IDR Chair Slides: Agenda & Status

Sue Hares and John Scudder

3/24/2015

15:20 – 17:20

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

All IETF Contributions are subject to the rules of RFC 5378 and RFC 3979 (updated by RFC 4879).

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.

Chair's Status

Shepherds

- **Drafts at RFC Editor**
  - draft-ietf-idr-as0-06
- **Drafts at IESG**
  - draft-ietf-idr-error-handling-18 [Rob Shakir]
  - draft-ietf-idr-flowspec-redirect-rt-bis-03 [Mach Chen]
- **Drafts at IETF LC**
  - draft-ietf-idr-ls-distribution-10 [Jie Dong]
- **Drafts awaiting revision**
  - draft-ietf-idr-ls-distribution-impl-03 [Jie Dong]
  - draft-ietf-idr-as-migration-03 [Chris Morrow]

Document status

- WG Consensus (WG LC) in process to go to IESG
  - draft-ietf-idr-add-paths-10
  - draft-ietf-idr-add-paths-guidelines-07
  - draft-ietf-idr-ix-bgp-route-server-06
  - draft-ietf-idr-sla-exchange-04
- WG documents adopted
  - draft-ietf-idr-te-lsp-distribution-02
  - draft-ietf-idr-rtc-no-rt-00
  - draft-ietf-idr-rtc-hierarchical-rr-00
  - draft-ietf-idr-route-oscillation-stop-00
  - draft-ietf-idr-flowspec-l2vpn-00
  - draft-ietf-idr-flowspec-redirect-ip-02
  - draft-ietf-idr-performance-routing-01
  - Draft-ietf-idr-rs-bfd-00 (draft-ymbk-idr-rs-bfd-00)

Document status

- WG Consensus (WG LC) in process to go to IESG
  - draft-ietf-idr-add-paths-10
  - draft-ietf-idr-add-paths-guidelines-07
  - draft-ietf-idr-ix-bgp-route-server-06
  - draft-ietf-idr-sla-exchange-04

IDR work for May

- WG LC planned in April
  - draft-ietf-idr-add-paths-implementation-00
  - draft-ietf-idr-ls-distribution-impl-03
  - draft-ietf-idr-as-migration-04) (revision)
  - draft-ietf-idr-reserved-extended-communities-08
- WG LC planned in May
  - draft-ietf-idr-custom-decision-05
  - draft-ietf-idr-rtc-no-rt-00
- WG LC requires
  - 2 implementation
  - Summary presentation at Interim
  - Request to Chairs

Virtual Interims Planned for 4/1 – 6/15

Monday at 10:00-11:30 am ET

(Beijing 10:00-11:30pm/ CET: 16:00-17:30/ 07:00-08:30 PT)

- 4/13 - BGP Yang modules
- 4/27 – Route Server + oscillation drafts + Communities drafts
- 5/11 – Flow Specification + Custom Design + RTC
- 6/01 - NextHop issues + Traffic Engineering drafts
- 6/15 - TBD

Agenda (1)

- 1) Agenda Bashing/document Status [5 minutes]

AS and Communities Discussions [30 minutes]

2) AS-Migration [Wes George] [10 minutes]
[draft-ietf-idr-as-migration]

3) BGP Wide Communities [10 minutes]
draft-raszuk-wide-bgp-communities

4) BGP time stamp: Update [Stephane Litowski] [5 min.]
draft-litkowski-idr-bgp-timestamp/

5) draft-litkowski-idr-rtc-interas: [Stephane Litowski] [5 min.]
draft-litkowski-idr-rtc-interas/

IDR Agenda (2)

Flow Spec and Segment Routing [30 minutes]

- 6) draft-litkowski-idr-flowspec-interfaceset [Stephane Litkowski] [5 min]

- 7)draft-previdi-idr-bgpls-segment-routing-epe [Stefano Previdi] [10 min]

- 8) Advertising Per-node Admin Tags in BGP Link-State Advertisements [Pushpasis Sarkar] [10 minutes]
  draft-psarkar-idr-bgp-ls-node-admin-tag-extension/

NextHop [10 minutes]

- 9) draft-decraene-idr-next-hop-capability-00 [Bruno B. Decraene] [10 minutes] draft-decraene-idr-next-hop-capability/

IDR Agenda (3)

Yang Section [10 minutes]

- 10) OpenConfig BGP Config Model Update [Anees Shakih] [7 minutes]

  draft-shaikh-idr-bgp-model/

- 11) Zhdankin BGP Config Model Update [Keyur Patel] [3 minutes]

  draft-zhdankin-idr-bgp-cfg/

BGP Controls [10 minutes]

- 12) Route Leak Detection [K. Sriram] [10 minutes]
  - http://tools.ietf.org/html/draft-sriram-idr-route-leak-detection-mitigation-00
  - [companion draft in grow http://tools.ietf.org/html/draft-ietf-grow-route-leak-problem-definition-01]

IDR agenda (4)

- New Concepts [30 minutes]
- 13) draft-li-spring-mpls-path-programming-01 [Shunwan Zhang] [10 minutes]

- 14) BGP-LU for HSDN Label Distribution [Luyuan Fang] [10 min.]
  draft-fang-idr-bgplu-for-hsdn-00 [10]

- 15) BIER: Bit Index Explicit Replication (BIER)
  - Xiaohu Xu [10 minutes]
  - https://tools.ietf.org/html/draft-xu-idr-bier-extensions-00

IDR-AS-MIGRATION

IETF 92

Wes George

Shane Amante

Substantial Issues from IESG Eval

- No, really, are you **sure** this needs to be PS?
  - No interop…
  - Resolved? DISCUSS is gone
- Too much focus on business justification/drivers in the intro for a protocol document
- Too much focus on documenting how existing implementations work
- Examples using specific vendor's CLI/implementation, failure to document all vendors' implementations equally looks like favoritism/sales pitch

Proposed resolution - Intro

- Easy: tone down/remove language about billing (95 percentile, etc) as it relates to as-path length
- Harder: remove majority of justification of why this set of tools is important to operators.

Discussion:

- Provided extra justification for why IETF should do this work since this draft was “weird”
  - documenting existing feature, didn’t technically require interop
- Is “Operators use this and IETF shouldn’t break it” enough by itself?
- All text will be visible in old versions for historical record, does the WG want it in the RFC?

Proposed resolution – Vendor implementations

- Easy: Do nothing. These are no objection/abstain comments, so the draft could be pushed forward after revision to address other minor technical comments.
- Harder: Make an editing pass to reduce vendor-specific implementation references where possible
- Hardest: Rewrite normative section of document to specify what we want as if existing implementations don't exist, limit discussion of current implementations to an appendix implementation report

Proposed resolution – Vendor implementations

- Hardest: Rewrite normative section of document to specify what we want as if existing implementations don't exist, limit discussion of current implementations to an appendix implementation report

Discussion:

- Generic normative implementation results in most/all vendors' existing implementations being out of compliance with an *ex post facto* IETF standard
  - Do the vendors care? Does IETF?
- Is the current form "good enough" to meet the goal to document existing behavior?

Discussion

Photo:
Jared Mauch

Wide BGP Communities (update to ver -05)

draft-raszuk-wide-bgp-communities-05

Robert Raszuk, Jeff Haas, Andrew Lange, Shane Amante, Richard A. Steenbergen, Bruno Decraene, Paul Jakma, Shintaro Kojima, Juan Alcaide, Burjiz Pithawala, Saku Ytti

IETF 92, March 2015, Dallas, TX

Registered Wide BGP Communities

draft-raszuk-registered-wide-bgp-communities-00

Robert Raszuk, Jeff Haas, Richard Steenbergen, Bruno Decraene, Paul Jakma
Shintaro Kojima, Juan Alcaide, Burjiz Pithawala, Saku Ytti + **IDR members**

IETF 92, March 2015, Dallas, TX

Agenda

- Objective
- History
- Encoding
- Companion document

Registered Wide BGP Communities

Goals

- To define a new encoding which will allow operators much more flexible network control while in the same time simplify the amount of required inbound and outbound route policy
- To enable propagation of arbitrary set of targets and parameters which will be used during given policy execution
- To provide ability to use customer's own definition of parameterized communities as well as set of IANA maintained predefined registered wide bgp communities.
- Not intended to replace standard or extended BGP communities

History

- Presented originally as single draft at Maastricht IETF 78 2010 as work with Hannes G.

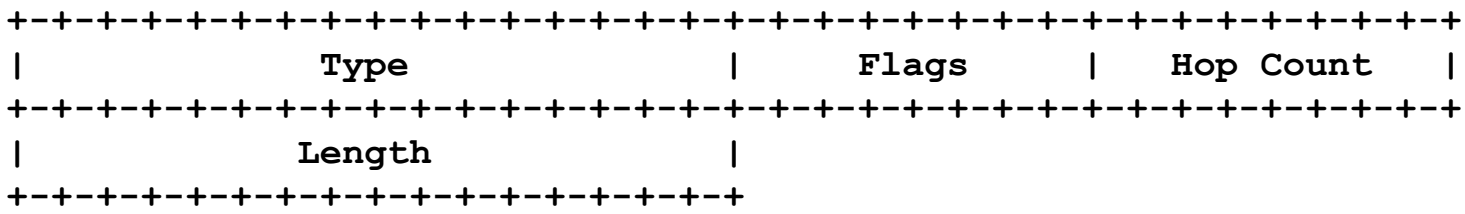
Recommendation of the IDR WG was to split the work into base spec & IANA registered set of wide communities

- Presented in Beijing IETF 79 2010 of splitted work.

Integrated with former related work lead by Andrew Lange (flexible-communities), started a lot of community discussions on encoding (attempt to encode handling algebra) & collect requirements for most useful registered values.

Encoding

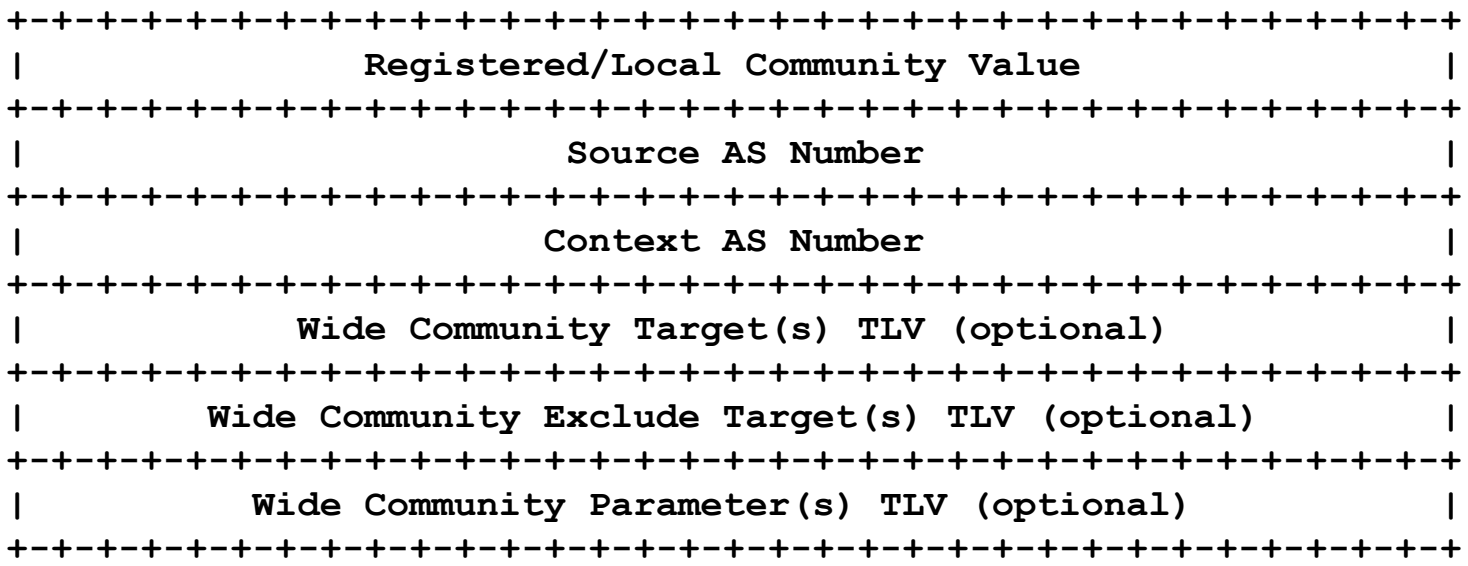
- New BGP attribute (optional, transitive):



- Defined flags: local/register & decrement or not hop count across confederation boundaries
- Hop count: propagation radius. 0 - do not propagate across EBGP. MUST be decremented when traversing EBGP boundary.

Encoding

- Container type 1:

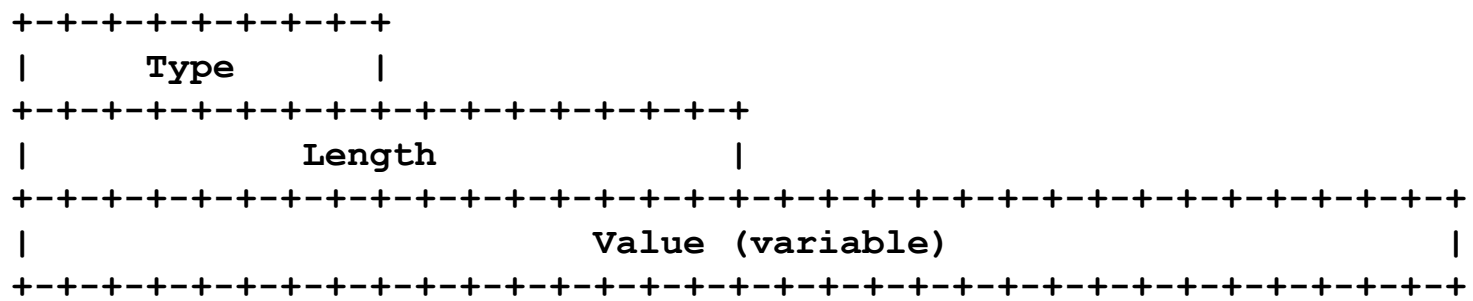


- Context AS: AS number which defined and published given local community value (for peers, customers or upstreams). For registered communities unless (re)defined MUST be 0.

- Targets and Parameters: TLVs containing atoms (zero or N) which carry values used in executing given community.

