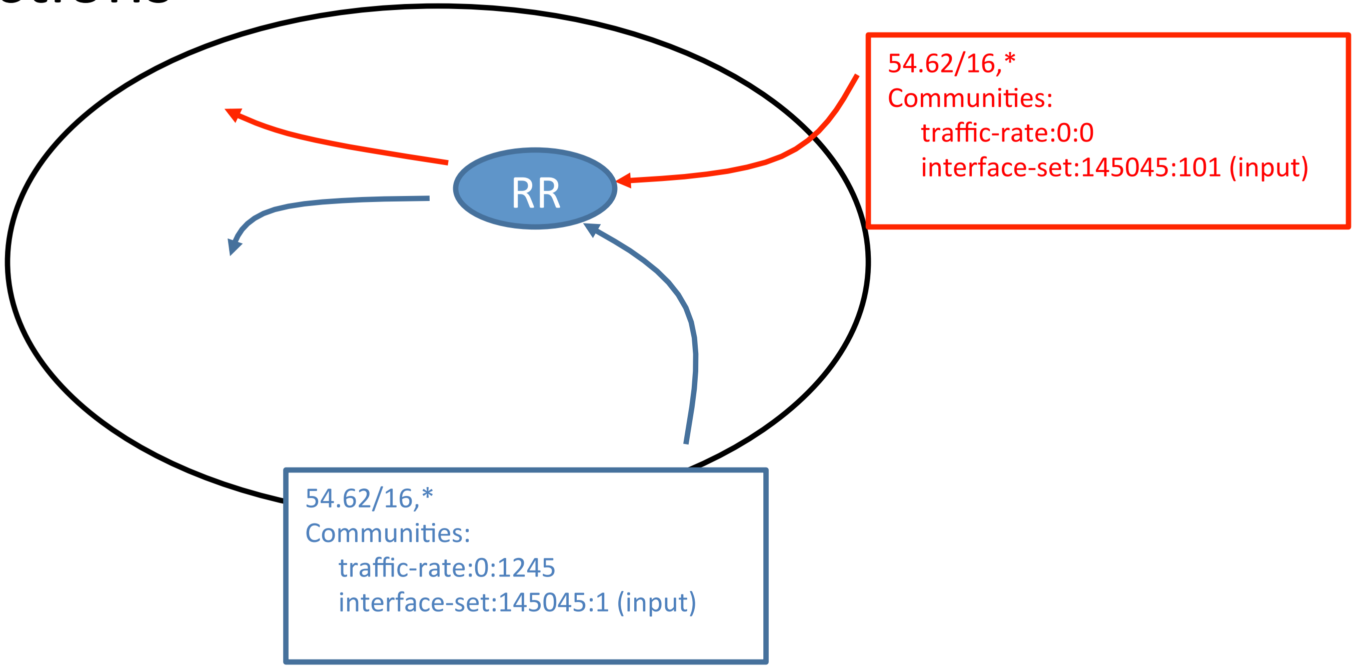
RR

54.62/16,*
Communities:
traffic-rate:0:0
interface-set:145045:101 (input)

54.62/16,*
Communities:
traffic-rate:0:1245
interface-set:145045:1 (input)

Issue to solve

- Multi-originator issue could be partially solved using ADD-PATH
  - But there is still a need to handle conflicting actions



RR

54.62/16,*
Communities:
traffic-rate:0:0
interface-set:145045:101 (input)

54.62/16,*
Communities:
traffic-rate:0:1245
interface-set:145045:1 (input)

Next steps …

- Requires feedback from WG

- Address comments from the list in next version

AS Migration draft-ietf-idr-as-migration-01

Wes George

Shane Amante

Changelog

- Add RFC2119 boilerplate and two implementation sections (3.2 and 4.2) with normative language describing how to implement the features described by the document
- Updated vendor-specific documentation references
  - Moved to an implementation report appendix
- Incorporates review feedback from several off-list reviews from Cisco and Juniper folks. – Thanks!

Open items for WGLC

- Doc status – Info or PS?
  - Need feedback now that it's actually defining the behavior in normative language
- Does this need to formally update RFC4271 or others?
  - Interop not strictly required (locally significant to a given BGP speaker)
  - Not mandatory to implement
- Terminology – C vs. J vs. "other"