IDR WG 93

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.

Monday Session

http://trac.tools.ietf.org/wg/idr/trac/wiki

- Status of Drafts: 14 IESG, 6 new, 5 old, 3 pending
  - 3 at IESG, 7 going to IESG, 2 Early adoption IESG
  - 5 new drafts, 3 in adoption, 4 await implementations
  - 3 Administrative drafts (2 ready for IESG)
  - Not passed WG adoption call: 3
  - Not passed WG LC: 1
- Draft authors will be responsible for
  - Protocol Implementation reports on Wiki
  - Testimonials for Administrative Drafts

Agenda (1)

7/20/2015 17:40 - 18:40 Prague Time

Admin Trivia
==========
Agenda Bashing and Status Q&A 3 minutes
 Due to the short IDR meetings.
 The status is online.
 http://trac.tools.ietf.org/wg/idr/trac/wiki/idr-draft-status

The chairs will answer questions on status on Monday.

Agenda (2)

Existing Work: [30 minutes]

draft-ietf-idr-bgpls-segment-routing-epe.	3 minutes
(Stefano Previdi)	[17:40-17:45]
draft-ietf-idr-te-lsp-distribution	7 minutes
(Jie Dong)	[17:45-17:52]
draft-ietf-idr-bgp-optimal-route-reflection	5 minutes
(Bruno Decraene)	[17:55-18:00]
ietf-rs-bfd	
(Randy Bus)	[18:00-18:10]
draft-jdurand-auto-bfd-00	3 minutes

Monday Agenda (3)

Update on proposed drafts [18:20-18:50]
draft-keyupate-idr-bgp-prefix-sid 5 minutes
(Stefano Previdi) [18:10-18:15]

draft-walton-bgp-hostname-capability-00 10 minutes
(Daniel Walton) [18:15-18:25]

draft-fang-idr-bgplu-for-hsdn 15 minutes
[Luyuan Fang] [18:25-18:40]

Friday Agenda (1)

Friday: 7/25/2015 11:50-13:20 [90 minutes]

Agenda Bashing and status [5 minutes]
draft-ietf-idr-flowspec-l2vpn-01 [5 minutes]
[Weiguo Hao]

IDR yang drafts [holding spot] [15 minutes]

draft-hao-ls-trill-01 [10 minutes]
[Donald Eastlake]

Friday Agenda (2)

Flow Specification Changes [50 minutes]

draft-hao-idr-flowspec-nv03-00 [5 minutes] [Weiguo Hao]
draft-li-idr-flowspec-rpd [15 minutes] [Eric Wu]
draft-liang-idr-bgp-flowspec-label [10 minutes] [Jianjie You]
draft-wu-idr-flowspec-yang-cfg [10 minutes] [Eric Wu]
draft-liang-idr-flowspec-orf-00 [10 minutes] [Jianjie You

Label Assignment
draft-zhang-idr-upstream-assigned-label-solution-00
[Sandy]

IDR WG

Segment Routing BGPLS Egress Peer Engineering Extensions

draft-ietf-idr-bgpls-segment-routing-epe-00

Stefano Previdi (sprevidi@cisco.com)
Clarence Filfsils (cfilsfil@cisco.com)
Saikat Ray (raysaikat@gmail.com)
Keyur Patel (keyupate@cisco.com)
Jie Dong (jie.dong@huawei.com)
Mach (Guoyi) Chen (mach.chen@huawei.com)
Acee Lindem (acee@cisco.com)

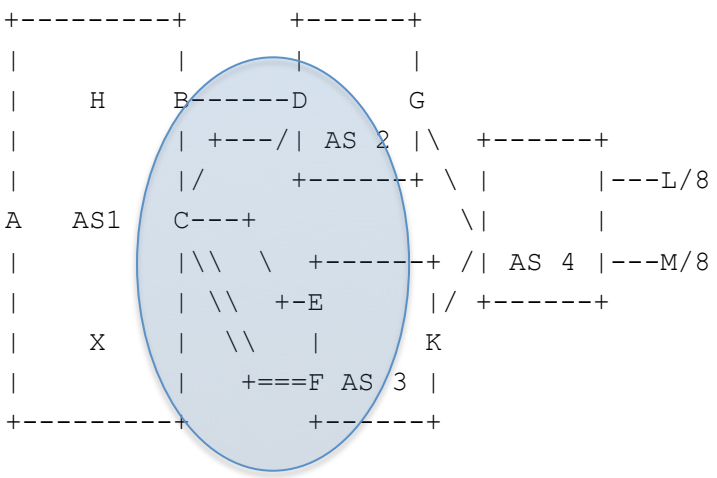
IETF93 – Prague, July 2015

SR-EPE BGP-LS Extensions

- draft-ietf-idr-bgpls-epe-extensions-00
- WG doc (draft-previdi-idr-bgpls-epe-extensions-03)

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.

SR-EPE BGP-LS Extensions

- Changes:
  - Clarification text on BGP Identifier
  - PeerNode SID and PeerAdj SID TLVs
    - In previous version both object where represented by an Adj-SID TLV
    - Now use two distinct SIDs: PeerNode and PeerAdj
  - Interface addresses/identifiers
  - New contributor: Acee

Distribution of TE LSP State using BGP

draft-ietf-idr-te-lsp-distribution-03

Jie Dong, Mach Chen (Huawei)

Hannes Gredler (Juniper)

Stefano Previdi (Cisco)

Jeff Tantsura (Ericsson)

IETF93 IDR Jul. 2015 Prague

Overview

- Provide a mechanism for collecting TE LSP states
- Based on the BGP-LS architecture
  - unified protocol for network layer information distribution
- Complimentary to the PCE based LSP report
  - Some LERs may not be PCC
  - Reduced the session overhead with BGP RR

Updates since WG Adoption

- Add support for Segment Routing TE LSPs
  - A new Protocol-ID needs to be assigned for Segment Routing
- Add “Operational Considerations” section
  - Align with the BGP-LS base document
- Update the TE LSP objects list in the LSP State TLV
- Update the IANA section

Comments Received

- How to specify the switching type of non-packet LSPs?
  - A: could use the Generalized Label Request Object in the LSP State TLV
- How to distinguish RSVP and PCEP objects?
  - A: could define top level TLVs based on the source of TE LSP information
  - e.g. RSVP, PCEP, etc.
- Authors would like to discuss the solutions for these comments, then update the draft accordingly

Next Steps

- Solicit more review and comments
- Revise the draft accordingly

BGP Optimal Route Reflection (BGP-ORR)

draft-ietf-idr-bgp-optimal-route-reflection-10

Robert Raszuk	Mirantis
Christian Cassar	Cisco Systems
Erik Aman	TeliaSonera
Bruno Decraene	Orange
Stephane Litkowski	Orange Business Service

Recaps of goals

- Providing hot potato routing
  - closest ASBR from the client/ingress perspective
- Decouple RR best path selection from RR IGP location
- Ease RR "mobility" with regards to its clients
  - network topology change
  - new clients
  - changing RR location (including during maintenance)

Significant changes introduced in -09 & -10

- A single solution kept: overwrite RR IGP location during best path selection
  - either one arbitrary IGP location for the whole RR
  - or one location per (peer) group
  - or up to one location per client (i.e. client's IGP location)
  - choice of granularity is left to the implementation.
- Alternative options/refinement dropped:
  - angular distance approximation
  - client-RR signaling of Group ID
  - mandating per client's computation
  - per client BGP policy
- Still requires that RR knows all path before IGP tie-break.
  - e.g. BGP ADD-PATH between RR
- Many editorial changes / text rewrite.

Significant changes introduced in -09 & -10 (2)

- As a result:
  - Proposal has being simplified
  - Draft is now in line with existing implementations
  - Well applicable to NFV where the RR function may be hosted on general purpose IT resources (less "in" the network) and more easily moved.

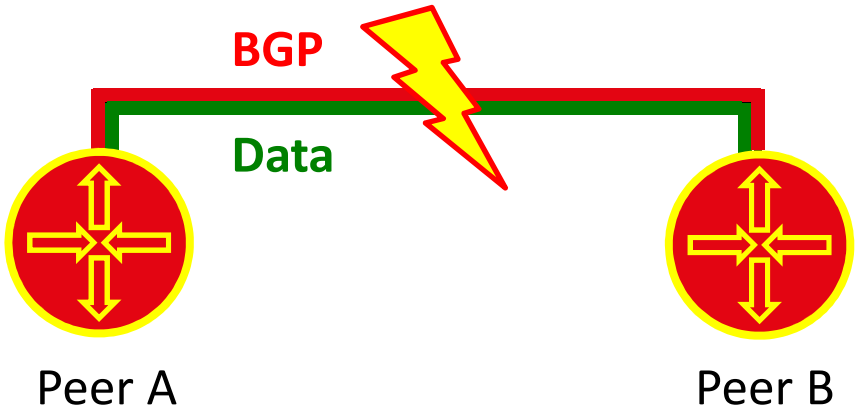
- Given all changes, you may want to re-read latest version.