Encoding

- Atom encoding:



- Type 1: Autonomous System number list
- Type 2: IPv4 prefix (1 octet prefix length + prefix) list
- Type 3: IPv6 prefix (1 octet prefix length + prefix) list
- Type 4: Integer list
- Type 5: IEEE Floating Point Number list
- Type 6: Neighbor Class list
- Type 7: User-defined Class list
- Type 8: UTF-8 String

Registered Wide BGP Community Values

draft-raszuk-registered-wide-bgp-communities-00** (should be -02)

Registered Wide BGP Communities

Type	Name	Atom types used
1.	BLACKHOLE	- / - / -
2.	SOURCE FILTER	- / - / -
3.	SOURCE DO RPF	- / - / -
4.	HIGH PRIORITY PREFIX	- / - / -
5.	ATTACK TARGET	- / - / -
6.	NO ADVERTISE TO AS	1 / - / -
7.	ADVERTISE TO AS	1 / - / -
8.	ADVERTISE AND SET NO EXPORT	1 / - / -
9.	FROM PEER	- / - / -
10.	FROM CUSTOMER	- / - / -
11.	INTERNAL	- / - / -
12.	FROM UPSTREAM	- / - / -

Registered Wide BGP Communities

Type	Name	Atom types used
13.	FROM IX	- / - / -
14.	LEARNED FROM AS	1 / - / -
15.	PATH HINT	1 / - / -
16.	NEGATIVE PATH HINT	1 / - / -
17.	PREPEND N TIMES BY AS	1 / - / 4
18.	PREPEND N TIMES TO AS	1 / - / 4
19.	REPLACE BY	1 / - / -
20.	LOCAL PREFERENCE	- / - / 4
21.	AS PATH TTL MAX RADIUS	- / - / 4
22.	GEO-LOCATION	- / - / 8(x5)
23.	Free pool	

Conclusion

- Both drafts have been stable since Berlin IETF when authors converged on type 1 encoding.
- Other types if proposed to be defined in separate documents
- Authors feel drafts are ready and ask for call on the list regarding adoption as IDR WG documents.

draft-litkowski-idr-bgptimestamp-01

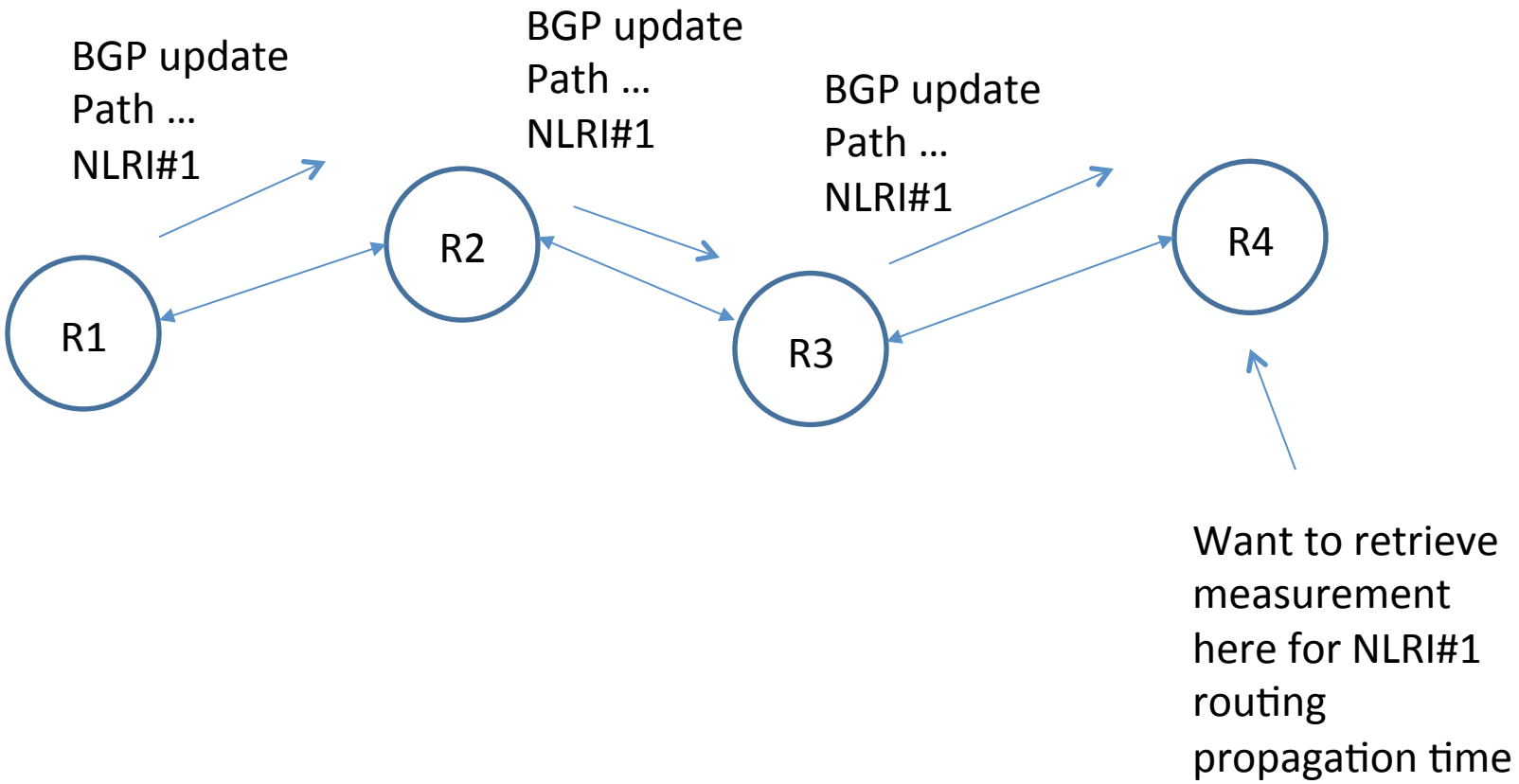
S. Litkowski

J. Haas

K. Patel

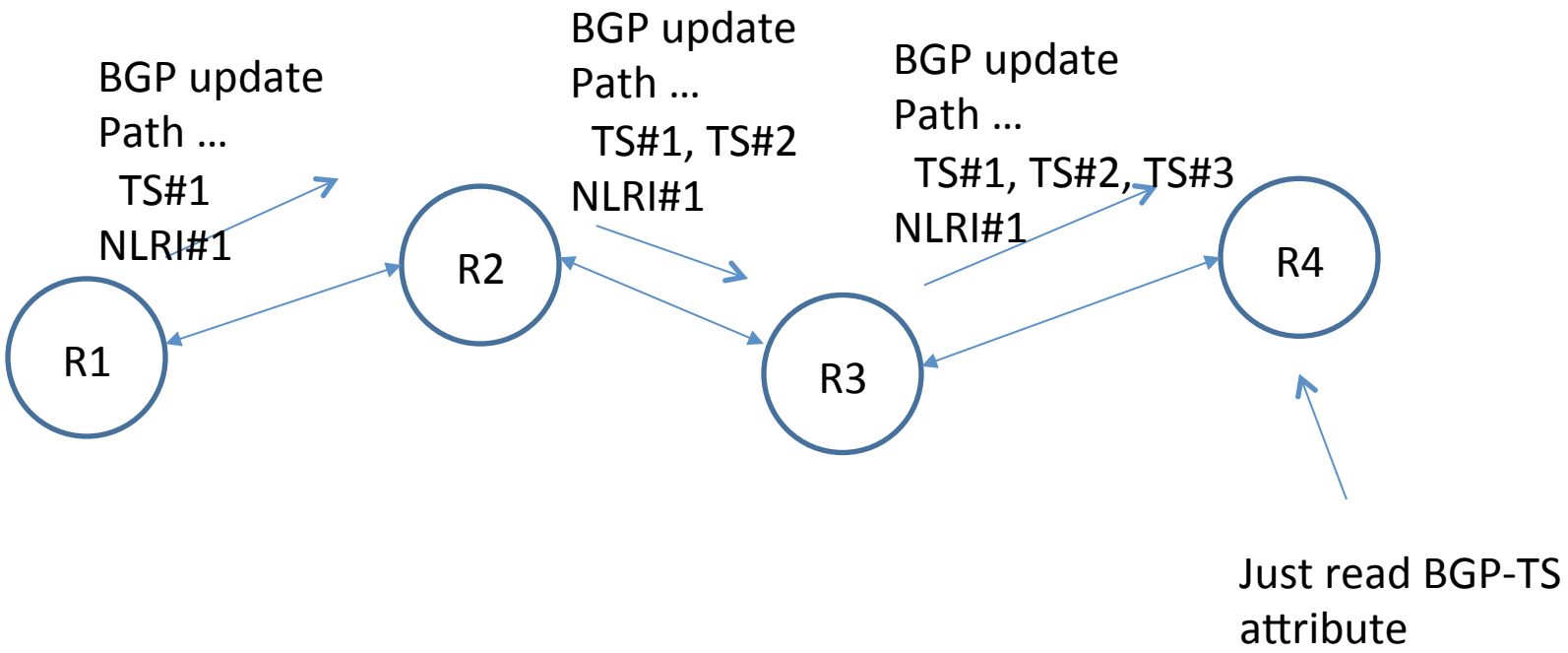
Problem statement

- Need to be able to measure propagation time of routing information



Proposed solution

- New BGP attribute : BGP-TS
- Use a timestamp vector that will be updated by each BGP Speaker in the path vector.



Changes from -00

- We took into account comments from WG :
  - Adding send timestamp as it sounds possible from implementation point of view, also giving some constraint on where the send timestamp should be applied
  - Add a section about limiting churn :
    - BGP-TS attribute must not be taken into account when evaluating the need to send a new update (prevents duplicate updates that would just update timestamp)
  - Add a section about update packing :
    - Not considered as critical as we are targeting some NLRIs to be monitored, so only those NLRIs will not be packed
  - Changed encoding to fit send timestamp and more optimized (may require still some work, but not a focus now)

Changes from -00

- Adding procedures to manage stale timestamp :
  - Stale indicator
  - Stale indicator is inserted when :
    - A path is received or originated, and decision process does not select it as best
    - A path is received or originated, and decision process select it as best, the path must be exported and then stale indicator is inserted.

Next step

- Requested feedback from BMWG about accuracy of the solution => no feedback yet

- Comments from WG required to progress
  - Do you find it accurate enough ?
  - How would it impact implementations ?
  - Others ?

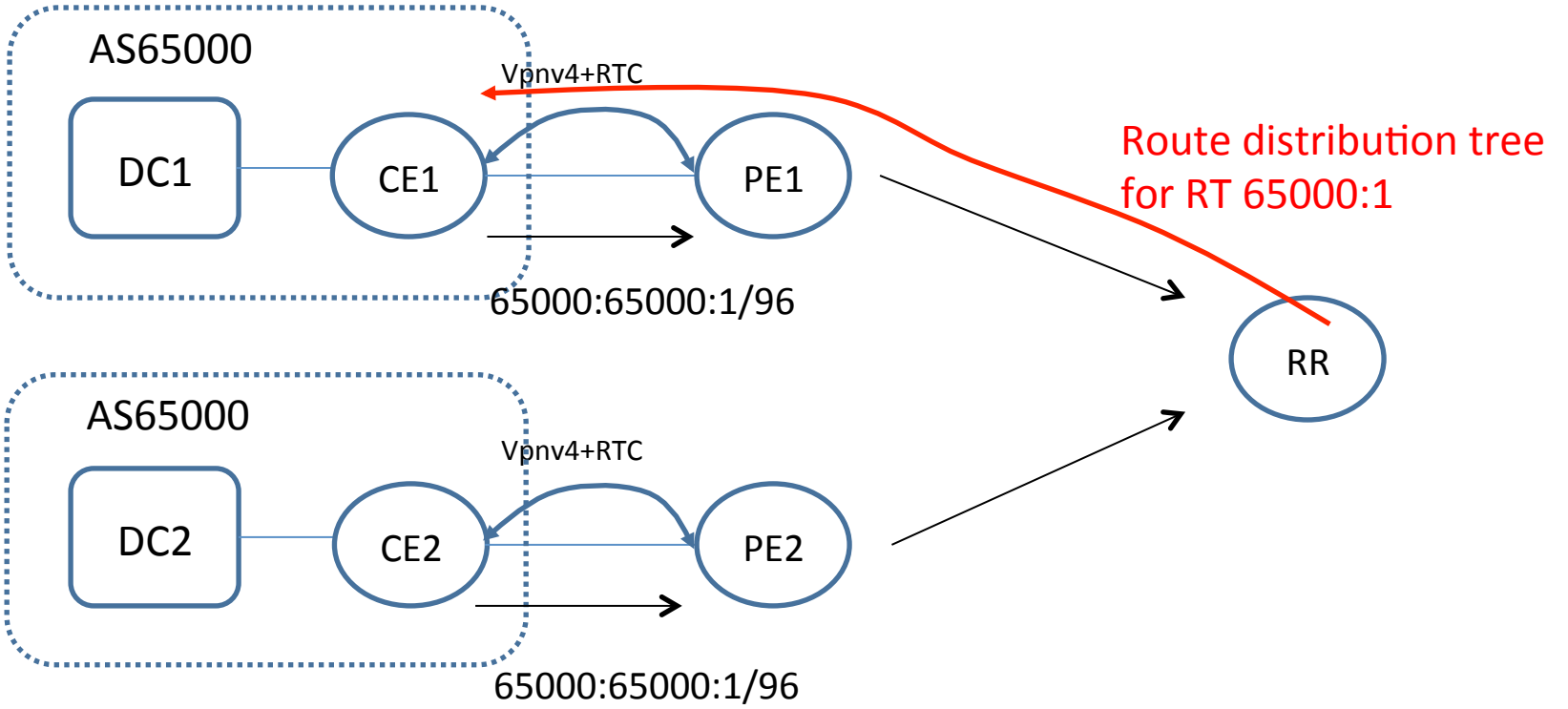
draft-litkowski-idr-rtc-interas-01

S. Litkowski

J. Haas

K. Patel

Problem statement



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

Changes btw -00 and -01

- Complete rewrite of the problem statement
- Explaining that RFC4684 can be interpreted in two ways regarding pruning that are both compliant :
  - Peering type based pruning
  - NLRI type based pruning

- Explaining pros and cons of both approaches
- We keep the proposal from -00 concerning NLRI type based pruning :
  - Authorize to disable pruning for specific ASes or all private ASes

Next step

- Comments from WG on the way we explain the problem ?
- Is the solution fine ?

- Ask for WG adoption

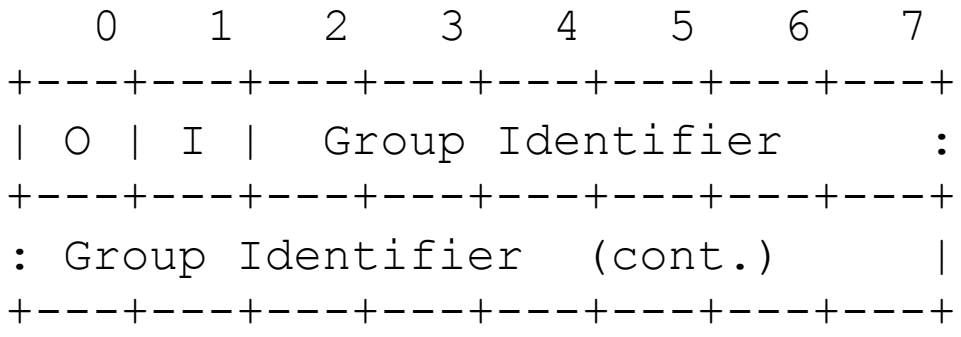
draft-litkowski-idr-flowspec-interfaceset-02

S. Litkowski
J. Haas
K. Patel
A. Simpson

Goal

- Provide to apply selectively filtering rules to specific set of interfaces within the network and provide direction of filtering

- Proposed new extCT (no change since last time)



Changes since IETF90

- We took into account comments from WG
  - Provides details on why managing filter direction is useful
  - Provides guidelines of interactions with some permanent traffic actions (ACL, flow collection)
    - Flow collection :
      - Must happened before BGP FS actions at ingress
      - Must happened before BGP FS actions at egress
    - Permanent ACLs :
      - Do not mix FS entries with static entries (ordering issue)
      - Must happened before BGP FS actions at ingress
      - Must happened before BGP FS actions at egress

Next steps

- Welcome comments …
- Would ask WG adoption

IDR WG

Segment Routing BGPLS Egress Peer Engineering Extensions

draft-previdi-idr-bgpls-segment-routing-epe-02

Stefano Previdi (sprevidi@cisco.com)

Clarence Filfsils (cfilsfil@cisco.com)

Saikat Ray (sairay@cisco.com)

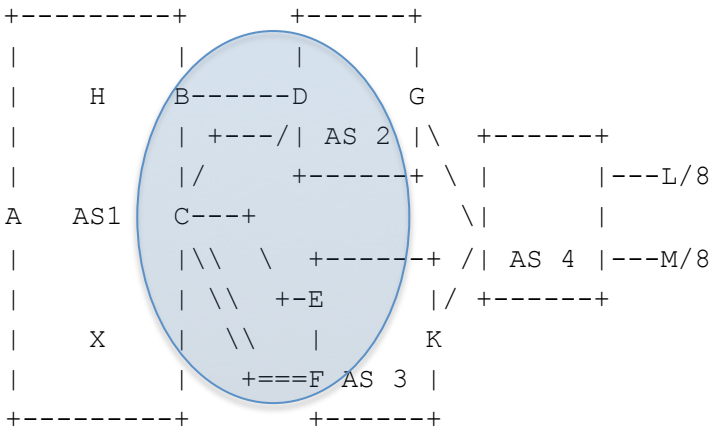
Keyur Patel (keyupate@cisco.com)

Jie Dong (jie.dong@huawei.com)

Mach (Guoyi) Chen (mach.chen@huawei.com)

Motivations

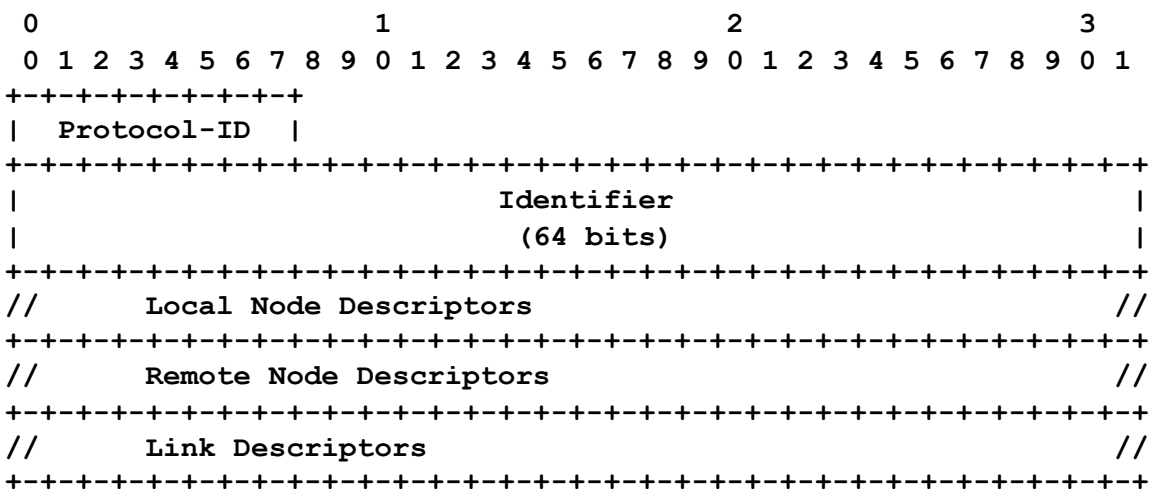
- Problem statement / use case described in draft-filsfils-spring-segment-routing-central-epe



- 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.

-02 Update

- Use of new BGP-LS Protocol ID (TBA) instead of a NLRI-Type



-02 Update

- Description is unchanged from -01
  - Local Node Descriptors
  - Remote Node Descriptors
  - Link Descriptors

Questions?

Thanks!

Advertising Per-node Admin Tags in BGP LS

draft-psarkar-idr-bgp-ls-node-admin-tag-00

Pushpasis Sarkar psarkar@juniper.net
Hannes Gredler hannes@juniper.net
Stephane Litkowski stephane.litkowski@orange.com

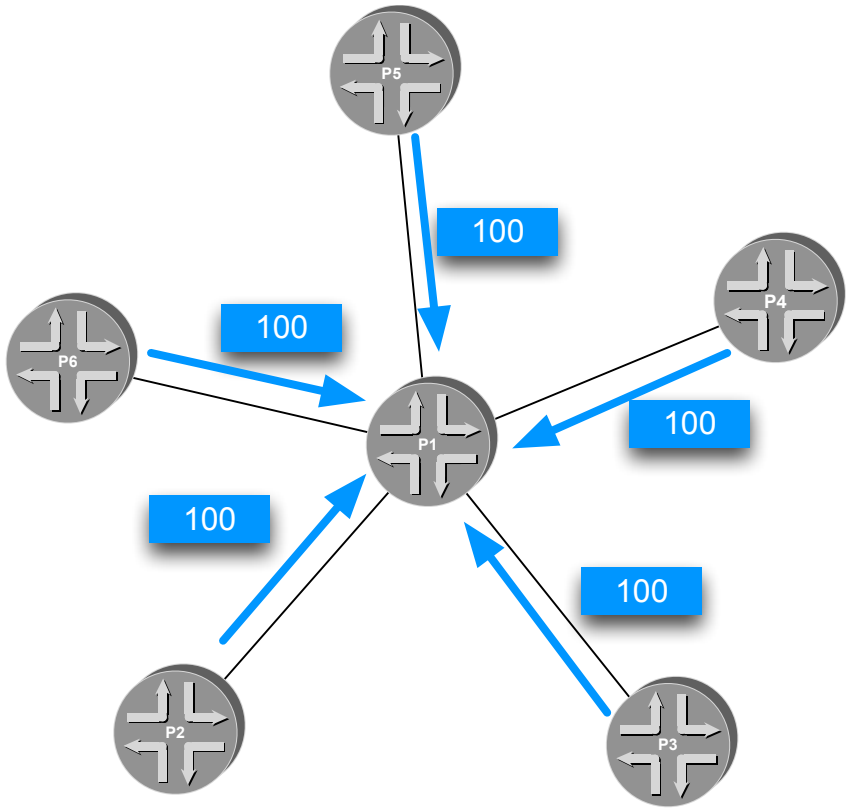
Summary

- Prior Art
- Current Draft Proposal
- Guidelines on Implementation
- Next steps

Prior Art

Why Node-Admin Tag?

- Link colors [RFC 3630, RFC5305]
  - Does not really represent a node characteristic.
  - Even if used to represent node characteristic, ***all incoming links need to be colored*** (one per node characteristic type).



Prior Art

Why Node-Admin Tag?

- Prefix tags [RFC5130]
  - If the router-ID is considered the prefix representing the node
    - Router-ID encoded in TLV134 or TLV242.
    - Corresponding tag encoded in TLVs 135, 235, 236 and 237.
    - ***Additional implementation complexity***
  - No prefix tagging mechanism for OSPF yet
    - Looking for ***consistency across protocols***
    - draft-ietf-ospf-node-admin-tag-00
  - Most Traffic Engineering Database (TED) schema support
    - ***Nodes***
    - ***Links***
    - But ***not Prefixes***

Prior Art

- Per-Node Admin Tags
  - Introduced in IETF-89
    - [I-D.ietf-isis-node-admin-tag]
    - [I-D.ietf-ospf-node-admin-tag]
  - 32-bit unsigned integer value.
  - Represents a specific node characteristic exhibited by one or more nodes in the network
    - One tag per type of node characteristic.

Prior Art

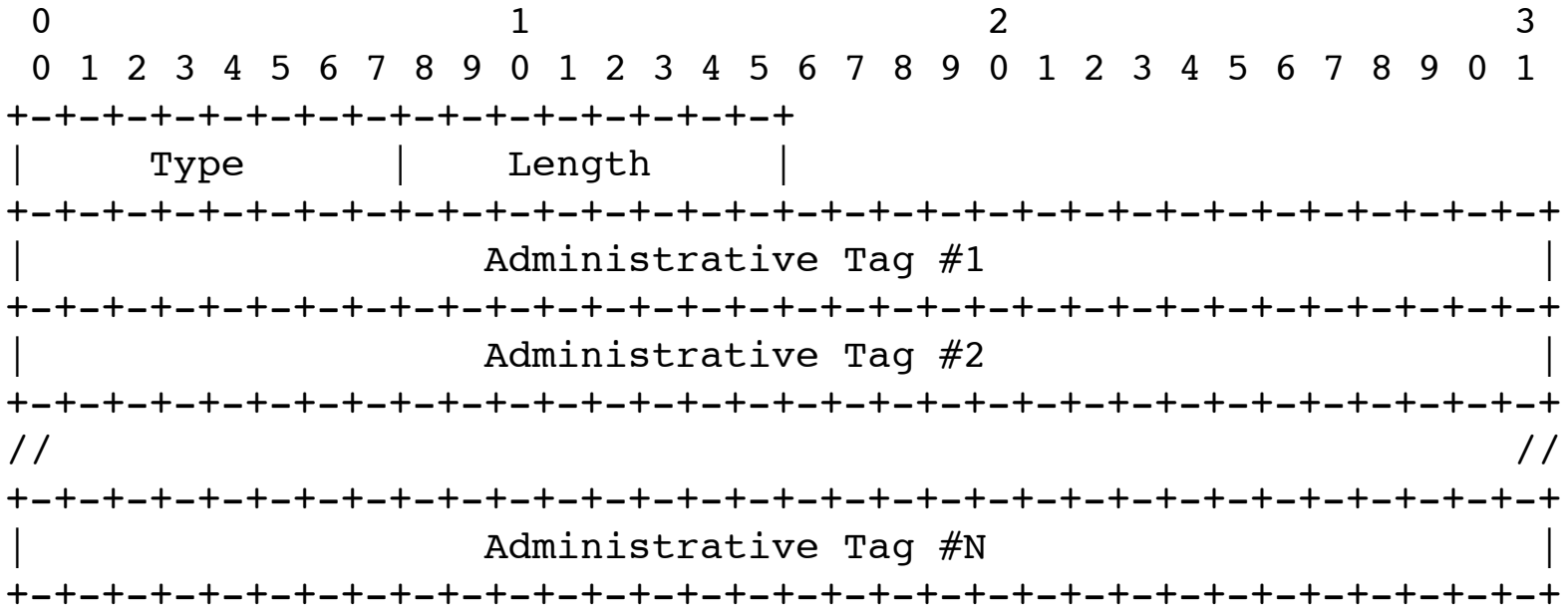
- Per-Node Admin Tags (contd.)
  - Facilitates logical grouping of nodes in network
    - One tag per group (per node characteristic type).
    - Multiple nodes exhibiting same characteristics
      - Tagged with same tag value.
    - Single node exhibiting multiple characteristics
      - Belongs to multiple groups.
      - Tagged with multiple tag values (one per group or node characteristics).

Prior Art

- Meaning of a node-admin tag is
  - Local to the network operator.
  - But unique across all the nodes in the same administrative domain.
  - Independent of the order the nodes are tagged with.
- Facilitate any routing applications, that
  - Require advertisement of any node characteristics within the network deployment.
  - No need to define well-known values for each new characteristic required to be advertised.

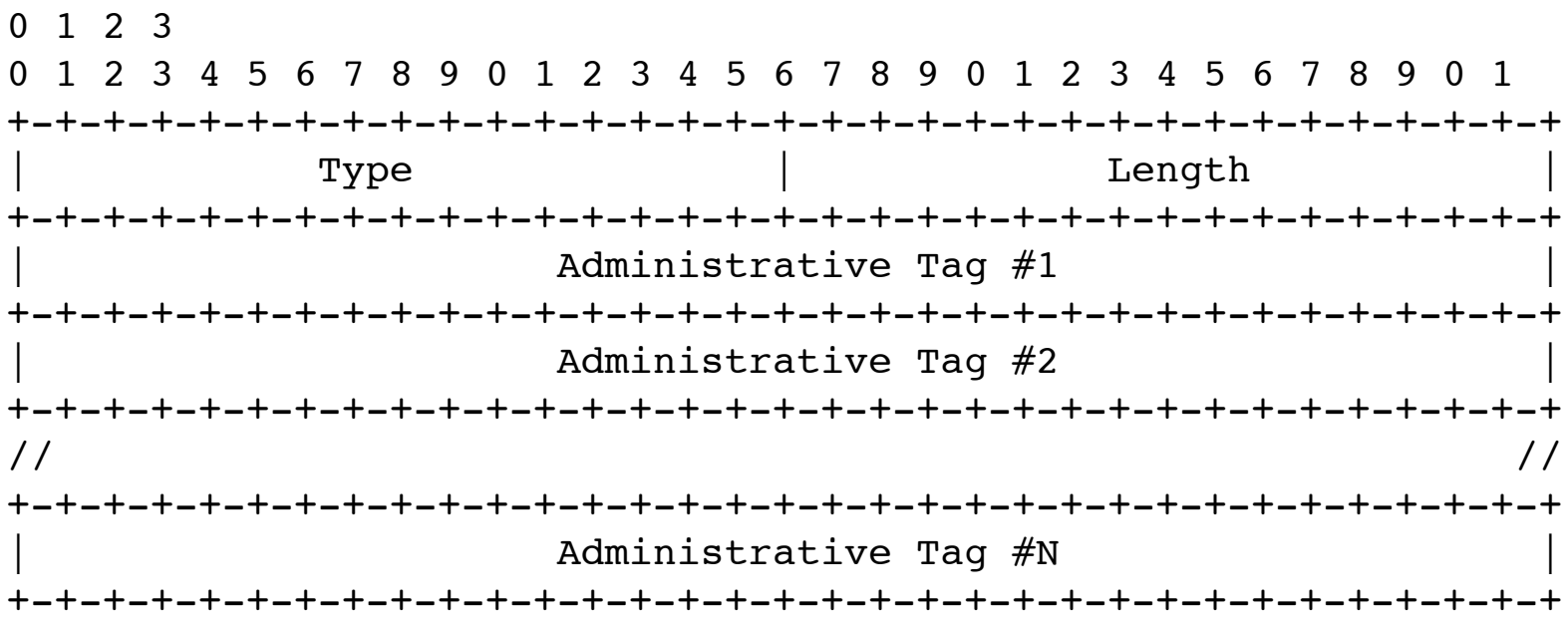
Prior Art

- ***New SubTLV of ISIS Router-Cap TLV #242 (RFC 4971)***
  - Unbound List of 32-Bit node colors (TLV-max-size costraints still applies)



Prior Art

- ***New TLV in OSPF Router Information LSA***
  - List of 32-bit admin tags (node colors).

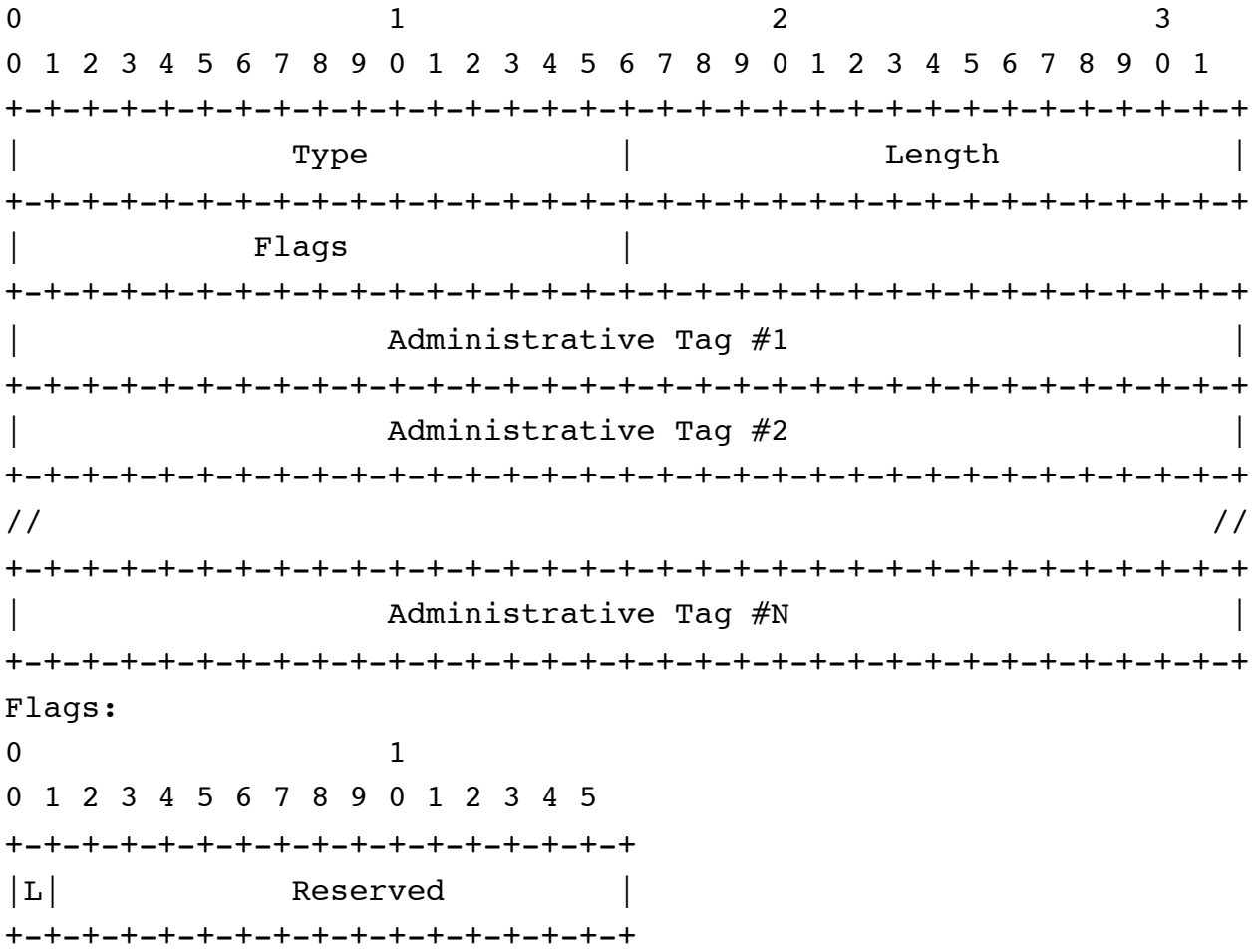


Draft Proposal

- BGP LS speaker(s) attached to each IGP area
  - Learn per-node admin tags from IGP link-state advertisements
  - Originate the same in corresponding BGP-LS advertisements.
    - As new Node-Admin Tag TLVs in appropriate instance of BGP-LS Node NLRI..
- BGP LS receivers
  - Learn all Node-Admin Tags
  - Associate them with the Area-Id(OSPF) or Level(IS-IS) attributes received in the same Node NLRI
- Facilitate any intra/inter-AS applications
  - Require grouping of nodes with similar characteristics/ capabilities
    - Both within and across AS boundaries.