Thank you

Making Route Servers Aware of Data Link Failure at IXPs

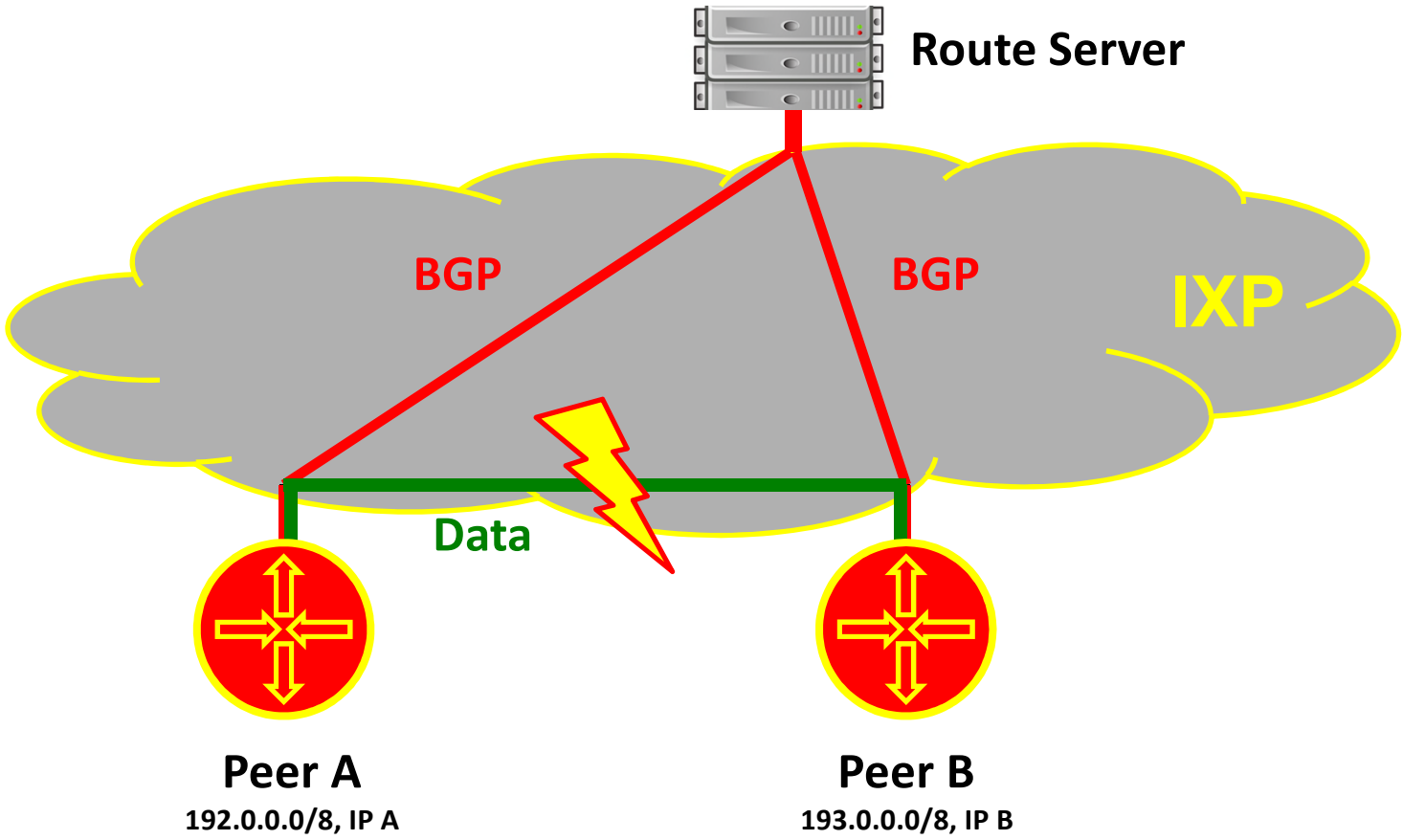
Arnold Nipper, Randy Bush, Jeffrey Hass, John Scudder, Thomas King

Typical Scenario: BGP Session



If the **data plane** breaks, the **control plane** is able to detect this.

Challenge: Route Server at IXPs



Problem: If the **data plane** breaks, the **control plane** is not able to detect this. Data traffic is lost!

Solution

1. Client routers must have a means of verifying connectivity amongst themselves
   ➔ **Bidirectional Forwarding Detection, RFC 5880**
2. Client routers must have a means of communicating the knowledge so gained back to the route server
   ➔ **North-Bound Distribution of Link-State and TE Information using BGP, Draft**

- Bidirectional Forwarding Detection (BFD):
  - Hello packets are exchanged between two client routers (comparable to BGP Hello)
  - Asynchronous mode (default)
  - Rate: 1 packet / second, detection after 3 missing packets
- North-Bound Distribution of Link-State and TE Information using BGP (BGP-LS):
  - Model IXP network as nodes (client routers and route server) and links (data plane reachability)
  - Per peer: Next-Hop Information Base (NHIB) stores reachability for all next-hops

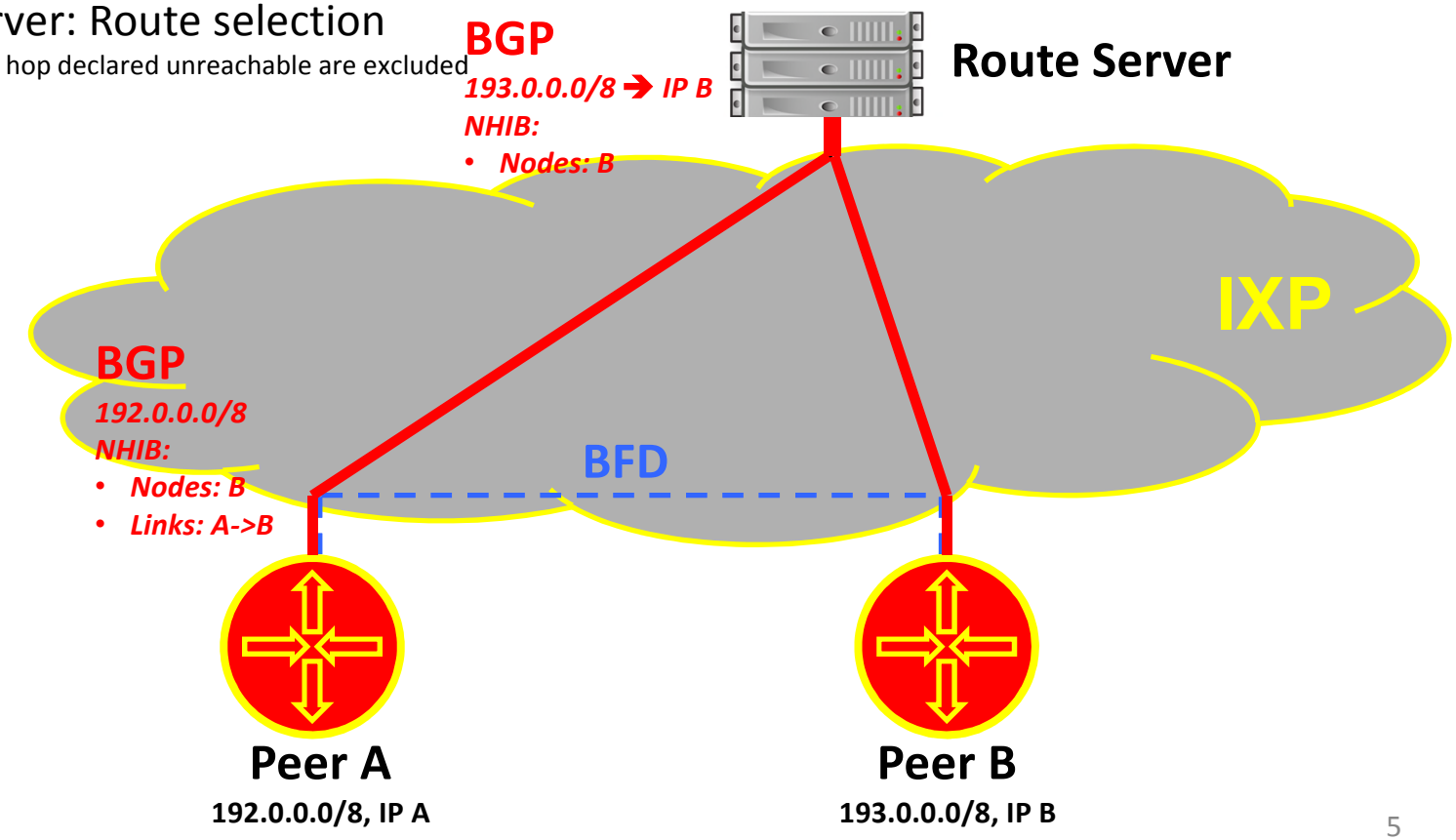
Solution

1. Route Server: NHIB updated
2. Client Router: Verify connectivity

BFD connections are setup automatically

3. Client Router: NHIB updated
4. Route Server: Route selection

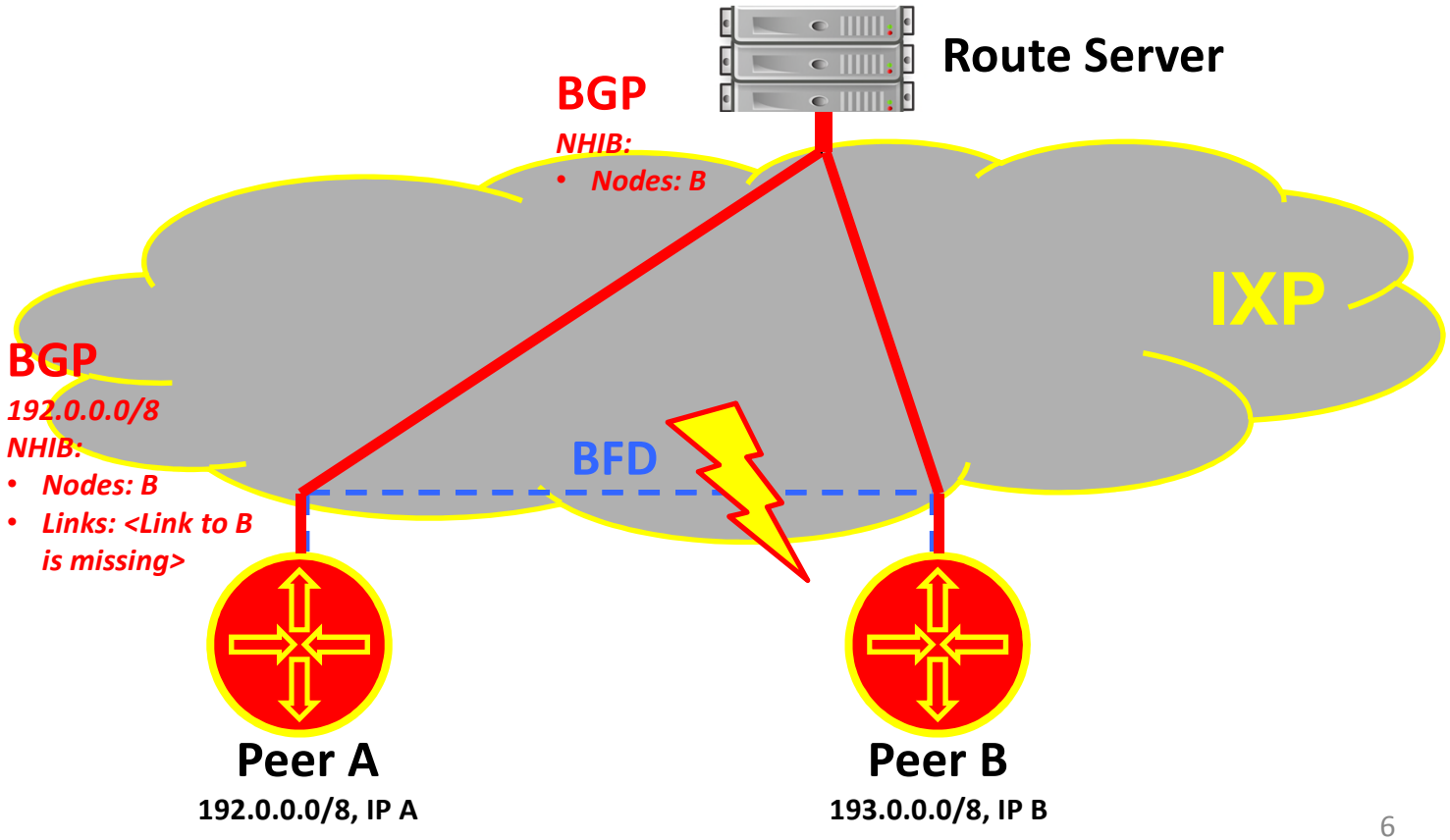
All routes with next hop declared unreachable are excluded



Data Link Breakage

1. Client Router: Data link break detected
2. Client Router: NHIB updated
3. Route Server: Route selection

All routes with next hop declared unreachable are excluded



Status of Internet Draft

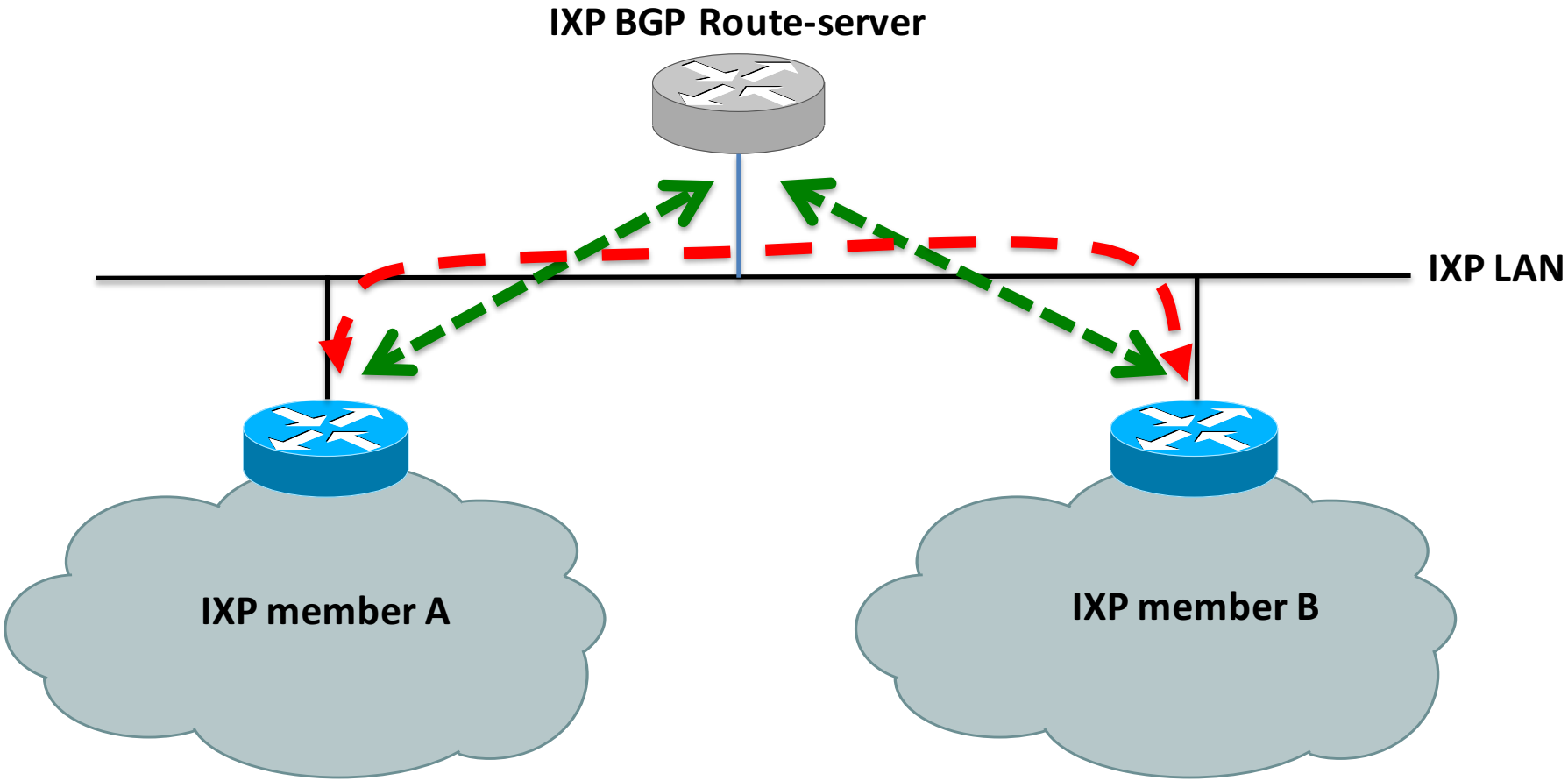
- IDR WG adoption achieved
- Version 00 -> 01: Switch from NH-Cost to BGP-LS
  - NH-Cost Internet Draft is inactive and not supported by router vendors
  - BGP-LS provides similar mechanisms and is / will be implemented by router vendors

draft-jdurand-auto-bfd-00
Path validation toward BGP next-hop with AUTO-BFD

Jérôme Durand, Cisco

David Freedman, Claranet

Problem we want to solve (sec 1)



2 EBGP peerings Established

Connectivity between peers down

➔ BLACK HOLE

Solution requirements (sec 3)

- Solution independent of IXP. IXP members MUST be able to detect and remedy to such issues without anything on IXP

➔ Main difference from **draft-ymbk-idr-rs-bfd**

- Other requirements detailed in section 3

Auto-BFD Solution (sec 4)

- AUTO-BFD is configured on the BGP peering to the BGP RS.
- Every time a new BGP next-hop is received from this peering, AUTO- BFD triggers a new BFD session with this next-hop
  - Asynchronous mode
  - Timers and security configuration can be locally added
- Routes coming from the AUTO-BFD enabled BGP neighbor are then checked based on the BGP next-hop and its BFD session state.
- Acceptance of routes is then subject to the administrative policy based on BFD session state (discard route, change LP…)

➔ IXP member is in control

Session ageing 1/2 (sec 5)

- Important: we don't want sessions to stay for ever
- The tricky part: it must work with asymmetric policies
  - A stops sending routes to B (through BGP RS)
  - B still sends routes to A (through BGP RS)

  ➔ We don't want B to tear down the BFD session as A would then believe B is down

Session ageing 2/2 (sec 5)

- IXP members implementing AutoBFD signal they still need the session (ie. that they still receive routes) using bfd.LocalDiag in BFD control packets
- Members do not tear down a session when they receive this flag.
- If there is no flag, members tear down the session after a give timer

➔ **No change to BFD protocol**

Ask for the WG

- Questions ? Comments ?
- Adopt as WG document

Thank you !

IDR WG

Segment Routing Prefix SID extensions for BGP

draft-keyupate-idr-bgp-prefix-sid-05

Stefano Previdi (sprevidi@cisco.com)
Clarence Filfsils (cfilsfil@cisco.com)
Keyur Patel (keyupate@cisco.com)
Arjun Sreekantiah (asreekan@cisco.com)
Saikat Ray (raysaikat@gmail.com)
Hannes Gredler (hannes@juniper.net)
Acee Lindem (acee@cisco.com)

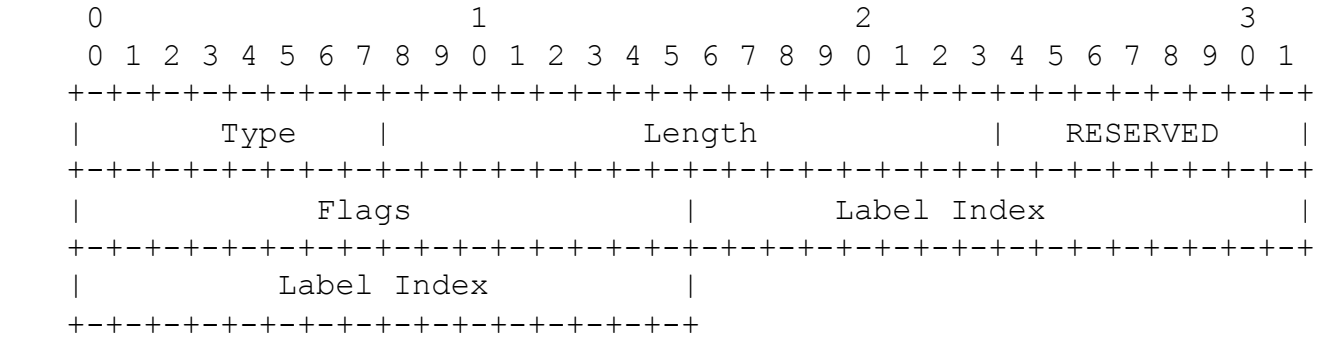
IETF93 – Prague, July 2015

BGP Prefix-SID Attribute

- New BGP Attribute addressing the use case described in draft-filsfils-spring-segment-routing-msdc
- Version 5
  - Merged with draft-gredler-idr-bgplu-prefix-sid-00
  - Added support of SR-IPv6 dataplane with SR-IPv6-SID TLV
  - Added Originator SRGB TLV when SRGB is to be learned through Prefix-SID attribute
  - Applicable to Labeled unicast prefixes (RFC3107) and MP-BGP unlabeled unicast IPv6 prefixes (RFC4760)
- Multiple implementations exist
- Ready for WG adoption

BGP Prefix-SID Attribute

- Label Index TLV

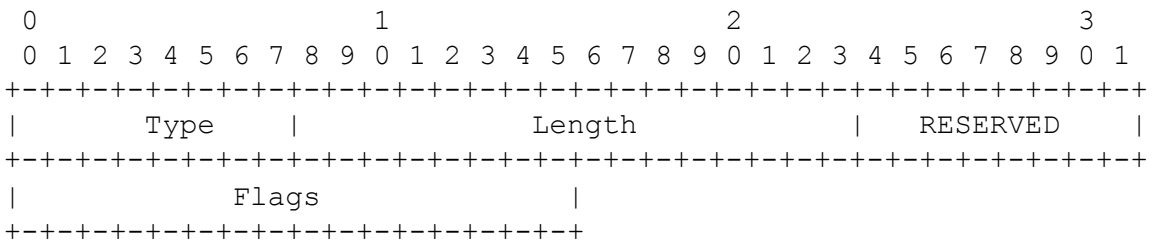


where:

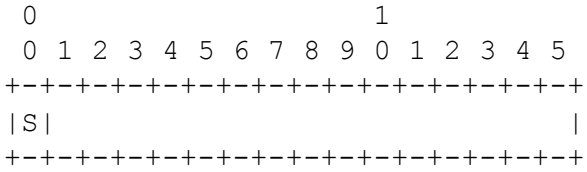
- Type is 1.
- Length: is 7, the total length of the value portion of the TLV.
- RESERVED: 8 bit field. SHOULD be 0 on transmission and MUST be ignored on reception.
- Flags: 16 bits of flags. None are defined at this stage of the document. The flag field SHOULD be clear on transmission and MUST be ignored at reception.
- Label Index: 32 bit value representing the index value in the SRGB space.