Draft Proposal

- ***New Link State Node Admin Tag TLV in BGP-LS Node NLRI***



Implementations Guidelines

- While copying node admin tags from IGP link-state advertisements to corresponding BGP-LS Node NLRI
  - Separate ‘Node Admin Tag TLV’(s) with ‘L’ bit set to ‘1’
    - To carry all ‘level/area-wide’ node-admin tags.
  - Separate ‘Node Admin Tag TLV’(s) with ‘L’ bit reset to ‘0’
    - To carry all ‘domain-wide’ node-admin tags.
- .

Next Steps

- Questions and Comments?
- Working group review.
- Adoption as a WG draft.

BGP
Next-Hop Capabilities

draft-decraene-idr-next-hop-capability-00

Bruno Decraene Orange

Introduction: Recaps on RFC 5492 "Capabilities Advertisement with BGP-4"

- RFC 5492 (BGP capabilities) advertises capabilities of the BGP peer.
  - BGP session related
  - (BGP) control plane capabilities
- The BGP peer may not be the BGP Next Hop:
  - Route Reflection (iBGP)
  - Route Server (eBGP)
  - Next Hop unchanged (not setting Next Hop Self)
- Hence not a way to advertise capability of the BGP Next-Hop.
  - Forwarding planes capabilities

Next-Hop Capabilities encoding

1. New non-transitive BGP Attribute
2. Carries set of Next-Hop Capabilities
3. A Next-Hop Capability is encoded as a TLV

- In short:
  - same encoding as BGP capabilities but carried in a non transitive attribute

Next-Hop Capabilities operation

- We want the capability to be removed when the Next-Hop is changed.
- → For compliant peers:
  - if Next-Hop unchanged: attribute SHOULD be passed unchanged
  - if Next-Hop changed: attribute MUST be removed
    - new one may be attached to reflect capabilities of the new Next-Hop
- → For non compliant peers:
  - As the attribute is non-transitive attribute, it will be removed (as per RFC 4271).

Error handling

- Error condition: lengths mismatch
  - attribute length mismatch the sum of (capabilities lengths+2)
- Error handling: “attribute discard”
  - Assuming implementations do not allow changing route preference based on Next-Hop Capabilities…
  - Is this a safe assumption? Otherwise “treat as withdraw”?
    - or “attribute discard” on eBGP, and “treat as withdraw” in iBGP?

1 generic BGP Next-Hop Capabilities Attribute vs N BGP attributes (1 per application)

- Why defining a generic attribute?
- For IDR / implementations: doing the work once
  - single attribute used / single doc
  - single spec/coding/tests
- For the application: incremental deployment
  - non-transitive attribute required
  - a new non-transitive attribute would be unknown hence removed by existing implementations

First application proposed: "Entropy Label" Next-Hop Capability

- Capability sent if either:
  - BGP Next-Hop can process Entropy Label
  - BGP Next-Hop will perform a MPLS SWAP and not have to process Entropy Label
- When received, means: may send packets with a MPLS entropy label for this Next Hop/NLRI
- Based on the ELC BGP attribute defined in section 5.2 of [RFC6790] but then deprecated.
- Do we want to also advertise the Readable Label Depth?
  - number of labels readable by transit LSR for ECMP load-balancing hashing
    - as defined in draft-ietf-mpls-spring-entropy-label
  - could be RLD of NH or RLD from NH to egress (NLRI).
  - In the Value field? In a different NH Capability? (as RLD is independent of ELC)

Next

- Feedback & comments welcomed.

Thank you

BGP Model for Service Provider Networks

OpenConfig network operator working group
www.openconfig.net

draft-idr-shaikh-bgp-model-01

Anees Shaikh (Google), Kevin D'Souza (AT&T),
Rob Shakir (BT), Deepak Bansal (Microsoft)

IETF 92
IDR WG

Recap of OpenConfig BGP model

- OpenConfig operator working group -- models based on real usage
  - operators examining our own configurations and operational parameters
  - items that are widely available in major implementations

- Scope of the model
  - base BGP protocol configuration (global, neighbor, and peer-group templates)
  - support for multiple address families
  - policy (in conjunction with generic routing policy model)
  - operational state data

```
+--rw bgp!
     +--rw global
     |     ...
     +--rw neighbors
     |     ...
     +--rw peer-groups
           ...
```

Changes from -00 version

- Updates to BGP and policy model in draft-01
  - new operational state structure and data items
  - restructured AFI / SAFI configuration (generic and per AFI/SAFI)
  - separation of routing policy from BGP
  - new base protocol configuration items

- Several vendor implementations in progress
  - many changes in the model based on implementor feedback
  - related to model structure, location in the hierarchy, support for specific features, etc.

Coordination with other BGP models

Discussion with co-authors of draft-zhdankin-netmod-bgp-cfg

Summary of feedback / differences:

- policy
  - draft-zhdankin leaves routing policy largely out of scope due to perceived implementation differences
- operational state
  - draft-zhdankin leaves operational state out of scope for vendor-specific state data and statistics
- we have addressed a number of structural suggestions

Latest modules available in: `https://github.com/YangModels/yang/tree/master/experimental/openconfig`

Additional material

BGP AFI / SAFI configuration structure

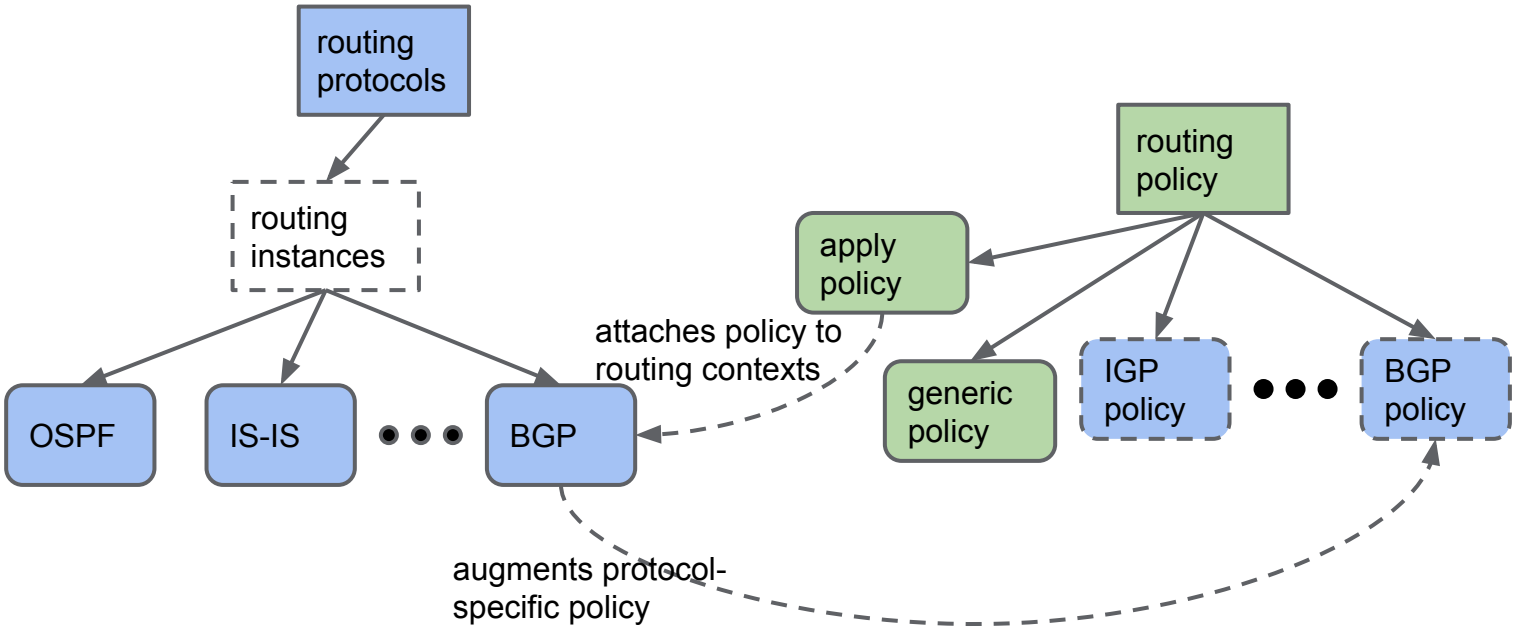
```
+--rw bgp!
   +--rw global
      +--rw afi-safi
         +--rw afi-safi* [afi-safi-name]
            +--rw afi-safi-name
            +--rw route-selection-options
            +--rw use-multiple-paths!
            +--rw config
            +--ro state
            +--rw apply-policy
            +--rw ipv4-unicast!
            +--rw ipv6-unicast!
            +--rw ipv4-labelled-unicast!
            +--rw ipv6-labelled-unicast!
            +--rw l3vpn-ipv4-unicast!
            +--rw l3vpn-ipv6-unicast!
            +--rw l3vpn-ipv4-multicast!
            +--rw l3vpn-ipv6-multicast!
            +--rw l2vpn-vpls!
            +--rw l2vpn-evpn!
```

supported AFI-SAFI types

global AFI-SAFI options

```
+--rw bgp!
   +--rw neighbors
      +--rw neighbor* [neighbor-address]
         +--rw afi-safi
            +--rw afi-safi* [afi-safi-name]
               +--rw afi-safi-name
                  +--rw route-selection-options
                  |  +--rw config
                  |     +--rw always-compare-med
                  |     +--rw ignore-as-path-length?
                  |     +--rw external-compare-router-id?
                  |     +--rw advertise-inactive-routes?
                  |     +--rw enable-aigp?
                  |     +--rw ignore-next-hop-igp-metric?
                  +--rw use-multiple-paths!
                  |  +--rw ebgp
                  |  |  +--rw config
                  |  |     +--rw allow-multiple-as?
                  |  |     +--rw maximum-paths?
                  |  +--rw ibgp
                  |     +--rw config
                  |        +--rw maximum-paths?   uint32
                  +--rw config
                     +--rw enabled?   boolean
```

Routing model structure decouples protocol and policy



Routing policy structure

generic routing policy model

```
+--rw routing-policy
      +--rw defined-sets!
      |  +--rw prefix-set* [prefix-set-name]
      |  |  +--rw prefix-set-name    string
      |  |  +--rw prefix*
      |  |        ...
      |  +--rw neighbor-set* [neighbor-set-name]
      |  |  +--rw neighbor-set-name    string
      |  |  +--rw neighbor* [address]
      |  |        ...
      |  +--rw tag-set* [tag-set-name]
      |     +--rw tag-set-name    string
      |     +--rw tag* [value]
      |           ...
      +--rw policy-definition* [name]
         +--rw name          string
         +--rw statement* [name]
            +--rw name           string
            +--rw conditions!
            |     ...
            +--rw actions!
                  ...
```

augmented with BGP-specific defined sets

```
+--rw routing-policy
      +--rw defined-sets!
      |  +--rw prefix-set* [prefix-set-name]
      |  |  +--rw prefix-set-name    string
      |  |  +--rw prefix*
      |  |        ...
      |  +--rw neighbor-set* [neighbor-set-name]
      |  |  +--rw neighbor-set-name    string
      |  |  +--rw neighbor* [address]
      |  |        ...
      |  +--rw tag-set* [tag-set-name]
      |  |  +--rw tag-set-name    string
      |  |  +--rw tag* [value]
      |  |        ...
      |  +--rw bgp-pol:bgp-defined-sets
      |     +--rw bgp-pol:community-set*
      |     |     ...
      |     +--rw bgp-pol:ext-community-set*
      |     |     ...
      |     +--rw bgp-pol:as-path-set*
      |           ...
```

BGP model overall structure

```
+--rw bgp!
   +--rw global
   |  +-- (global-configuration-options)
   +--rw neighbors
   |  +--rw neighbor* [neighbor-address]
   |  +-- (neighbor-configuration-options)
   +--rw peer-groups
      +--rw peer-group* [peer-group-name]
      +-- (neighbor-configuration-options)
```

- hierarchical configuration with overrides as in most existing implementations
- primary configuration at neighbor level with peer group templates
- policies may be applied at multiple levels or specific address families

BGP neighbor configuration items

```
+--rw bgp!
      +--rw neighbor* [neighbor-address]
         +--rw neighbor-address
         +--rw peer-as
         +--rw description?
         +--rw graceful-restart!
         +--rw apply-policy
         +--rw afi-safi* [afi-safi-name]
         +--rw auth-password?
         +--rw peer-type?
         +--rw timers
         +--rw ebgp-multihop
         +--rw route-reflector
         +--rw remove-private-as?
         +--rw bgp-logging-options
         +--rw transport-options
         +--rw local-address?
         +--rw route-flap-damping?
         +--rw send-community?
         +--rw error-handling
         +--rw as-path-options
         +--rw add-paths!
```

Summary

- OpenConfig BGP model updates have been reviewed by a number of operators
- Policy is now decoupled into OpenConfig generic routing policy model
- Model is under review by several vendors -- early feedback on implementation readiness has been positive
- Updated model addresses many differences between OpenConfig and draft-zhdankin BGP models
- Models available in public YangModels repository

`https://github.com/YangModels/yang/tree/master/experimental/openconfig`

Yang Data Model for BGP

draft-zhdankin-idr-bgp-cfg-00

Alexsandr Zhdankin, Keyur Patel, Alexander Clemm, Sue Hares, Mahesh Jethanandani, Xu Feng

IETF 92, March 2015, Dallas, US

Update

- Draft Name changed from *draft-zhdankin-netmod-bgp-cfg-01 to draft-zhdankin-idr-bgp-cfg-00*
  - Work for the Protocol Yang models to be done in the protocol WG
- Added Co-authors
  - Sue Hares (Huawei)
  - Mahesh Jethanandani (Ciena)
  - Xu Feng (Ericsson)
- Incorporated Comments from Adam Simpson (ALU) & Gunter Vandevelde

Update

- Update to BGP Model
  - Added BGP Protocol version
  - Added BGP Neighbor Groups (Peer Groups)
  - Added Neighbor based Transport Parameters
  - Added BGP Route Flap Dampening Support

BGP Yang Model

- module: bgp

```
+--rw bgp-routing
|  +--rw bgp-router
|  |  +--rw bgp-version?           string
|  |  +--rw local-as-number?       uint32
|  |  +--rw local-as-identifier?   inet:ip-address
|  |  +--rw rpki-config
|  |  |  .....
|  |  +--rw af-configuration
|  |     .....
|  +--rw bgp-neighbors
|     +--rw bgp-neighbor-af
|     |  ....
|  +--rw bgp-neighbors-groups
|     |  ....
```

Major Differences

- Neighbor list is a list inside a container in bgp-yang versus a list in oc-bgp model
- Neighbor group is a container with Nbrs having leaf reference in bgp-yang versus having an individual nbr for inheritance in oc-bgp
- Global Address-familes are containers in bgp-yang versus lists in oc-bgp model
- Import/export (partial L3VPN semantics) present in oc-bgp model

Major Differences

- Important to get these resolved to come to a single model at a high level
  - Versioning should help navigate the differences
- Worked with oc-bgp draft authors: Rob & Aneesh to resolve these differences
  - **New oc-bgp model has incorporated the comments and resolved most of the differences**

Moving Forward – Next Revision

- Plan on replacing enums with identities
  - Identities help avoid model revisions
- Use of deviations to harmonize command parameters across vendor implementations
- Use of if-feature for:
  - Vendor specific features
  - BGP extensions
- Oper State Model

Suggest towards a common draft and model to progress work forward

Major Differences

- RFCs supported by bgp-yang and oc-bgp

  RFC4271 BGP Protocol Specification **[bgp-yang & oc-bgp]**

  RFC1997 BGP Community Attribute **[bgp-yang & oc-bgp]**

  RFC4456 BGP Route Reflection **[bgp-yang & oc-bgp]**

  RFC4760 BGP Multiprotocol Specification **[bgp-yang & oc-bgp]**

  RFC3065 Autonomous System Confederations for BGP **[bgp-yang & oc-bgp]**

  RFC2439 BGP Route Flap Dampening **[bgp-yang & oc-bgp]**

  RFC 4724 BGP Graceful Restart **[bgp-yang & oc-bgp]**

  RFC 6811 BGP Origin Validation **[bgp-yang]**

Questions?

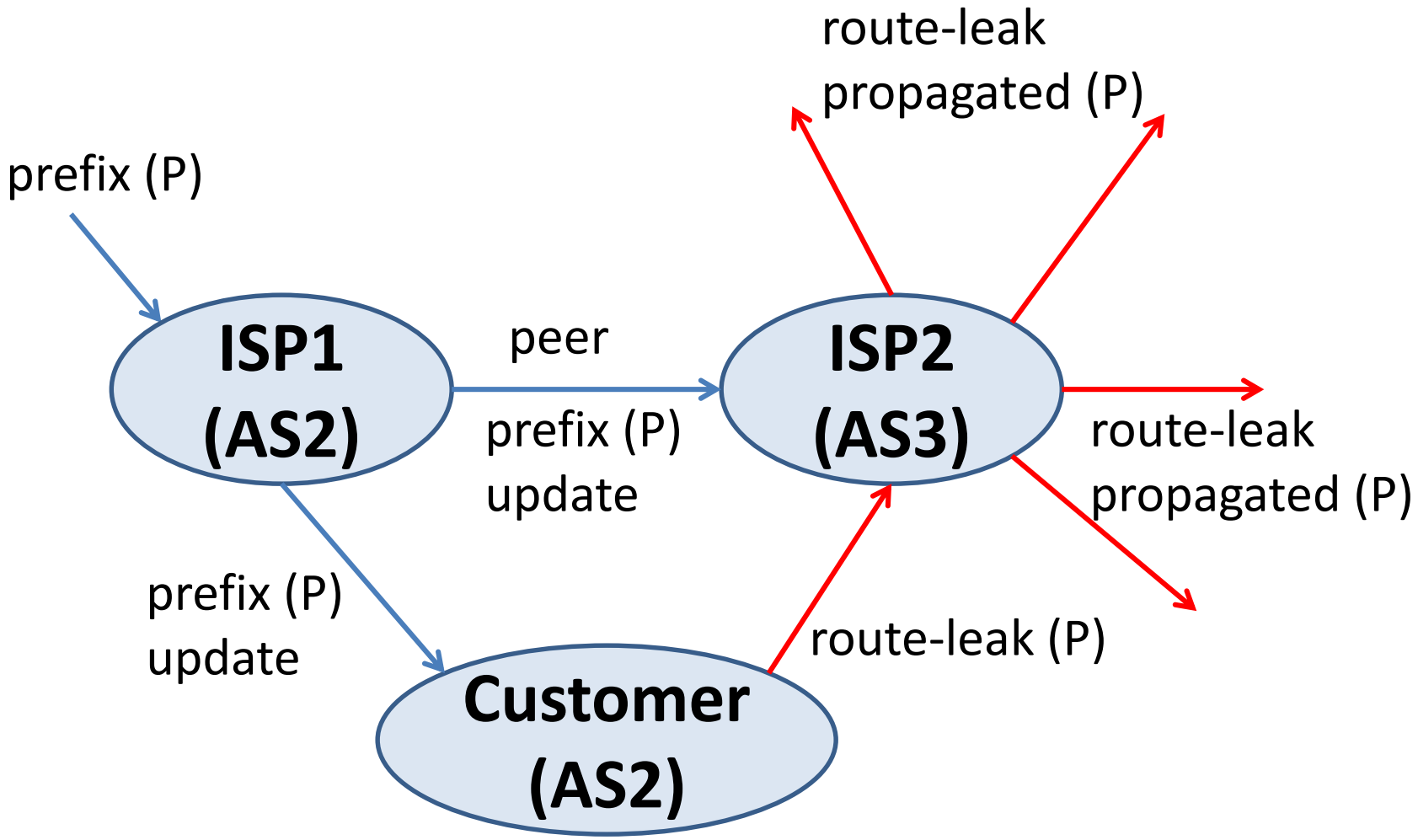
Methods for Detection and Mitigation of BGP Route Leaks

draft-sriram-idr-route-leak-detection-mitigation-00

K. Sriram and D. Montgomery
NIST
IDR WG Meeting, IETF 92, Dallas, Texas
March 24, 2015

Acknowledgements: The authors would like to thank Danny McPherson, Eric Osterweil, Jeff Haas, Warren Kumari, Jared Mauch, Amogh Dhamdhere, Andrei Robachevsky, Brian Dickson, Randy Bush, Chris Morrow, and Sandy Murphy for comments and suggestions.

Illustration of Basic Notion of a Route Leak



In general, ISPs prefer customer route announcements over those from others.

Anatomy of a Route Leak: Seven Types

Type 1: Type 1: U-Turn with Full Prefix

Type 2: U-Turn with More Specific Prefix

Type 3: Prefix Reorigination with Data Path to Legitimate Origin

Type 4: Leak of Internal Prefixes and Accidental Deaggregation

Type 5: Lateral ISP-ISP-ISP Leak

Type 6: Leak of Provider Prefixes to Peer

Type 7: Leak of Peer Prefixes to Provider

Details and example incidents provided in:
draft-ietf-grow-route-leak-problem-definition-01

Route Leak Detection/Mitigation in Origin Validation and BGPSEC

Type of Route Leak	Detection Coverage
Type 1: U-Turn with Full Prefix	None
Type 2: U-Turn with More Specific Prefix	Origin Validation (partial); BGPSEC (100% detection)
Type 3: Prefix Reorigination with Data Path to Legitimate Origin	Origin Validation (100% detection); BGPSEC does not detect*
Type 4: Leak of Internal Prefixes and Accidental Deaggregation	Origin Validation (partial); BGPSEC does not detect*
Type 5: Lateral ISP-ISP-ISP Leak	None
Type 6: Leak of Provider Prefixes to Peer	None
Type 7: Leak of Peer Prefixes to Provider	None

*BGPSEC protocol performs path validation only, and does not include OV (spec version 11)

Begin Sender Specification

(Simple Enhancement to Existing BGP or BGPSEC)

Route Leak Protection (RLP) Field Encoding by Sending Router (Method 1)

- RLP is proposed to be a 2-bit field set by each AS along the path
- Can be carried in a Transitive Community attribute in BGP or in the Flags field in BGPSEC (TBD)
- The RLP field value SHOULD be set to one of two values as follows:
  - 00: This is the default value (i.e. "nothing specified"),
  - 01: This is the 'Do not Propagate Up' indication; sender indicating that the prefix-update SHOULD NOT be subsequently forwarded 'Up‘ towards a provider AS,
  - 10 and 11 values are for possible future use.

Route Leak Protection (RLP) Field Encoding by Sending Router (Method 2)

Only the following is different w.r.t. Method 1:

- The RLP field value SHOULD be set to one of two values as follows:
  - 00: This is the default value (i.e. "nothing specified"),
  - 01: “Do not Propagate Up” indication
  - 10: “Propagate to Customers Only” indication
  - 11: “Do not Propagate” (i.e. NO_EXPORT)

Agreeing on the semantics of these indications is important.

End of Sender Specification

Sending Router's Intent

- Note: There is no explicit disclosure about the nature of a peering relationship.
- (In Method 1) By setting RLP indication to 01, merely asserting that this prefix-update that I've forwarded to my neighbor SHOULD not be propagated 'Up' (i.e. on a c2p link) by said neighbor or any subsequent AS in the path of update propagation.

Recommendation for Receiver Action for Detection of Route Leaks of Types 1, 2 and 7

(When Sender is using Method 1)

Receiving router SHOULD mark an update a Route-Leak if ALL of the following conditions hold true:

a) The update is received from a customer AS.

b) The update has the RLP field set to '01' (i.e. 'Do not Propagate Up') indication for one or more hops (excluding the most recent) in the AS path.

Note: Reason for “excluding the most recent” – an ISP should look at RLP values set by ASes preceding the customer AS in order to ascertain a leak .

Recommendation for Receiver Action for Detection of Route Leaks of Types 5 and 6

(When Sender is using Method 1)

Receiving router SHOULD mark an update a Route-Leak if ALL of the following conditions hold true:

a) The update is received from a peer AS.

b) The update has the RLP field set to '01' indication for one or more hops (excluding the most recent) in the AS path.

Note: I this case, the RLP indication of ’01’ is more strictly interpreted to mean that the update should not be propagated on a lateral peer link either.

An Example Receiver Action for Mitigation of Route Leaks

- If an update from a customer AS or a peer AS is detected and marked as a “Route-Leak”, then the receiving router SHOULD prefer an unmarked update from another neighbor AS, if available.

Path for Success

- Mid and large size ISPs can participate early, and be the detection/mitigation points for route leaks.
- More the ISPs that adopt, greater the success (benefits accrue incrementally).

Note: In a case like that of Moratel's leak (in November 2012) of Google's prefixes, the attack is mitigated if Google would set its RLP field value to 01 in its prefix update announcement to Moratel, and PCCW would in turn use the receiver action recommended on Slide 11 to identify the update from Moratel as a Route Leak.

Summary and Conclusion

- Identified categories of route leaks
- Some of these are already mitigated in OV or BGPSEC
- Presented an enhancement of BGP that detects and mitigates all route leaks (when combined with Origin Validation)
- The RLP field can be carried in a Transitive Community Attribute or in BGPSEC Flags field
- The RLP field may need to be protected in order to prevent tampering and/or malicious route leaks

Service-Oriented MPLS Path Programming (SoMPP)

draft-li-spring-mpls-path-programming-01/

draft-li-idr-mpls-path-programming-01

Zhenbin Li, Shunwan Zhuang

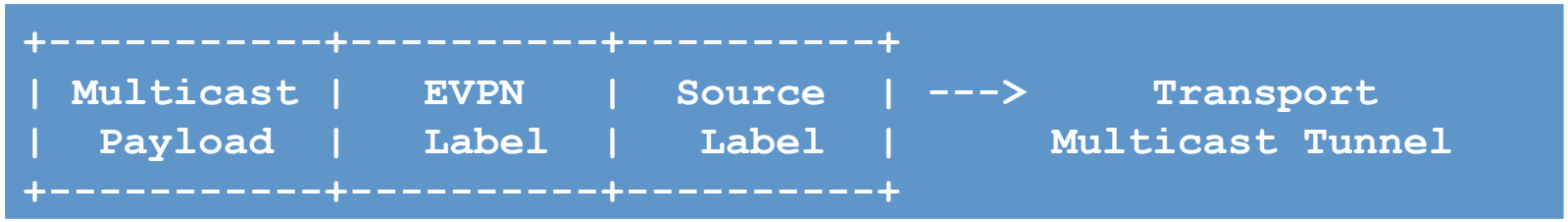
Huawei Technologies

IETF 92, Dallas, TX USA

Introduction of SoMPP

- Service-oriented MPLS programming proposed by [I-D.li-spring-mpls-path-programming] is to provide customized service process based on flexible label combinations.
- Use cases for unicast service MPLS path programming is shown as follows:

- Use cases for multicast service MPLS path programming is shown as follows (using BUM in EVPN as the example) :



Architecture of SoMPP

- 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.

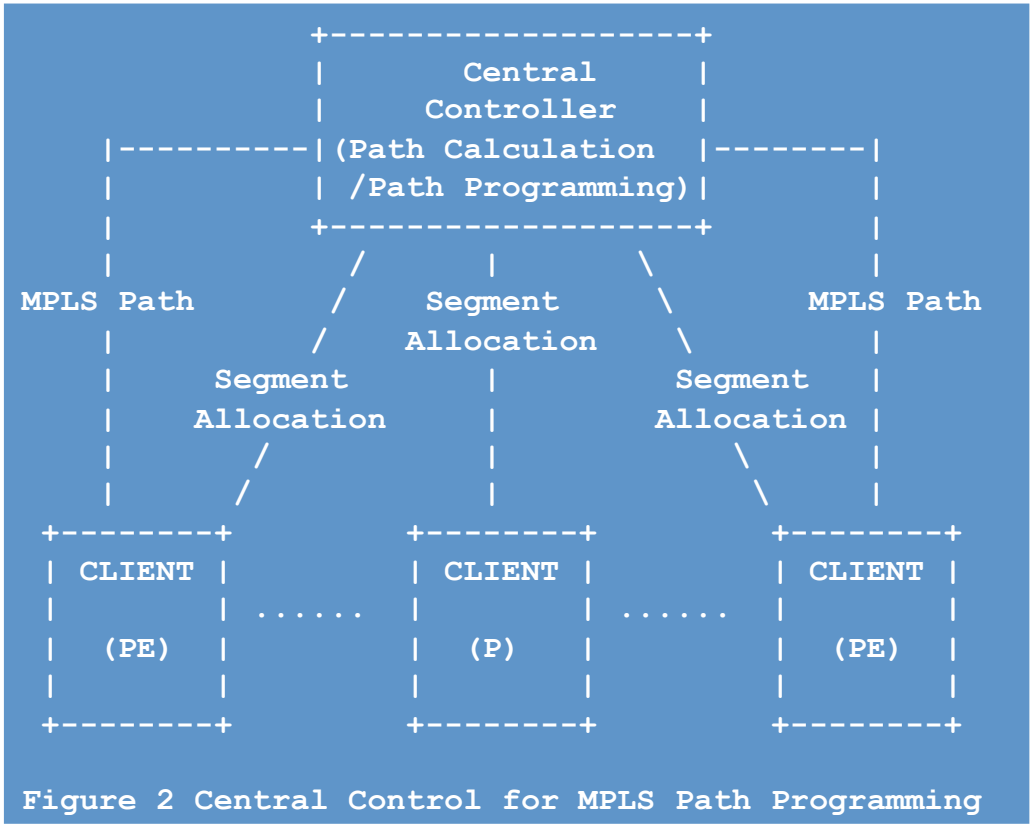


Figure 2 Central Control for MPLS Path Programming

Updates of draft-li-spring-mpls-path-programming-01

- This document defines the concept of MPLS path programming, then proposes use cases, architecture and protocol extension requirements in the service layer for the SPRING architecture.
- Updates:
  - Enhanced requirements definition.

    REQ 05: BGP extensions SHOULD be introduced to specify the end-points to accept the prefix advertised by the central controller.

    REQ 06: BGP extensions SHOULD be introduced to specify the priority for the prefix with attributes of MPLS path programming advertised by the central controller.

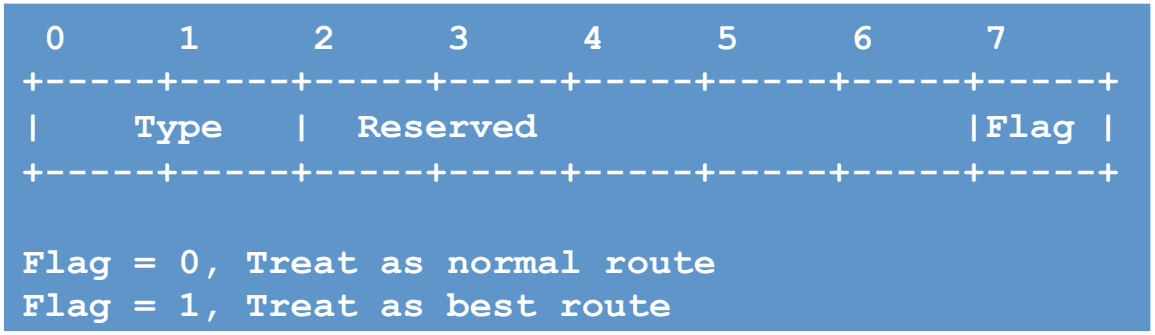
    REQ 07: When route selection is done in the client node, the path advertised by the central controller SHOULD has higher priority than the path calculated on the client's own.