BGP Prefix-SID Attribute

- SR IPv6 SID



o Type is 2.
o Length: is 3, the total length of the value portion of the TLV.
o RESERVED: 8 bit field. SHOULD be 0 on transmission and MUST be ignored on reception.
o Flags: 16 bits of flags defined as follow:

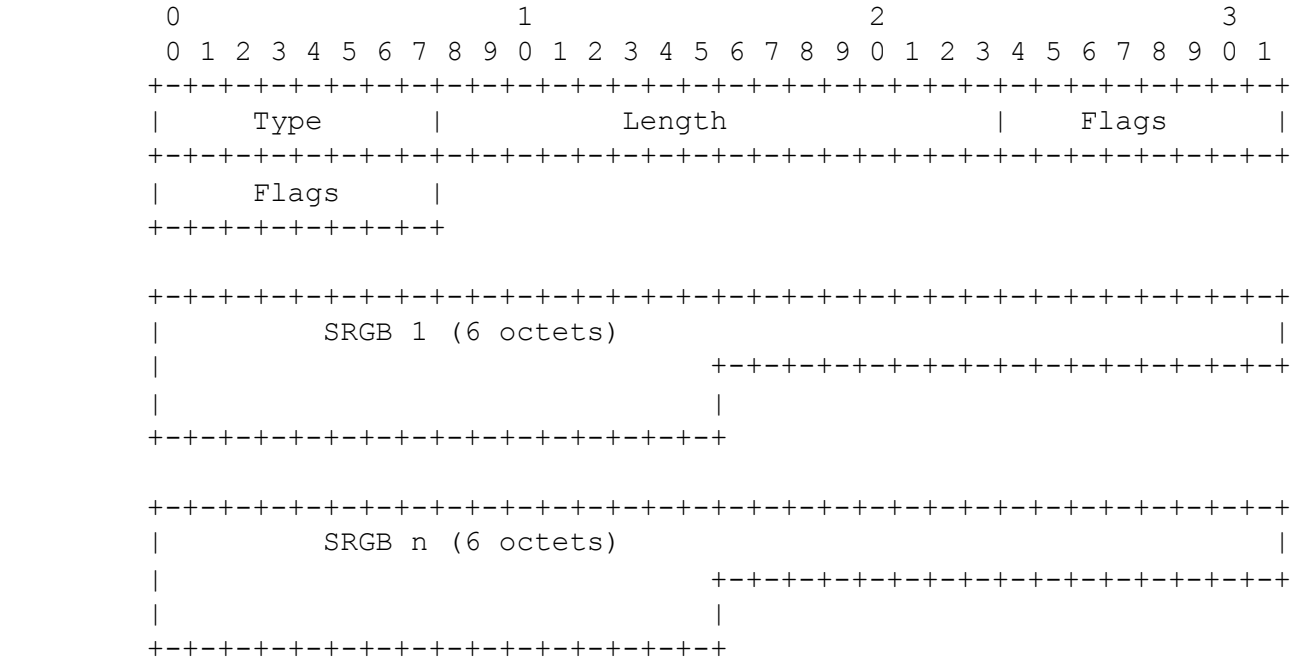


where:

S flag: if set then it means that the BGP speaker attaching the Prefix-SID Attribute to a prefix is capable of processing the IPv6 Segment Routing Header (SRH, draft-previdi-6man-segment-routing-header) for the segment corresponding to the originated IPv6 prefix.

BGP Prefix-SID Attribute

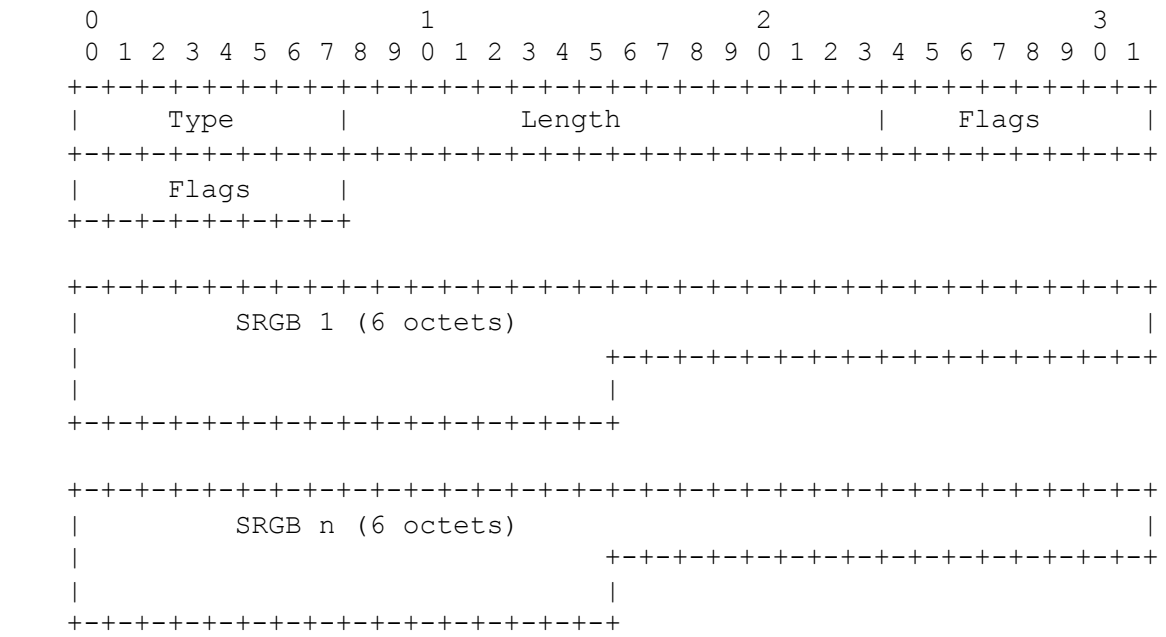
- Originator SRGB



- Type is 3.
- Length is the total length of the value portion of the TLV: 2 + multiple of 6.
- Flags: 16 bits of flags. None are defined in this document. Flags SHOULD be clear on transmission an MUST be ignored at reception.
- SRGB: 3 octets of base followed by 3 octets of range. Note that SRGB field MAY appear multiple times.

BGP Prefix-SID Attribute

- Originator SRGB TLV



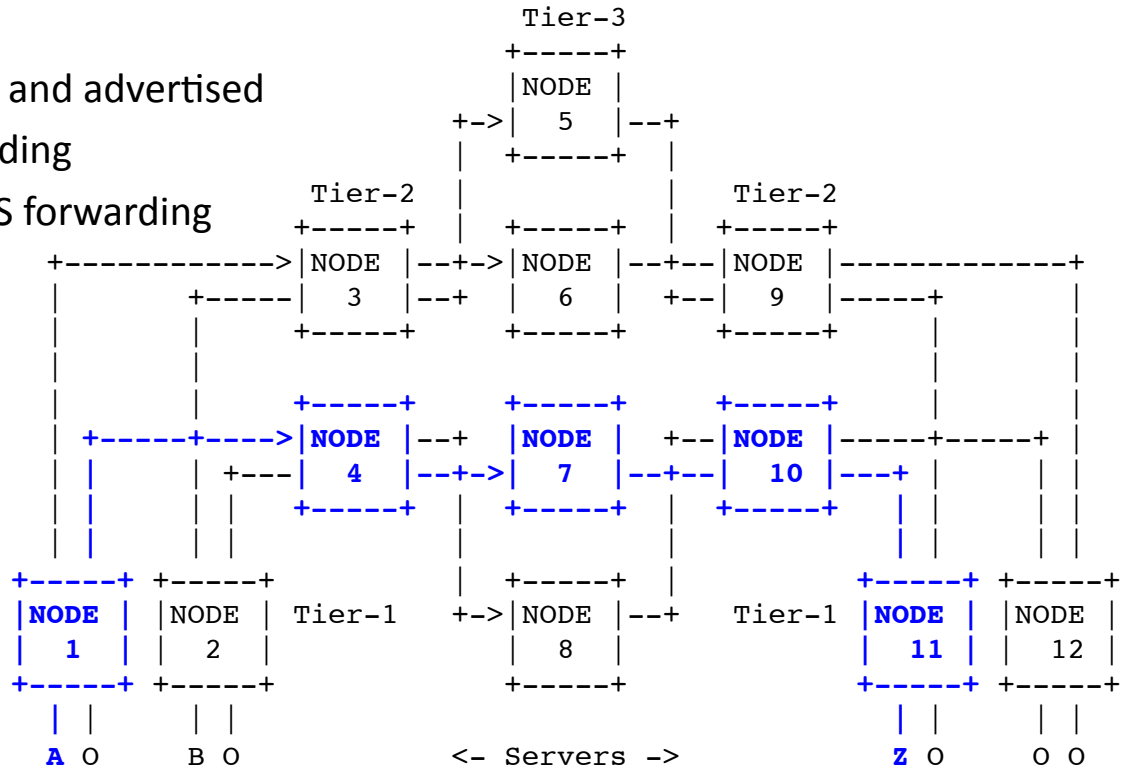
The Originator SRGB TLV contains the SRGB of the router originating the prefix to which the BGP Prefix SID is attached and MUST be kept in the Prefix-SID Attribute unchanged during the propagation of the BGP update.

The originator SRGB describes the SRGB of the node where the BGP Prefix Segment end. It is used to build SRTE policies when different SRGB's are used in the fabric (draft-filsfils-spring-segment-routing-msdc).

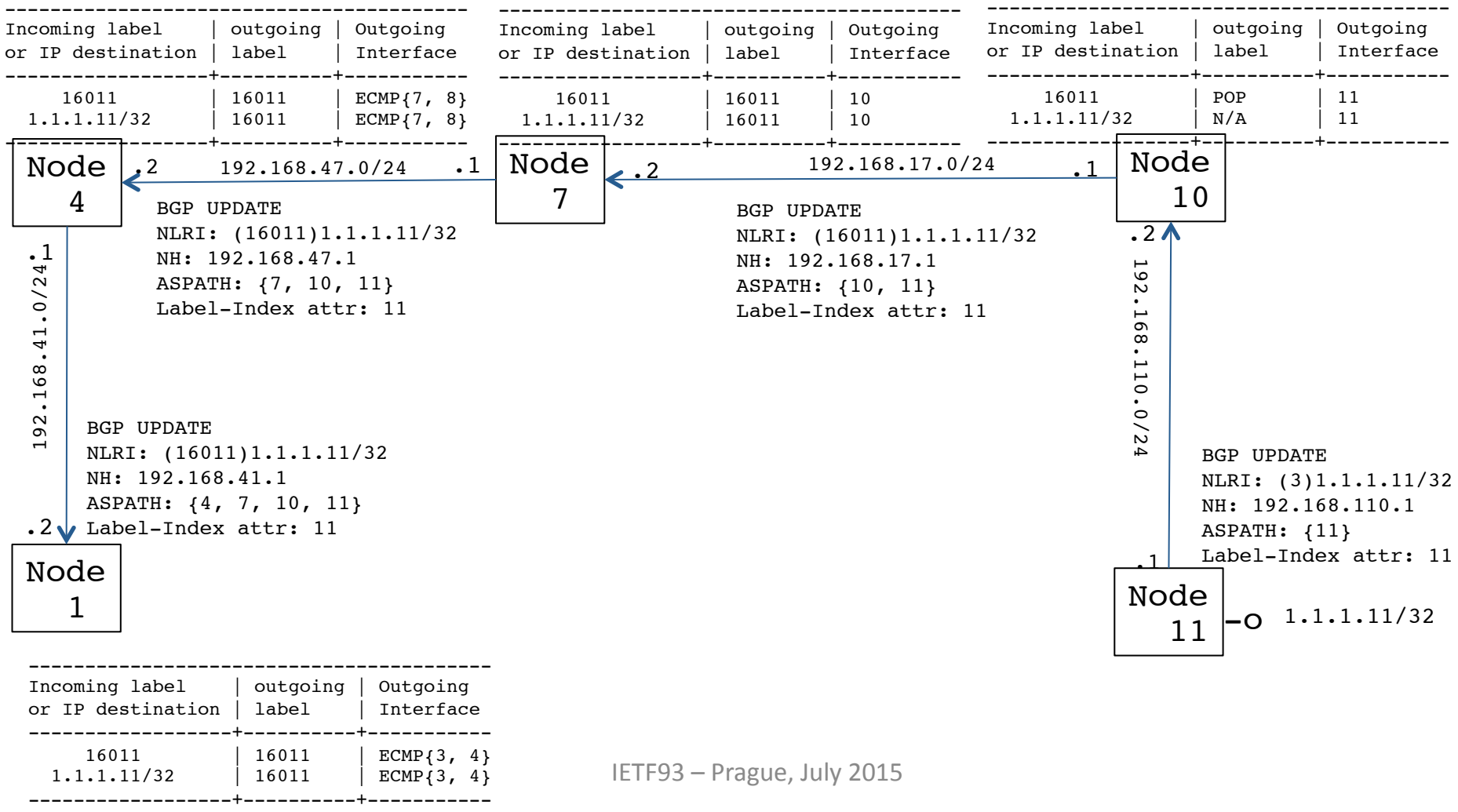
The originator SRGB may only appear on Prefix-SID attribute attached to prefixes of SAFI 4 (labeled unicast, [RFC3107]).

Reference topology

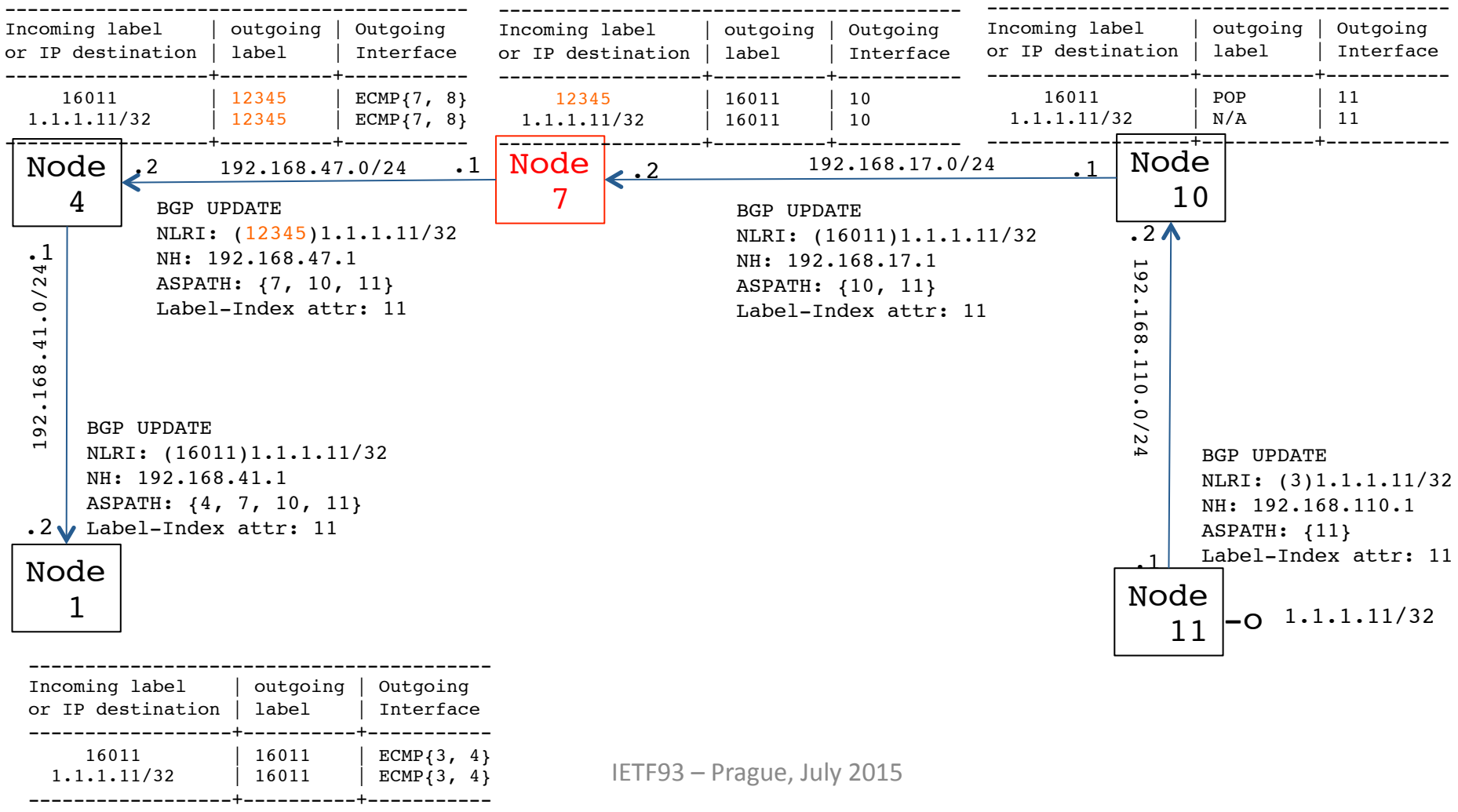
- Node ‘x’ has ASN ‘x’
- BGP IPvX labeled-unicast sessions (3107) between directly connected nodes
- Node ‘x’ has loopback 1.1.1.x/32
- Loopbacks are redistributed into BGP and advertised
- Tier-2 and Tier-3 nodes: MPLS forwarding
- Tier-1 nodes: IP2MPLS or MPLS2MPLS forwarding
- SRGB: [16000, 23999]
- Label index for 1.1.1.x/32 is ‘x’



BGP Prefix SID: Control and dataplane



BGP Prefix SID: Non-SR node in the middle



Hostname Capability for BGP

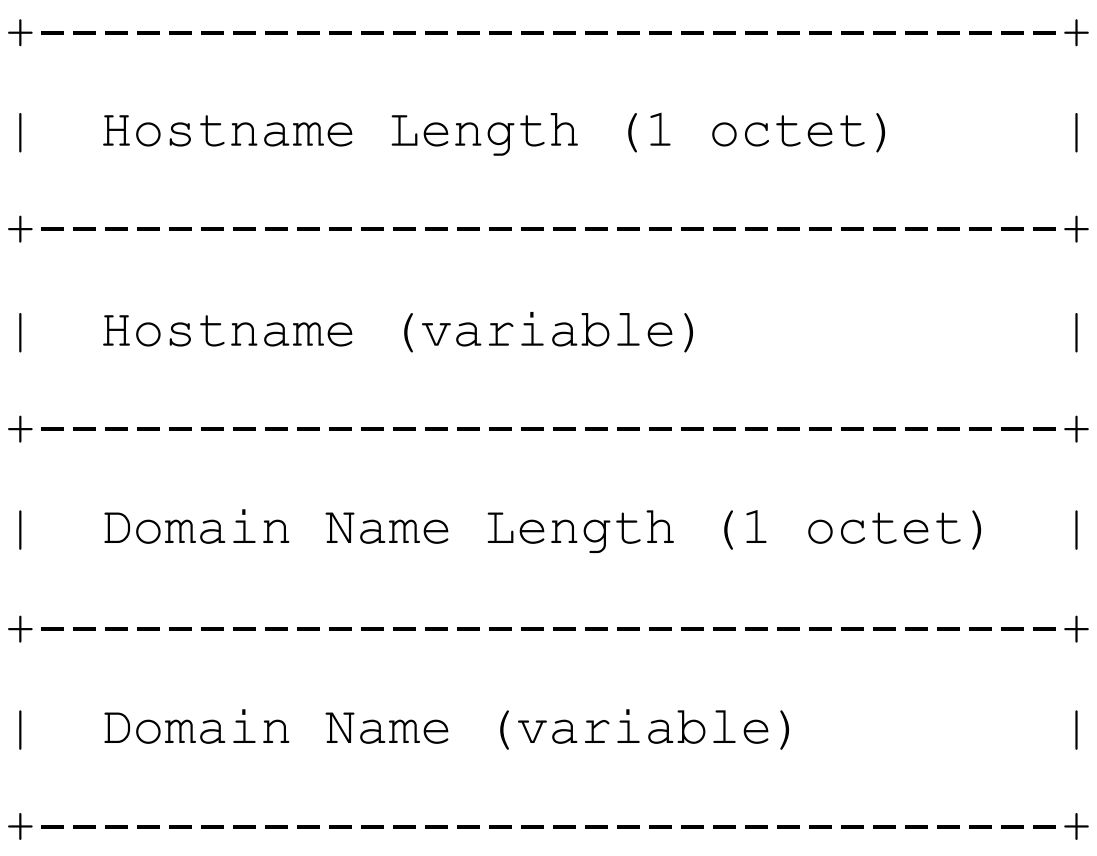
draft-walton-bgp-hostname-capability-01

Daniel Walton
Dinesh Dutt
Cumulus Networks

BGP Hostname Capability

- Advertise the normal hostname of the node to peers via an optional capability
- UTF-8
- Used for display/troubleshooting purposes only
- Knob will be used to turn on/off the display

BGP Hostname Capability



Motivation

- BGP speakers in the data center peer to physical interfaces and not loopbacks
  - There is no IGP
  - BGP peer to everyone that is directly connected
- This means you can't memorize “10.0.0.1 is spine-1, 10.0.0.2 is spine-2, etc”
  - Peer via /30s or link-local addresses so these are unique everywhere
  - Every speaker is peering to a different set of addresses

Motivation

- Knowing the hostname maybe useful outside of DC too
  - No longer need to memorize which loopback is which speaker