Updates of draft-li-idr-mpls-path-programming-01

- This document defines BGP extensions to support service-oriented MPLS path programming.
- Updates
  - Enhanced BGP Extensions of SoMPP
    - Extended Label attribute
    - Extended Unicast Tunnel Attributes
    - Extended PMSI Tunnel Attribute
    - Route Flag Extended Community
    - Destination Node Attribute

Route Flag Extended Community

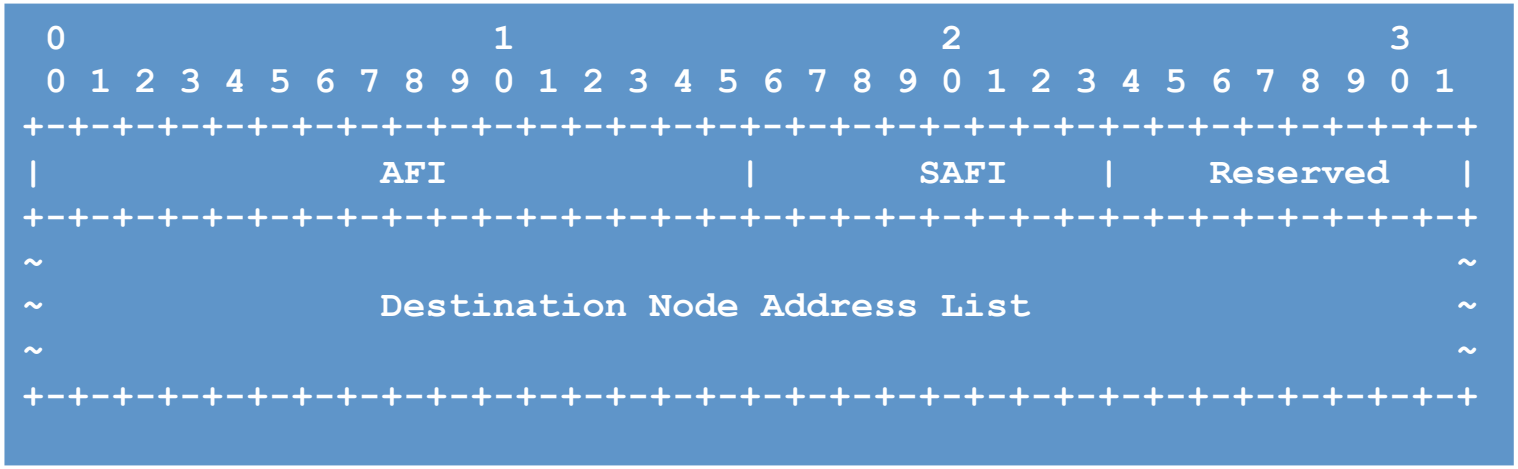
- The Route Flag Extended Community is used to carry the flag appointed by a BGP route server (e.g., a central controller):



- ✓ When a router receives a BGP route with a Route Flag Extended Community and the Flag set to "1", it SHOULD use the route as the best route when select the route from multiple routes for a specific prefix.

Destination Node Attribute

- A new optional, non-transitive BGP attribute called as the "Destination Node attribute”, can be applied to any address family:



- ✓ The Destination Node attribute is used to carry a list of node addresses, which are intended to be used to determine the nodes where the route with such attribute SHOULD be considered.
- ✓ If a node receives a BGP route with a Destination Node attribute, it MUST check the node address list. If one address of the list belongs to this node, the route MUST be used in this node. Otherwise the route MUST be ignored silently.

Prototype Work: Intra VPN Traffic Steering (1)

- Intra VPN Traffic Steering
  - PCE-Initiated LSP is created in the ingress nodes.
  - BGP extensions is to change the tunnel for the VPN routes in the ingress node.
- All traffic steering work is initiated from the controller which can save huge configuration work of the traditional traffic steering method.

Prototype Work: Intra VPN Traffic Steering (2)

Steps:

T1: VPN uses Tunnel 1 (PCE-initiated LSP)

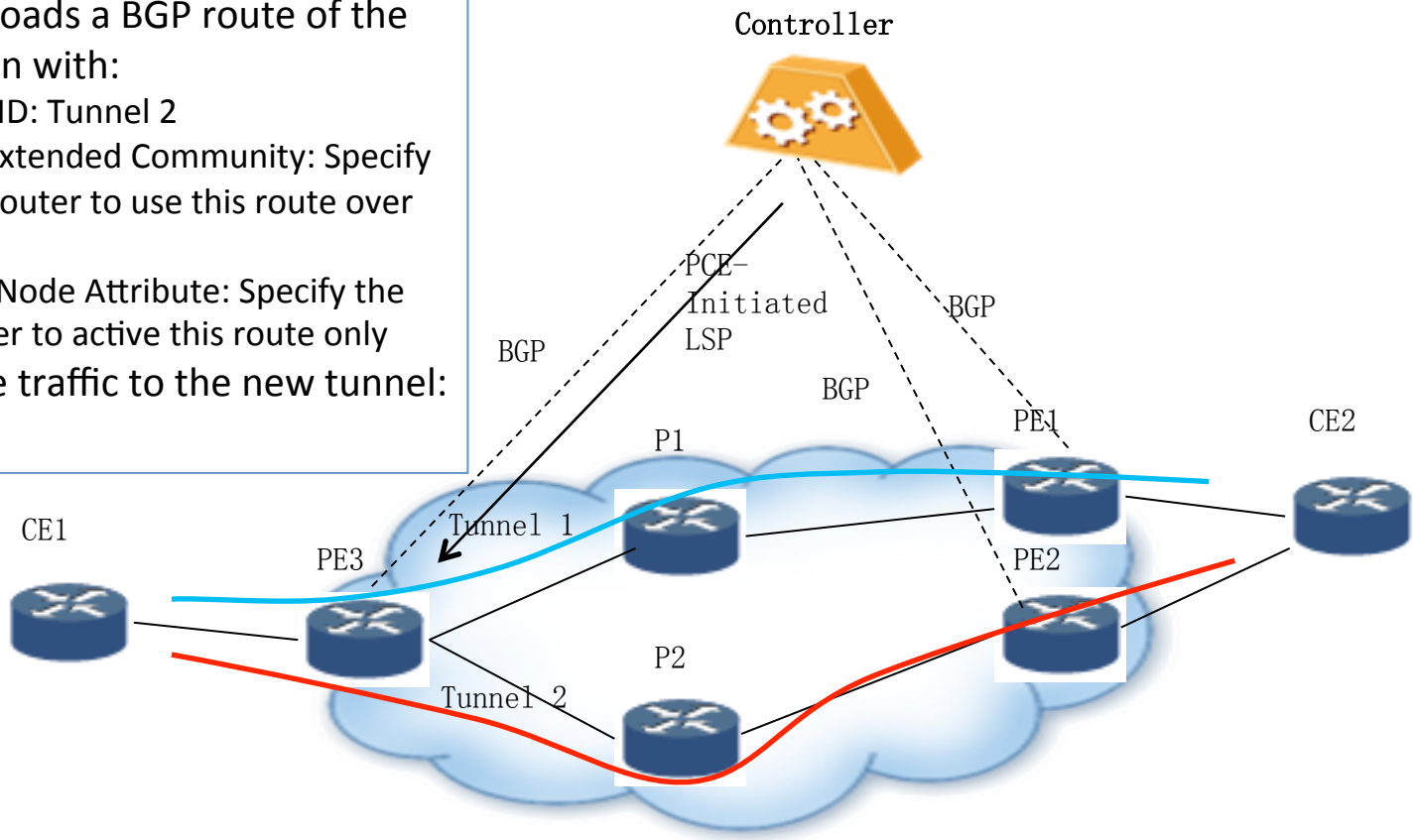
T2: Tunnel 1 is overloaded

T3: Controller creates Tunnel 2 (PCE-initiated LSP)

T4: Controller downloads a BGP route of the traffic destination with:

1) New tunnel ID: Tunnel 2
2) Route Flag Extended Community: Specify the ingress router to use this route over other path
3) Destination Node Attribute: Specify the ingress router to active this route only

T5: VPN switches the traffic to the new tunnel: Tunnel 2



Next Step

- Seek comments and feedbacks
- Revise the draft

BGP-LU for HSDN Label Distribution

draft-fang-idr-bgplu-for-hsdn-00

Luyuan Fang, lufang@microsoft.com

Chandra Ramachandran, csekar@juniper.net

Fabio Chiussi, fchiussi@cisco.com

Yakov Rekhter

IDR meeting, IETF 92

March 24, 2014, Dallas, TX

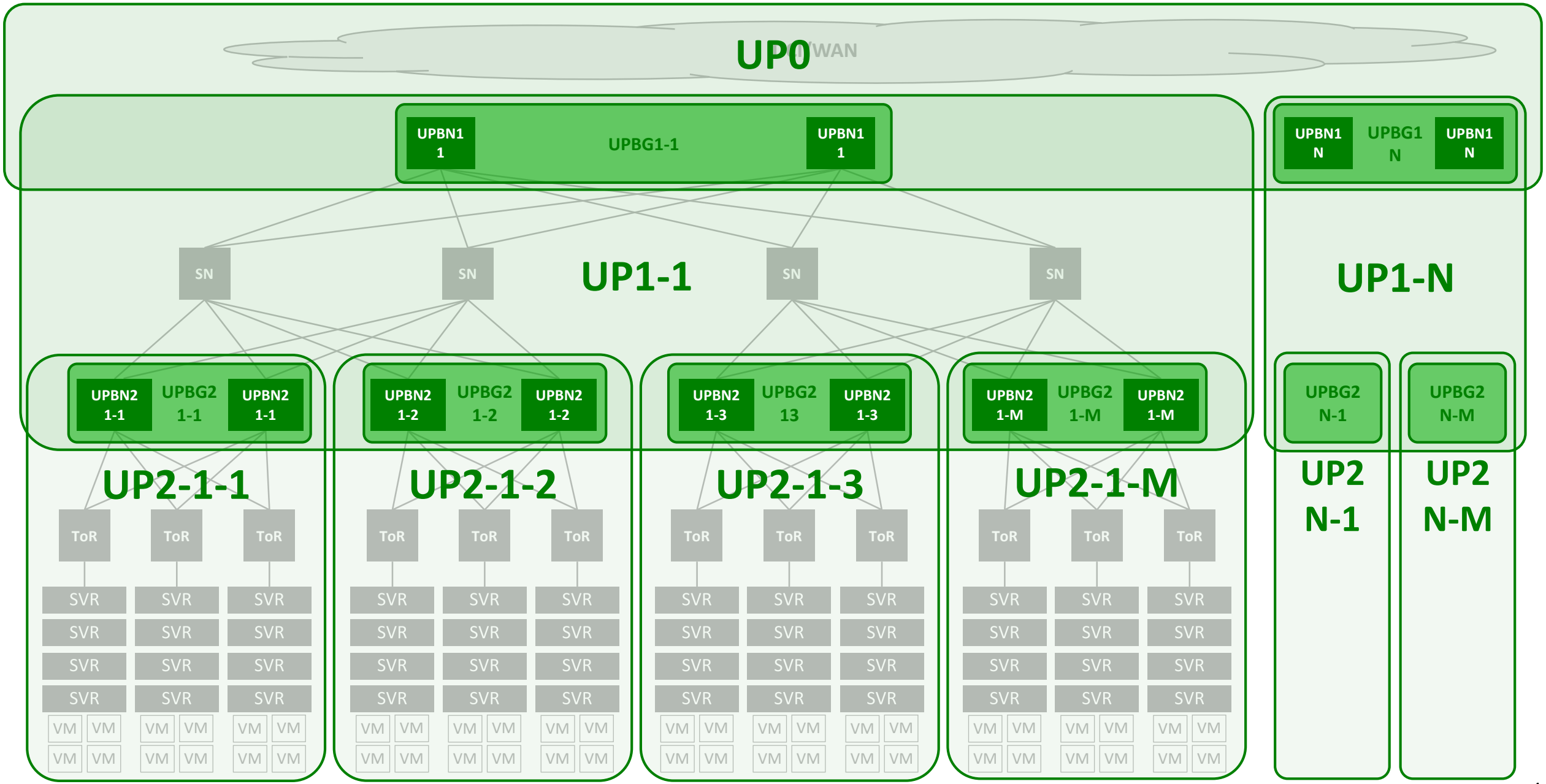
Purpose of the draft

Use BGP Labeled Unicast (BGP-LU), with modified BGP Route Reflector (RR) operation, as one of the options, for label distribution in the Hierarchical SDN (HSDN) (draft-fang-mpls-hsdn-for-hsdc-01) control plane (hybrid approach) for the hyper-scale Data Center (DC) and cloud networks.

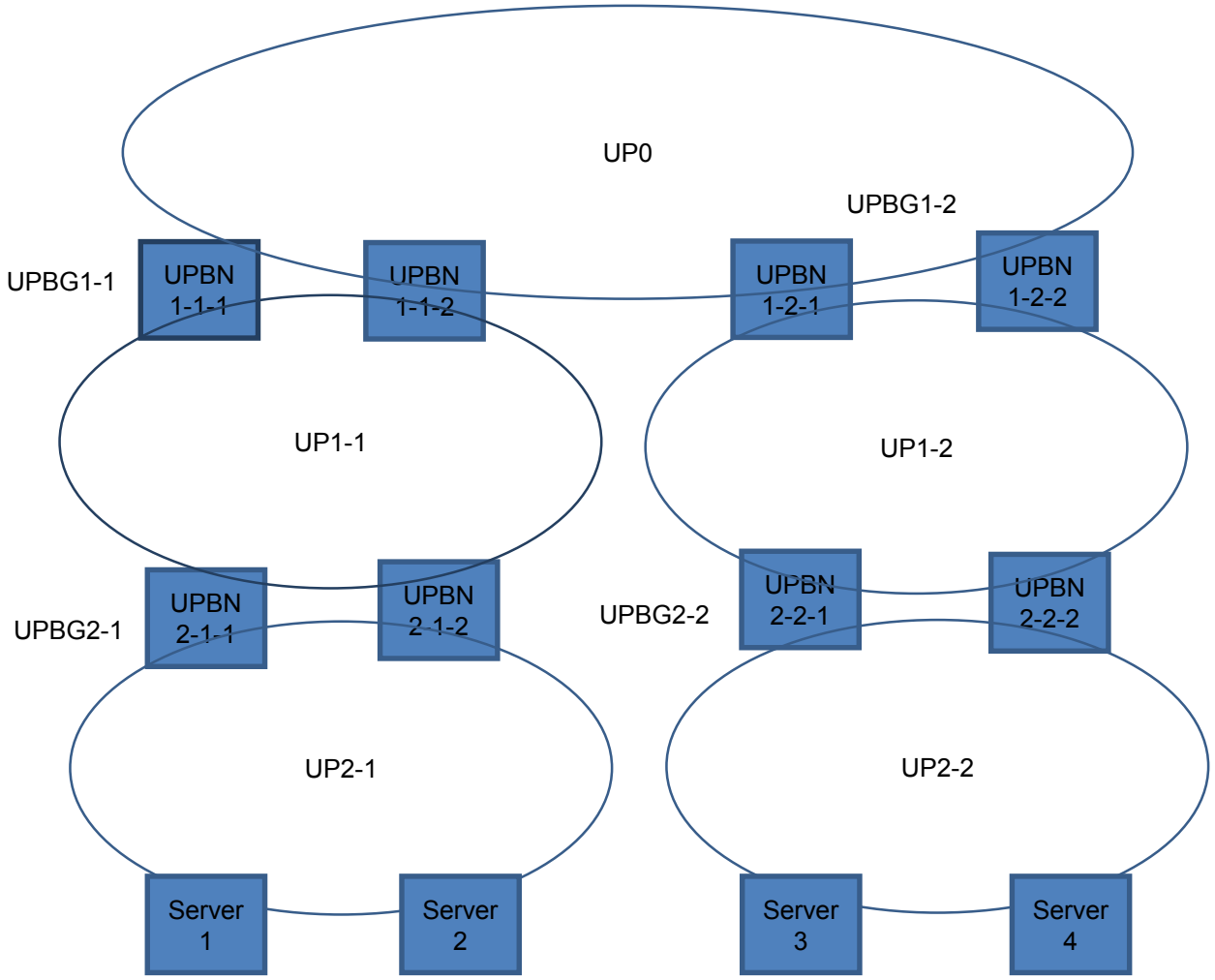
Terminoglogy

- UP: Underlay Partition.
- UPBN: Underlay Partition Border Node.
- UNBG: Underlay Partition Border Group.
- RR: BGP Route Reflector.
- BGP Peer Group: Collection of BGP peers for which a set of policies are applied on a BGP speaker.
- Label Mapping Server: A node present in each Underlay Partition that allocates labels for destinations in the partition.
- Label Mapping RR (LM-RR): A modified or customized BGP RR that uses BGP-LU to advertise label bindings for destinations in UP. LM-RR is an implementation of Label Mapping Server that uses BGP-LU to advertise the labels for partition destinations.
- Peer Community: An IP based extended community carried in BGP update that represents UPBG of a partition.
- Route Resolver: A single or a collection of entities that provides the MPLS label stack to reach a destination underlay end device.