Simple Example - Before

```
leaf-11# show ip bgp summ

[snip]

Neighbor             V    AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down  PfxRcd

fe80::202:ff:fe00:5  4 65200   14878   14875      0   0    0 12:18:34      7

fe80::202:ff:fe00:9  4 65200   14880   14876      0   0    0 12:18:27      7

fe80::202:ff:fe00:B  4 65001   14879   14877      0   0    0 12:18:19      2

fe80::202:ff:fe00:E  4 65002   14881   14879      0   0    0 12:18:12      2


Total number of neighbors 4

leaf-11#
```

Simple Example - After

```
leaf-11# show ip bgp summ

[snip]

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  PfxRcd
spine-1         4 65200   14878   14875        0    0    0 12:18:34        7
spine-2         4 65200   14880   14876        0    0    0 12:18:27        7
tor-11          4 65001   14879   14877        0    0    0 12:18:19        2
tor-12          4 65002   14881   14879        0    0    0 12:18:12        2

Total number of neighbors 4

leaf-11#
```

Next Version

- Will require draft-ietf-idr-ext-opt-param-03
- Will provide guidance on
  - Output when peering to the same speaker multiple times
  - Output when two speakers use the same name

BGP-LU for HSDN Label Distribution
draft-fang-idr-bgplu-for-hsdn-01

Luyuan Fang, Chandra Ramachandran, Fabio Chiussi, Yakov Rekhter

IDR meeting, IETF 93

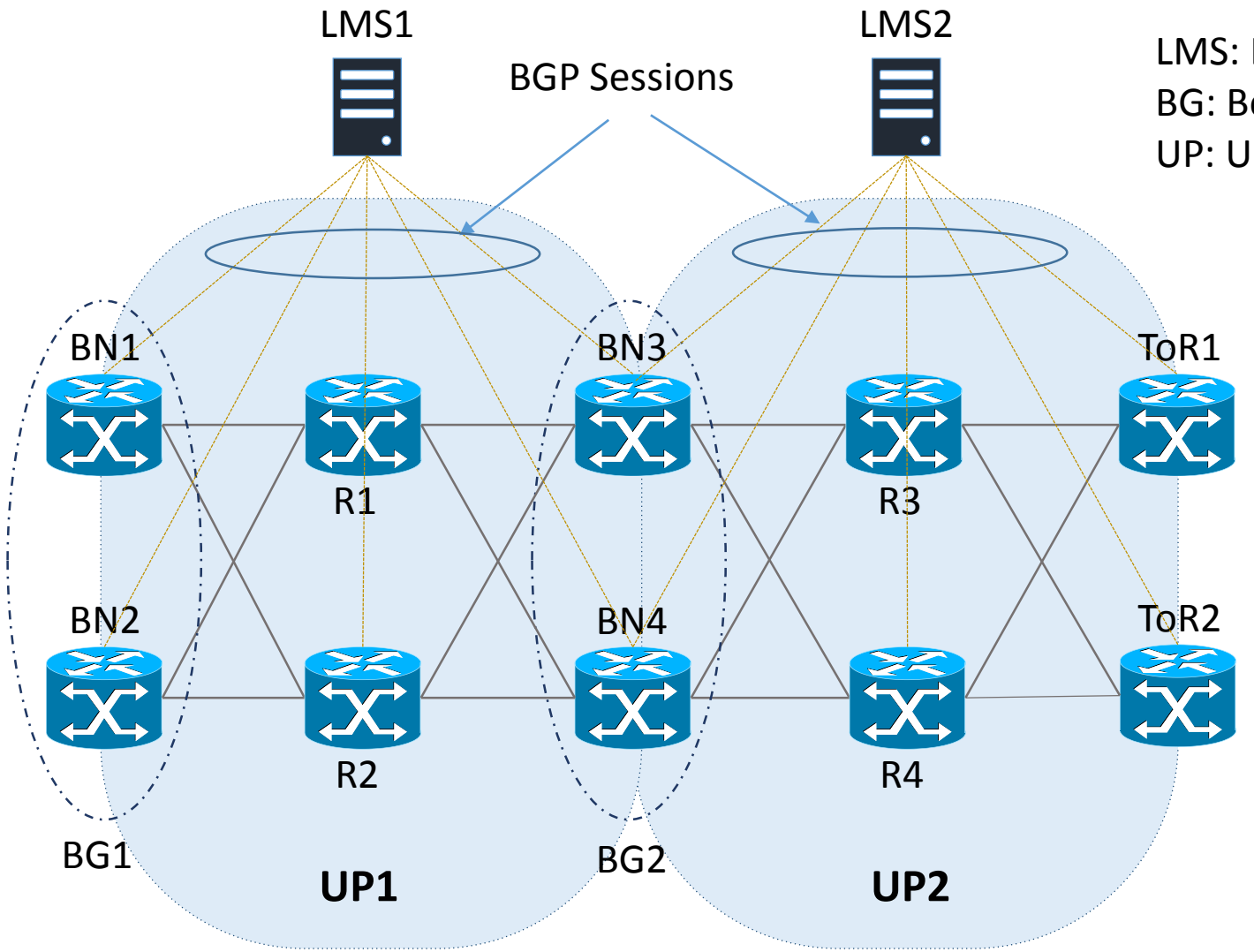
July 20, 2015, Prague

Summary of changes from 00 version

- Broaden applicability of the proposal
  - Any partitioned underlay network using BGP-LU label stacks
- Modified procedures so that they are applicable for partitions running either eBGP or IGP
  - Previous version only covered IGP
- Introduced a new extended community type so that the procedures can be used to:
  - Enable border nodes to unambiguously signal the remote BGP speaker(s) that new BGP-LU procedures requesting partition-unique label(s) should be executed
  - Enable Border Node and BGP speaking Label Mapping Server to scope the label request and the response to a unique partition

Example Topology

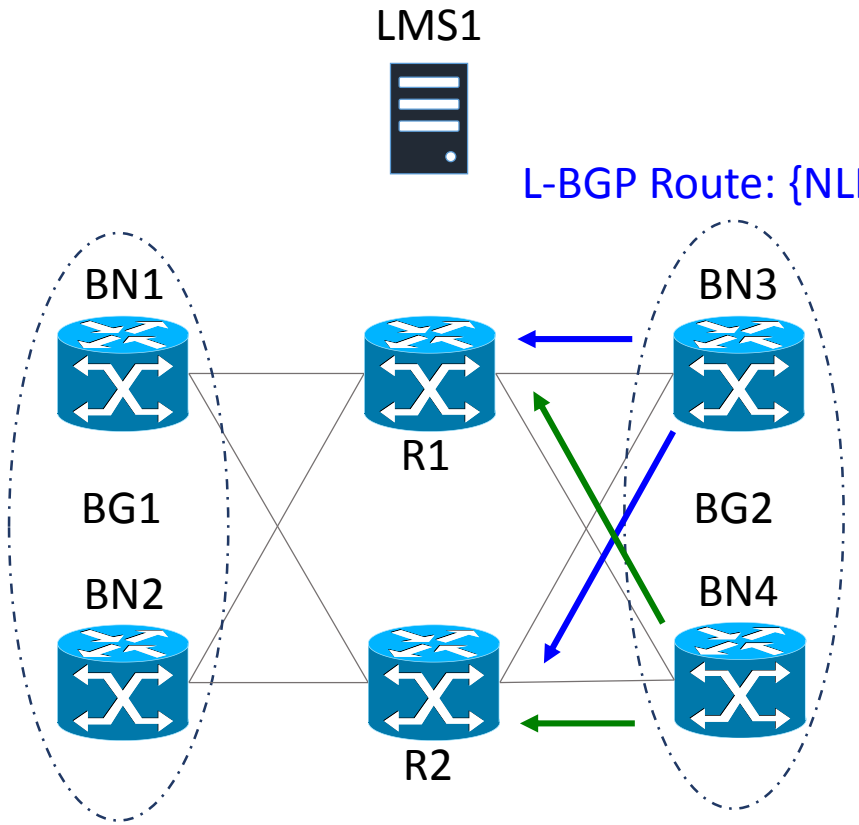
- An abstract model of DC in two level hierarchy



LMS: Label Mapping Server
BG: Border Group
UP: Underlay Partition

BN3, BN4, ToR1 and ToR2 have BGP peering with LMS2
BN1, BN2, BN3 and BN4 have BGP peering with LMS1

BNs of UP1 learn routes to BN3 and BN4



L-BGP Route: {NLRI = BN3-Loopback:CL3, NH = BN3, Com = BG2}

- CL3/CL4 is NULL label (PHP) if same LFIB is used for UP1 and UP2. Otherwise locally allocated Context Label is used
- BN3 and BN4 append a BGP community (value is BG2) when they advertise themselves in UP1

L-BGP Route: {NLRI = BN4-Loopback:CL4, NH = BN4, Com = BG2}

BNs of UP1 learn route to BN3 and BN4 (2)

L-BGP Routes (View of BN1):

{NLRI = BN3-Loopback:L113, NH = R1, Com = BG2}

{NLRI = BN4-Loopback:L114, NH = R1, Com = BG2}

{NLRI = BN3-Loopback:L123, NH = R2, Com = BG2}

{NLRI = BN4-Loopback:L124, NH = R2, Com = BG2}

- Interior routers in UP1 (R1 and R2) do not modify or remove border-group community in the L-BGP route
- When BN1 and BN2 receive the L-BGP routes for BN3 and BN4, they can conclude that BN3 and BN4 belong to BG2 group.

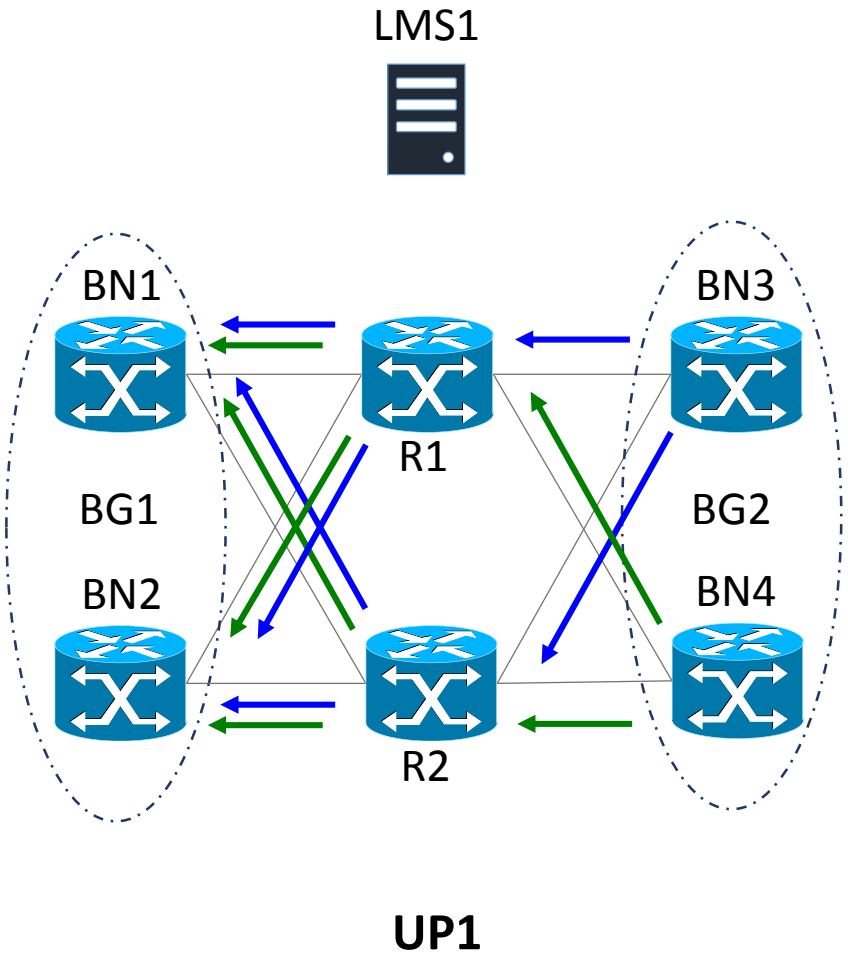
L-BGP Routes (View of BN2):

{NLRI = BN3-Loopback:L113, NH = R1, Com = BG2}

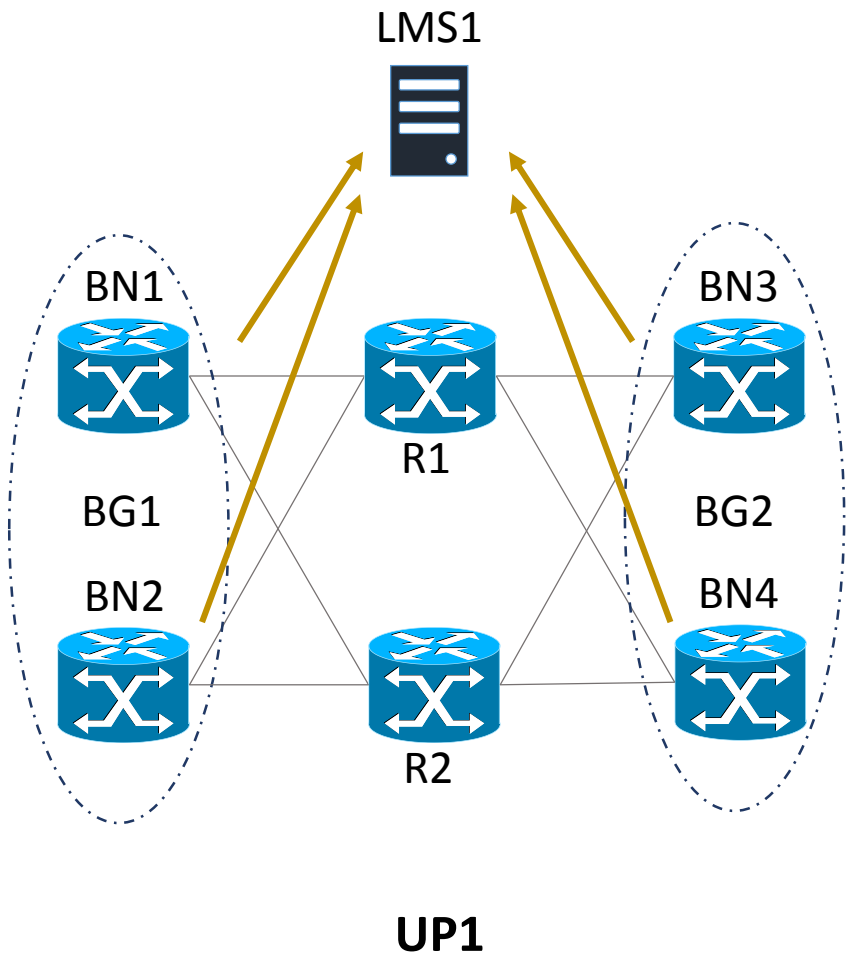
{NLRI = BN4-Loopback:L113, NH = R1, Com = BG2}

{NLRI = BN3-Loopback:L123, NH = R2, Com = BG2}

{NLRI = BN4-Loopback:L124, NH = R2, Com = BG2}



All BNs in UP1 Advertise Themselves to LMS1



L-BGP Routes (View of LMS1):

{NLRI = BN1-Loopback:CL1, NH = BN1, Com = BG1}

{NLRI = BN2-Loopback:CL2, NH = BN2, Com = BG1}

{NLRI = BN3-Loopback:CL3, NH = BN3, Com = BG2}

{NLRI = BN4-Loopback:CL4, NH = BN4, Com = BG2}

- LMS may not run regular BGP decision processes to compute routes
- LMS learns the group membership of BN3 and BN4 from the L-BGP advertisement

Partition labels – new procedures

- In the example so far, BN1 and BN2 have learnt BN3 and BN4 using normal BGP-LU procedures
- What is new?
  - BN1 and BN2 are configured to be partition border nodes for UP1 (the partition represented in the BGP extended community value)
  - When BN1 and BN2 learn a destination (BN3 or BN4) through L-BGP from BGP peers (R1 and R2) that belong to UP1 partition, then BN1 and BN2 do not allocate a label from platform label space and do not re-advertise
  - Instead, BN1 and BN2 “learn” the label for the destination (BN3 or BN4) in “partition label space” from the Label Mapping Server (LMS) through the new procedures specified in the draft

BN1 learns partition label for BN3 and BN4

IP Routes (from BN1 to LMS1):

{NLRI = BN3-Loopback, NH = BN1, Com = BG1, **Ext-com = R:UP1-context**}

{NLRI = BN4-Loopback, NH = BN1, Com = BG1, **Ext-com = R:UP1-context**}

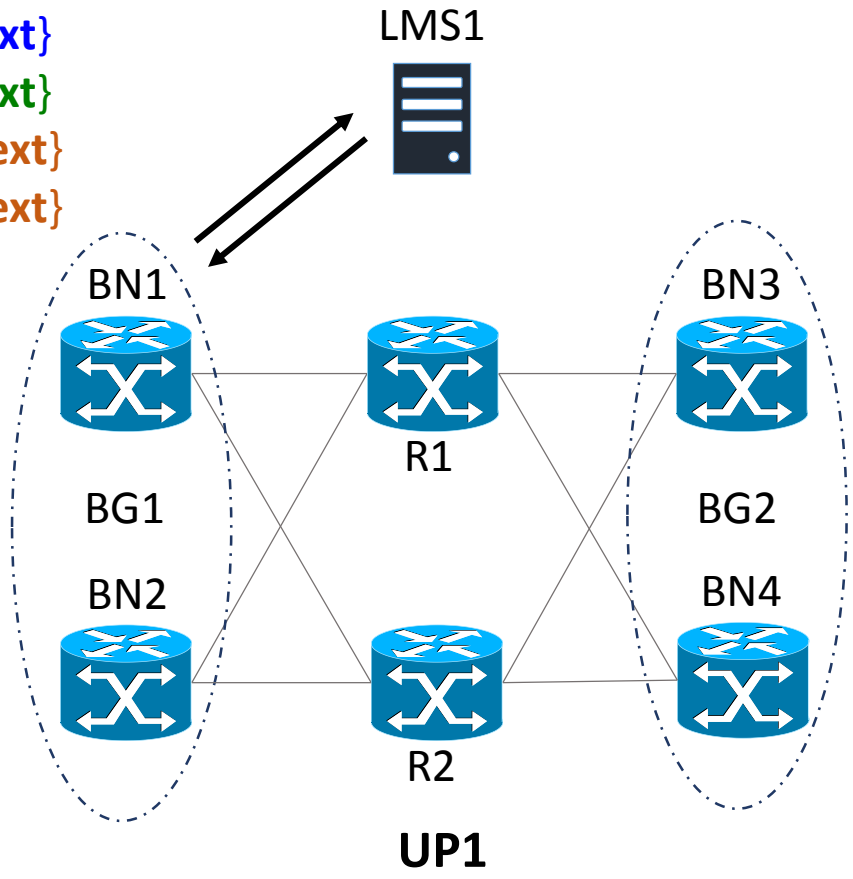
L-BGP Route (from LMS1 to BN1):

{NLRI = BN3-Loopback:PL13, NH = BN1, Com = BG1, **Ext-com = 0:UP1-context**}

{NLRI = BN4-Loopback:PL14, NH = BN1, Com = BG1, **Ext-com = 0:UP1-context**}

{NLRI = BN3-Loopback:PLG2, NH = BN1, Com = BG1, **Ext-com = G:UP1-context**}

{NLRI = BN4-Loopback:PLG2, NH = BN1, Com = BG1, **Ext-com = G:UP1-context**}



PLG2: Partition Label assigned for Border Node Group (BG2)

Ext-com:

- R: Request
- G: Group

BN2 learns partition label for BN3 and BN4

IP Routes:

{NLRI = BN3-Loopback, NH = BN2, Com = BG1, Ext-com = R:UP1-context}

{NLRI = BN4-Loopback, NH = BN2, Com = BG1, Ext-com = R:UP1-context}

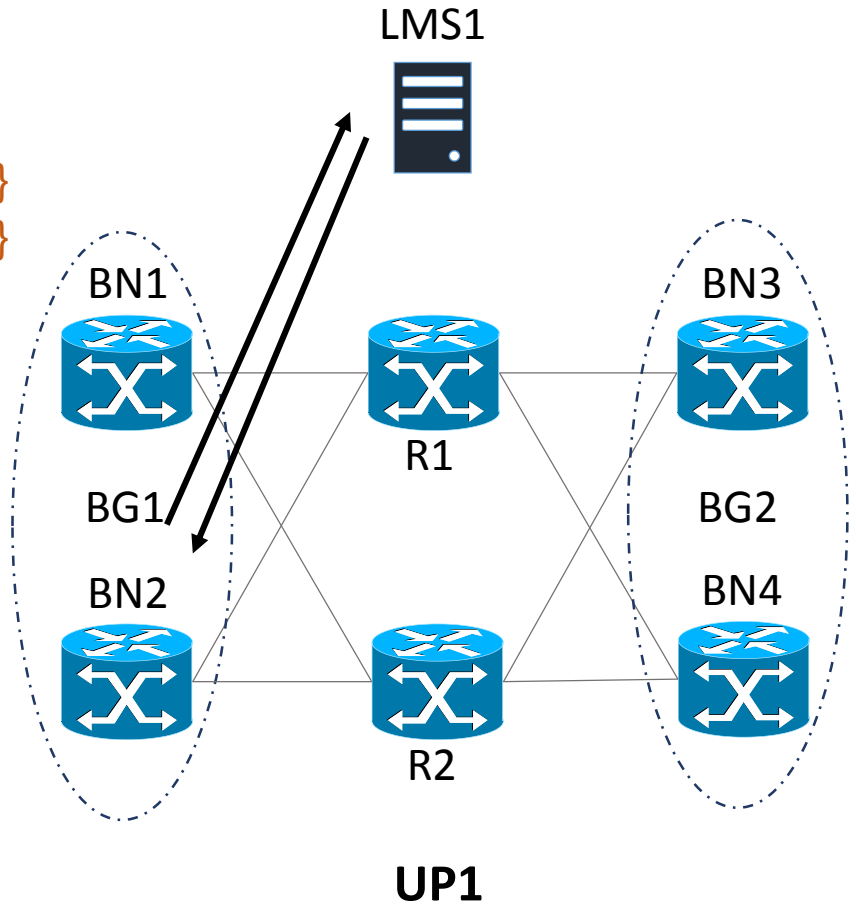
L-BGP Route:

{NLRI = BN3-Loopback:PL23, NH = BN2, Com = BG1, Ext-com = 0:UP1-context}

{NLRI = BN4-Loopback:PL24, NH = BN2, Com = BG1, Ext-com = 0:UP1-context}

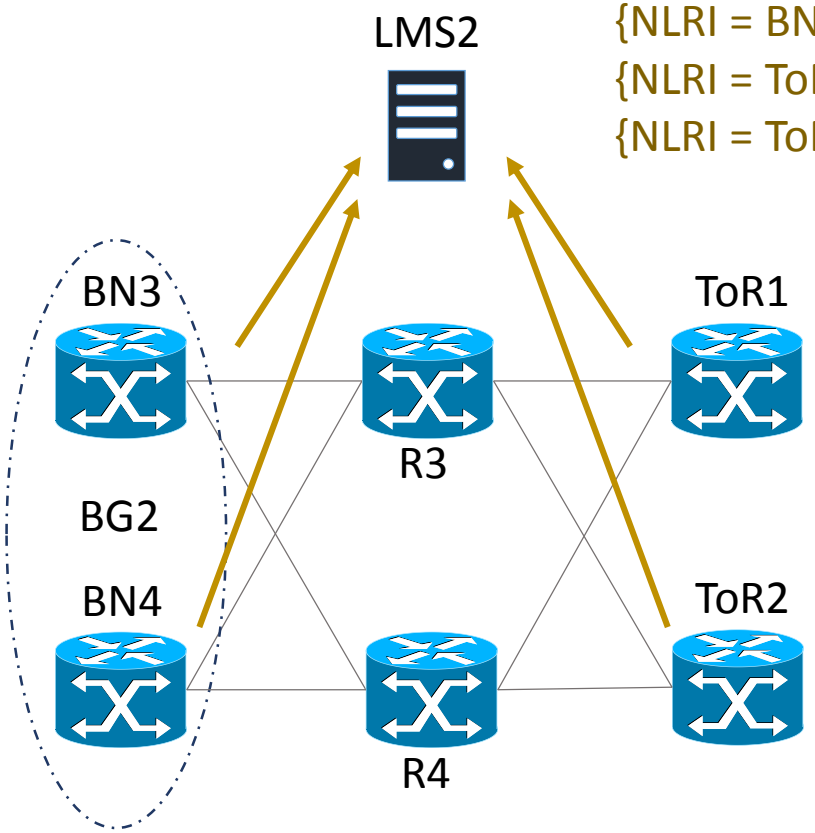
{NLRI = BN3-Loopback:PLG2, NH = BN2, Com = BG1, Ext-com = G:UP1-context}

{NLRI = BN4-Loopback:PLG2, NH = BN2, Com = BG1, Ext-com = G:UP1-context}



All BNs in UP2 Advertise Themselves to LMS2

L-BGP Routes (View of LMS2):
{NLRI = BN3-Loopback:NULL, NH = BN3, Com = BG2}
{NLRI = BN4-Loopback:NULL, NH = BN4, Com = BG2}
{NLRI = ToR1-Loopback:NULL, NH = ToR1}
{NLRI = ToR2-Loopback:NULL, NH = ToR2}

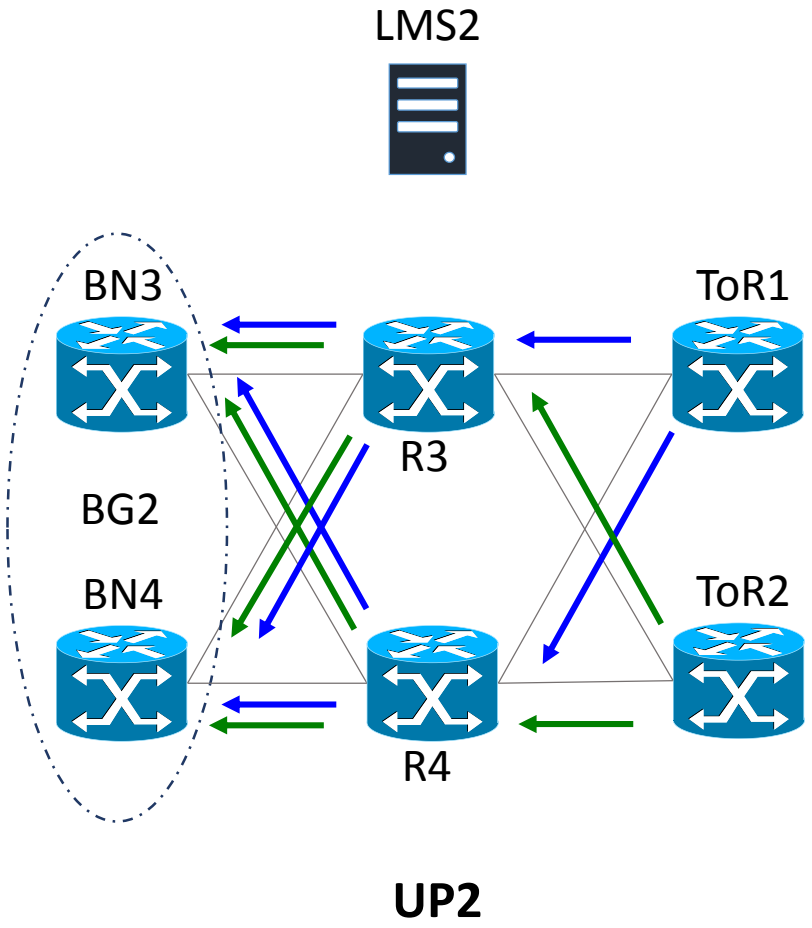


BNs of UP2 learn route to ToR1 and ToR2

L-BGP Routes (View of BN3):
{NLRI = ToR1-Loopback:L231, NH = R3}
{NLRI = ToR2-Loopback:L232, NH = R3}
{NLRI = ToR1-Loopback:L241, NH = R4}
{NLRI = ToR2-Loopback:L242, NH = R4}

ToR1 and ToR2 do not belong to any Border Groups in this example

L-BGP Routes (View of BN4):
{NLRI = BN3-Loopback:L231, NH = R3}
{NLRI = BN4-Loopback:L232, NH = R3}
{NLRI = BN3-Loopback:L241, NH = R4}
{NLRI = BN4-Loopback:L242, NH = R4}



BN3 learns partition label for ToR1 and ToR2

IP Routes (From BN3 to LMS2):

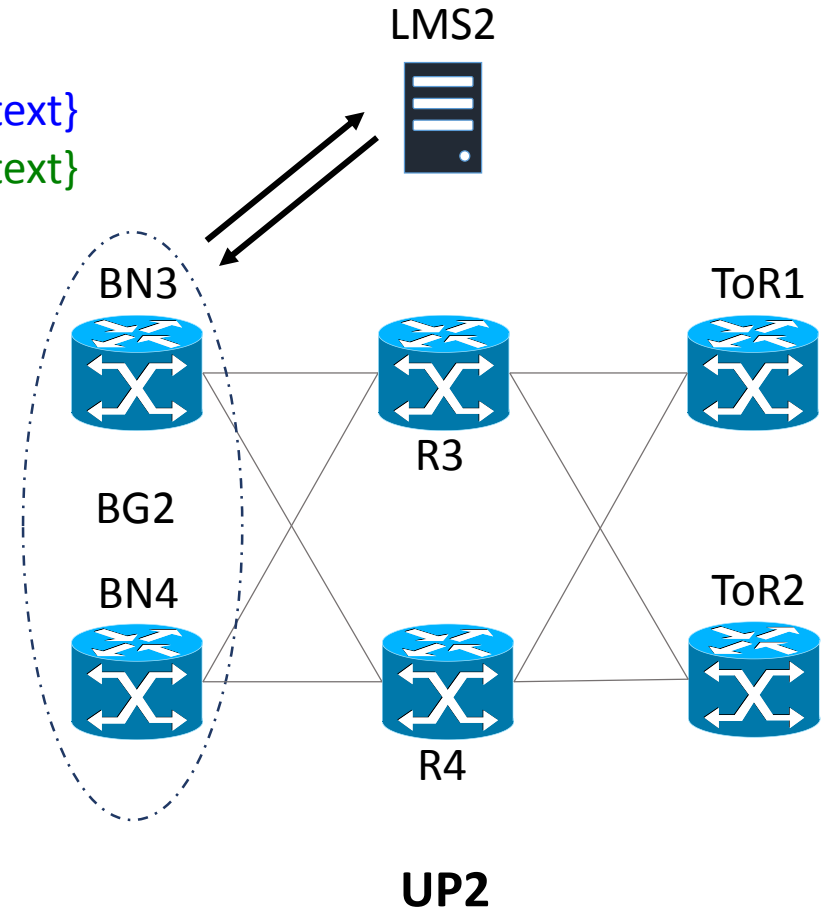
{NLRI = ToR1-Loopback, NH = BN3, Com = BG2, Ext-com = R:UP2-context}

{NLRI = ToR2-Loopback, NH = BN3, Com = BG2, Ext-com = R:UP2-context}

L-BGP Route (from LMS2 to BN3):

{NLRI = ToR1-Loopback:PL21, NH = BN3, Com = BG2, Ext-com = 0:UP1-context}

{NLRI = ToR2-Loopback:PL22, NH = BN3, Com = BG2, Ext-com = 0:UP1-context}



Summary and Next steps

- Summary:
  - Partitioning is a key aspect for scaling
  - BGP is natural glue to connect the partitions
  - New extended community allows to support underlay partition in an efficient and clean way, similar as L3VPN, and supports brownfield deployment well
  - BGP is used as protocol to request and learn the operator assigned labels
  - The procedure defined here can be used for any partition technology
- Next Steps
  - Gather feedback and welcome contributions from the working group
  - Asking for working group adoption after further revision