Reference Model of HSDN: Hiarchical Partition with UPBNs and UPBGs

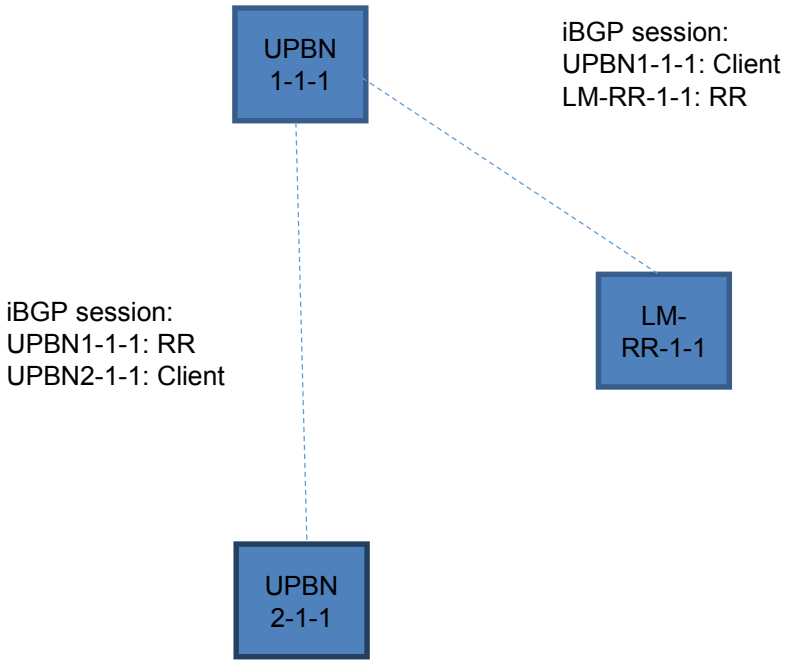


HSDN Example Topolgy



UPs running IGP

iBGP Sessions in UP1-1

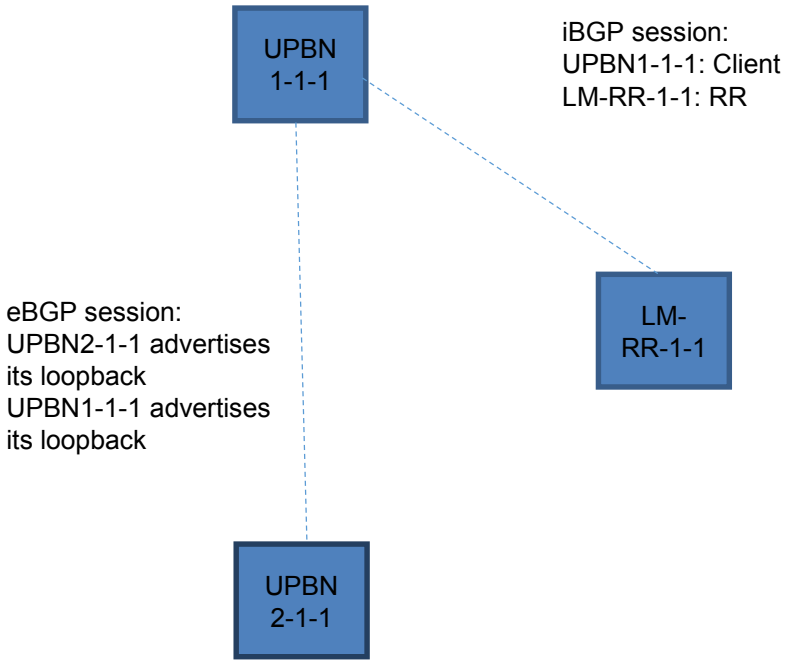


Note:

- On UPBN1-1-1, iBGP session with UPBN2-1-1 and iBGP session with LM-RR-1-1 belong to different peer-group
- On UPBN1-1-1, IGP for UP1-1 and UP0 are different instances and routes are not leaked between the instances

UPs running eBGP

eBGP Sessions in UP1-1



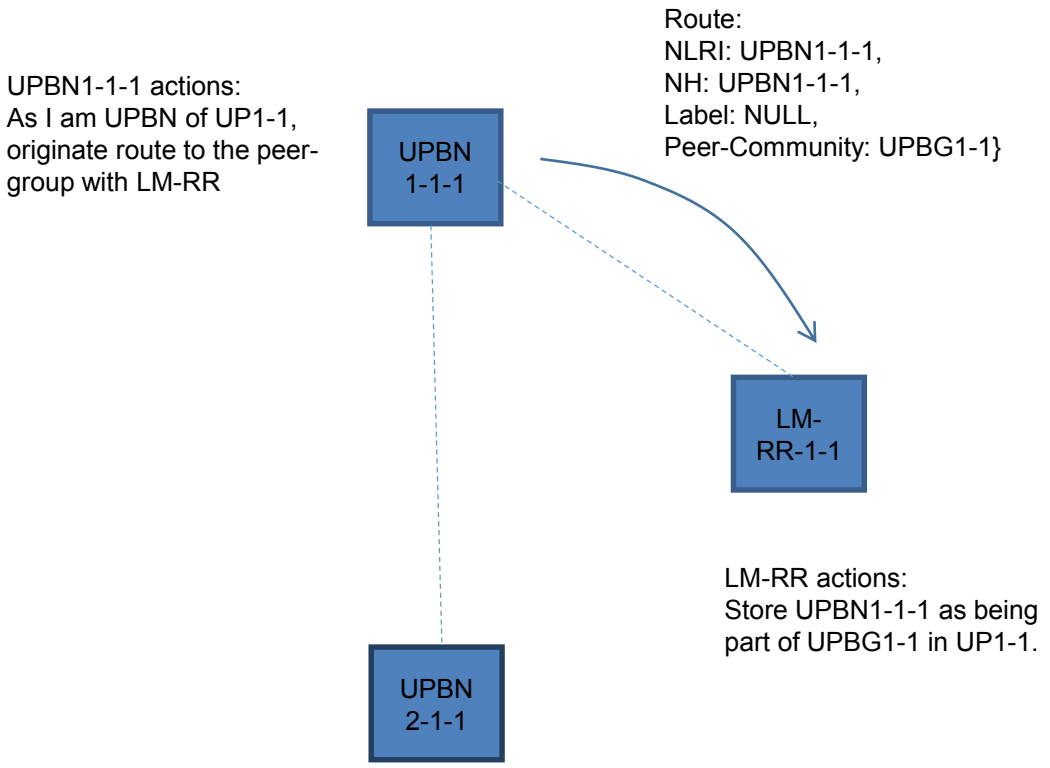
Note:

- UPBN1-1-1 and UPBN2-1-1 are in different AS
- If UPBN1-1-1 and UPBN2-1-1 are not directly connected, then there will be eBGP peerings such that each intermediate node in UP belongs to a different AS

BGP-LU procedures:

- The procedures described in the following slides are applicable for UPs running IGP or eBGP.

Step 1: UPBN1-1-1 originates self route



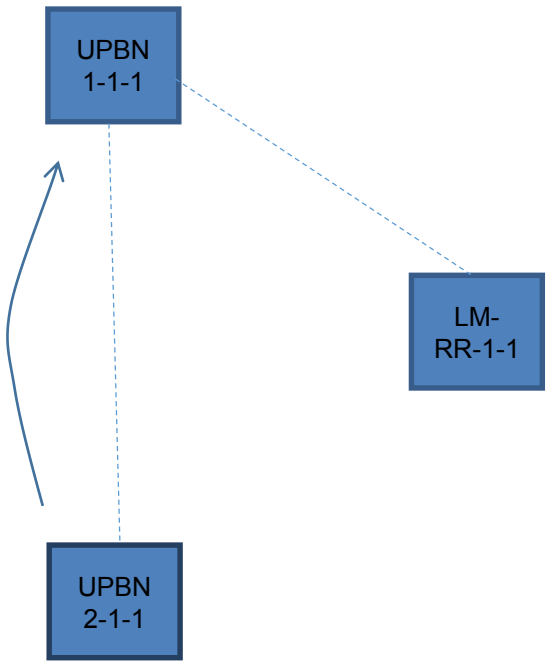
Note:

- Peer-Community is a new extended community
- Each UPBG will have a unique Peer-Community value
- LM-RR may act as “vanilla” BGP-RR for labeled-unicast routes

Step 2: UPBN2-1-1 originates self route

UPBN1-1-1 actions:
As I am UPBN of UP1-1 and I have received route from peer-group of UP1-1, then do not perform normal BGP-LU actions

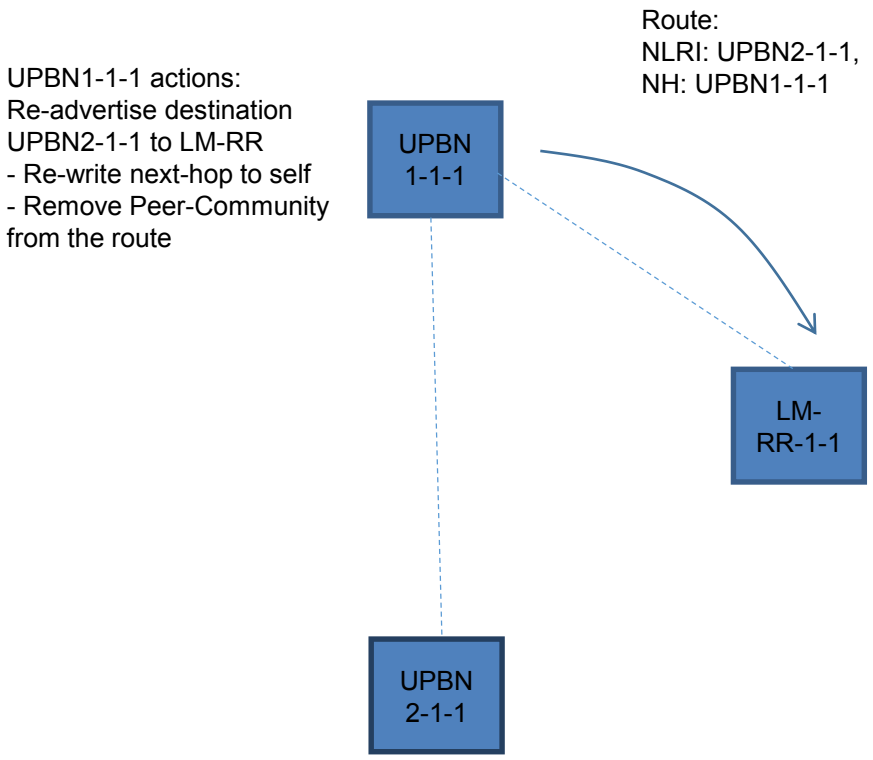
Route:
NLRI: UPBN2-1-1,
NH: UPBN2-1-1,
Label: NULL,
Peer-Community: UPBG2-1}



Note:

- UPBN2-1-2 will also advertise labeled-unicast route for itself to UPBG1-1 with Peer-Community UPBG2-1
- UPBN2-1-1 will also have iBGP session with UPBN1-1-2
- UPBNs of UP1-1 are RRs and destinations of UP1-1 are clients.

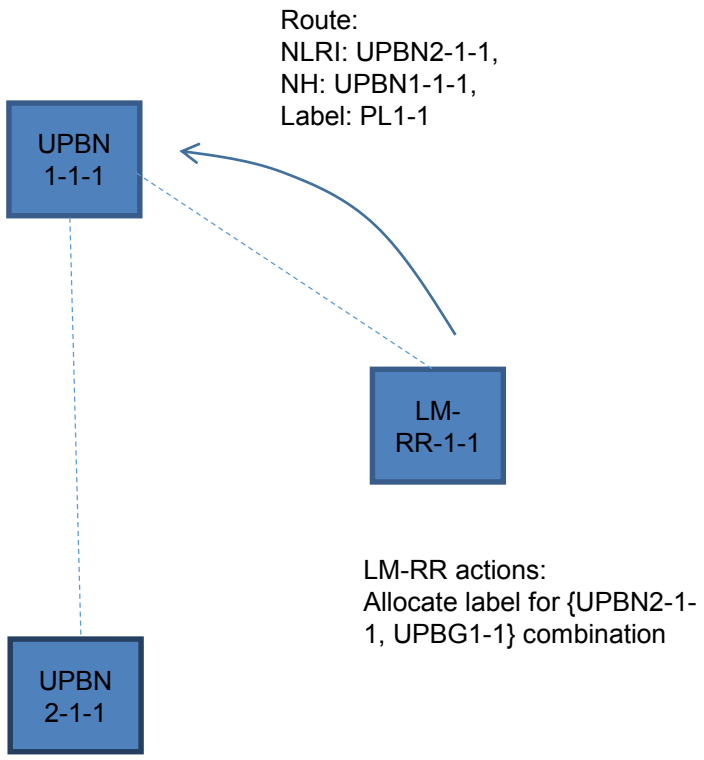
Step 3: UPBN1-1-1 re-advertises to LM-RR



Note:

- UPBN1-1-1 converts labeled-unicast route to inet family
- One can think of this as special UPBN action where unlike normal RFC3107 receiver, UPBN cannot allocate a label by itself and so it internally copies such “un-allocated” destinations to a special TIB called “LM-TIB”
- Any route in “LM-TIB” leads to route origination to iBGP peer-group with LM

Step 4: LM-RR allocates & advertises label

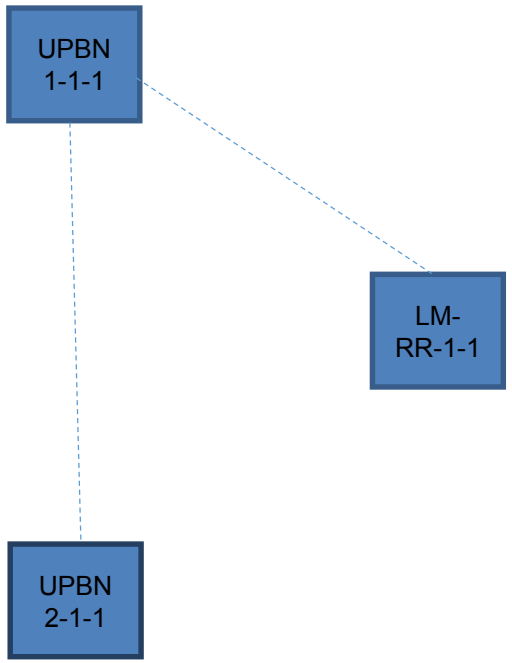


Note:

- LM-RR converts inet route to labeled-unicast family
- One can think of this as special LM-RR action where unlike normal RR, LM-RR cannot reflect “vanilla” IP routes
- LM-RR can be thought of as originating labeled-unicast route for each inet destination learnt where the label is allocated for each destination per UPBG (present in “LM-TIB”

Step 5: UPBN1-1-1 installs label in LFIB

UPBN1-1-1 actions:
- Install PL1-1 in LFIB
- Resolve UPBN2-1-1 using any LSP within UP1-1 to the destination



Note:
- One can think of this as special UPBN action where it internally copies the labeled-unicast route from LM-RR to “LM-TIB” making the destination UPBN2-1-1 as “allocated”
- Any “allocated” route in “LM-TIB” leads installation of label in LFIB

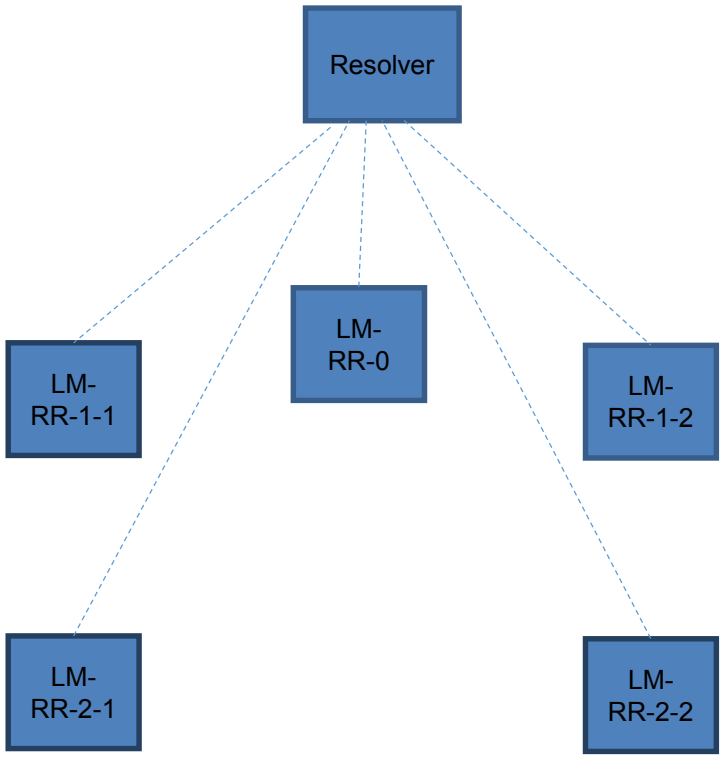
Summary so far…

- UPBN1-1-1 learns destination in its ‘own’ UP i.e. UPBN2-1-1
- UPBN1-1-1 places all these routes as “vanilla” IP destinations in its ‘LM-TIB’
  - This automatically triggers UPBN function resulting in advertisement to LM-RR peer group
- LM-RR learns and places these routes in its local ‘LM-TIB’
  - This automatically triggers LM function resulting in (a) labels allocated (or picked from static configuration) for “vanilla” IP destinations, and (b) origination of corresponding L-BGP route for the IP destinations
- UPBN1-1-1 learns L-BGP route from LM-RR and places these routes also in ‘LM-TIB’
  - Addition of L-BGP routes results in UPBN1-1-1 installing the labels in LFIB with forwarding action necessary to reach corresponding destination

Forwarding packets ‘up’ the hierarchy

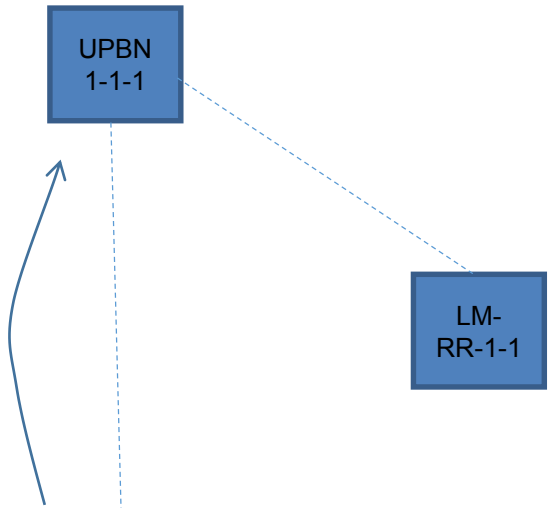
- Slides so far only focused on UP1-1 destination i.e. how UPBN1-1-1 forwards packets from UP0 to UP2-1-1
- How does UP1-1 forward packets from UP1-1 to UP0 destination?
- Solution: Statically configure labels corresponding to UP0 destinations i.e. UPBNs of L1 partitions on all LM-RRs
- But, how is it different from static configuration of labels on all routers?
  - It does not require static configuration on all routers, but only on much fewer LM-RRs!

Why Label Mapping RR? (1)



- LM-RRs apart from reflecting L-BGP routes in ‘LM-TIB’ to UPBN can also reflect them to Resolver (which is another BGP speaker)
  - Configure policy on Resolver so that routes from one LM-RR is not advertised to other LM-RRs
- What does this achieve?
  - Given an end-destination (or destination Server), Resolver can now use recursive route resolution using the L-BGP routes to determine the label stack to reach the end-destination
  - No other protocol is required
- What if number of BGP sessions on Resolver becomes a problem?
  - Let Resolver only speaks to LM-RR-0 and LM-RRs of level 1
  - LM-RRs at level 1 only speak to their corresponding child LM-RRs, and so on…

Why Label Mapping RR? (2)

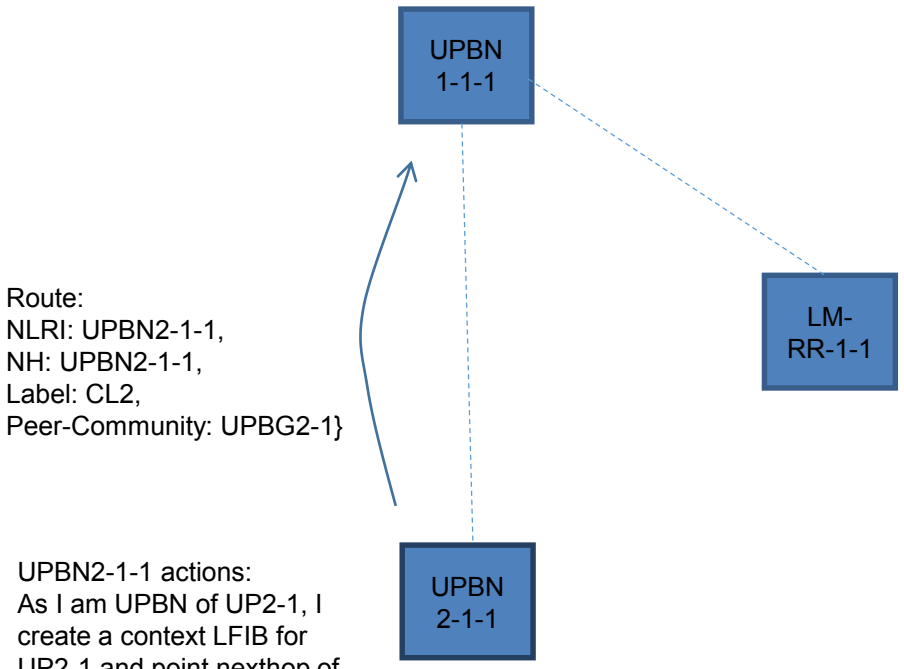


Route:
NLRI: UPBN2-1-1,
NH: UPBN2-1-1,
Label: CL2,
Peer-Community: UPBG2-1}

UPBN2-1-1 actions:
As I am UPBN of UP2-1, I create a context LFIB for UP2-1 and point nexthop of CL2 to point to the context LFIB

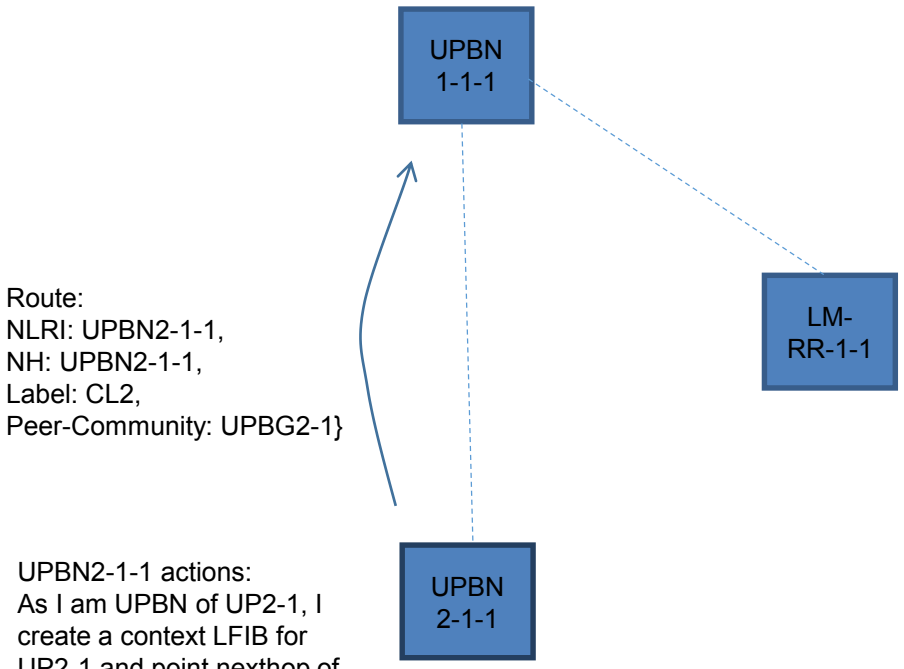
- UPBNs by policy can place all routes learnt from LM-RR in a context LFIB and advertise a label that points to the context LFIB when they advertise themselves to parent UP
- For example, UPBN2-1-1 advertises CL2 to parent UPBN1-1-1 so that packets to UP2-1 destinations are looked up in separate context

Why Label Mapping RR? (3)



- UPBNs by policy can place all routes learnt from LM-RR in a context LFIB and advertise a label that points to the context LFIB when they advertise themselves to parent UP
- For example, UPBN2-1-1 advertises CL2 to parent UPBN1-1-1 so that packets to UP2-1 destinations are looked up in separate context

But, why Label Mapping RR? (3)



- UPBNs by policy can place all routes learnt from LM-RR in a context LFIB and advertise a label that points to the context LFIB when they advertise themselves to parent UP
- For example, UPBN2-1-1 advertises CL2 to parent UPBN1-1-1 so that packets to UP2-1 destinations are looked up in separate context

Next Steps

- Initial draft, feedback is much appreciated
- Add procedure for label distribution for HSDN TE tunnels

BGP Extensions for BIER
draft-xu-idr-bier-extensions-01

Xiaohu Xu (Huawei)

Mach Chen (Huawei)

Keyur Patel (Cisco)

IJsbrand Wijnands (Cisco)

Tony Przygienda (Ericsson)

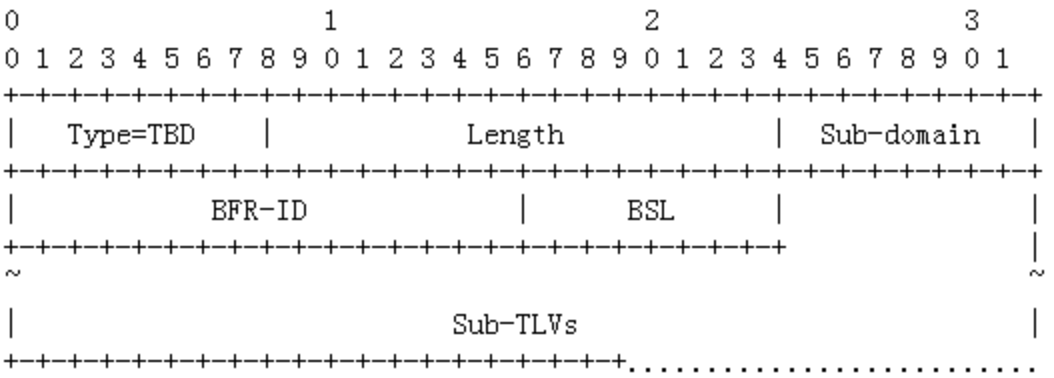
IETF92, Dallas

Motivation

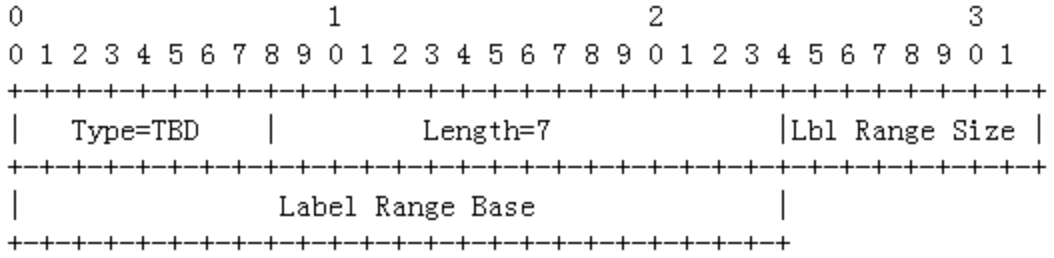
- **Bit Index Explicit Replication (BIER) is applicable in multi-tenant data center network environments for efficient delivery of BUM traffic while eliminating the need for maintaining multicast states in the underlay[I-D.kumar-bier-use-cases].**
- **BGP instead of IGP is used as an underlay in some large multi-tenant data center network environments [I-D.ietf-rtgwg-bgp-routing-large-dc].**
- **This document describes BGP extensions for advertising the BIER-specific information.**
  - **A new optional, transitive path attribute, referred to as the BIER attribute, can be attached to a BGP UPDATE message by the originator so as to indicate the BIER-specific information of a particular BFR which is identified by the /32 or /128 address prefix contained in the NLRI.**

BIER Path Attribute

- **The attribute type code for the BIER Attribute is TBD. The value field of the BIER Attribute contains one or more BIER TLV as shown below:**



- **MPLS-BIER Encapsulation sub-TLV is a sub-TLV of the BIER TLV encoding the MPLS-BIER specific information.**



Originating BIER Attribute

- **An implementation that supports the BIER attribute MUST support a policy to enable or disable the creation of the BIER attribute and its attachment to specific BGP routes.**
- **An implementation MAY disable the creation of the BIER attribute unless explicitly configured to do so otherwise.**
- **A BGP speaker MUST only attach the locally created BIER attribute to a BGP UPDATE message in which at least one of its routable addresses (e.g., a loopback address) is contained in the NLRI.**
  - **The routable address contained in the NLRI is RECOMMENDED to be the one used for establishing BGP sessions.**

Restrictions on Sending/Receiving

- **An implementation that supports the BIER attribute MUST support a per-EBGP-session policy, that indicates whether the attribute is enabled or disabled for use on that session.**
- **The BIER attribute MUST NOT be sent on any EBGP peers for which the session policy is not configured.**
  - **If an BIER attribute is received on a BGP session for which session policy is not configured, then the received attribute MUST be treated exactly as if it were an unrecognised non-transitive attribute. That is, "it MUST be quietly ignored and not passed along to other BGP peers".**
- **To prevent the BIER attribute from "leaking out" of an BIER domain, each BGP router on the BIER domain MUST support an outbound route announcement policy. Such a policy MUST be disabled on each EBGP session by default unless explicitly configured.**

Deployment Considerations

- **It‘s assumed by this document that the BIER domain is aligned with the Administrative Domain (AD) which are composed of multiple ASes (either private or public ASes).**
  - **Use of the BIER attribute in other scenarios is outside the scope of this document.**
- **Since the BIER attribute is an optional, transitive path attribute, a non-BFR BGP speakers could still advertise the received route with a BIER attribute.**
  - **This is desirable in the incremental deployment scenario where a BGP speaker could tunnel a BIER packet or the payload of a BIER packet to a BFER directly if the BGP next-hop of the route for that BFER is a non-BFR.**
- **A BGP speaker is allowed to tunnel a BIER packet to the BGP next-hop if these two BFR-capable BGP neighbors are not directly connected (e.g., multi-hop EBGP) .**

Next Steps

- **Comments?**