SPRING

IETF-94

Tuesday, November 3

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.

Document Status

- Documents Adopted
  - draft-ietf-spring-segment-routing-msdc
  - draft-ietf-spring-segment-routing-central-epe
  - draft-ietf-spring-segment-routing-ldp-interop
  - draft-ietf-spring-oam-usecase
- WGLC Completed
  - draft-ietf-spring-segment-routing

Document Status

- Submitted to IESG
  - draft-ietf-spring-problem-statement-05

Agenda

- Administrivia10 minutes
  - Chairs
- Update on WG drafts10 minutes
  - Roberta Maglione / Stefano Previdi
- Anycast Segments in MPLS based SPRING 20 minutes
  - Pushpasis Sarkar draft-psarkar-spring-mpls-anycast-segments-01
- Addressing SID conflicts 15 minutes
  - Les Ginsberg draft-ginsberg-spring-conflict-resolution-00
- Advertising Per-Algorithm Label Blocks 15 minutes
  - Chris Bowers draft-bowers-spring-adv-per-algorithm-label-blocks-02
- Interconnecting Millions Of Endpoints With Segment Routing 10-15 minutes
  - Dennis Cai draft-filsfils-spring-large-scale-interconnect
- Tunnel Segment in Segment Routing 10 minutes
  - Robin Li draft-li-spring-tunnel-segment-00

SPRING WG

Segment Routing Drafts Update

sprevidi@cisco.com

IETF94 – Yokohama, November 2015

SPRING WG drafts

- draft-ietf-spring-problem-statement
  - Version 05
  - Fixed IP addresses in illustration

SPRING WG drafts

- draft-ietf-spring-segment-routing
  - Version 06
  - Added clarification text about SIDs allocation
    - Domain, Topology, Algorithm
  - Added clarification about Anycast SID
  - Made author's list compliant with IETF rules
  - Integrated multiple comments received from mailing list
    - Missing "SR Domain" definition (in next revision)
  - Fixed IP addresses in illustrations

SPRING WG drafts

- draft-ietf-spring-segment-routing-mpls
  - Version 02
  - Fixed typo's

SPRING WG drafts

- draft-filsfils-spring-segment-routing-ldp-interop
- Now WG item:
  - draft-ietf-spring-segment-routing-ldp-interop
  - Version 00
  - Received comments from Sasha Vainshtein that will be integrated in next revision (work in progress)

SPRING WG drafts

- draft-filsfils-spring-segment-routing-msdc
- Now WG item:
  - draft-ietf-spring-segment-routing-msdc
  - Version 00
  - Fixed author's list according to IETF rules

SPRING WG drafts

- draft-filsfils-spring-segment-routing-central-epe
- Now WG item:
  - draft-ietf-spring-segment-routing-central-epe
  - Version 00
  - Fixed author's list according to IETF rules
  - Fixed IP addresses in illustrations

New Draft

- draft-filsfils-spring-sr-recursive-info
  - Defines a mechanism allowing a prefix to be resolved to a SID allocated to a different node
  - Addresses multiple use cases
    - Multiple loopback addresses associated the the same node
    - Multiple local services attached to the same node

Moved Draft

- draft-francois-spring-segment-routing-ti-lfa
  - Moved to RTGWG
  - Now draft-francois-rtgwg-segment-routing-ti-lfa

Anycast Prefix Segments in MPLS-based SPRING

draft-psarkar-spring-mpls-anycast-segments-01

Pushpasis Sarkar psarkar@juniper.net
Hannes Gredler hannes@gredler.at
Clarence Filsfils cfilsfils@cisco.com
Stefano Previdi sprevidi@cisco.com
Bruno Decraene bruno.decraene@orange.com
Martin Horneffer Martin.Horneffer@telekom.de

Summary

- Additional Contributors
- Proposed Solution – Update
  - Terminologies
  - Procedures

Additional Contributors

Clarence Filsfils cfilsfils@cisco.com

Stefano Previdi sprevidi@cisco.com

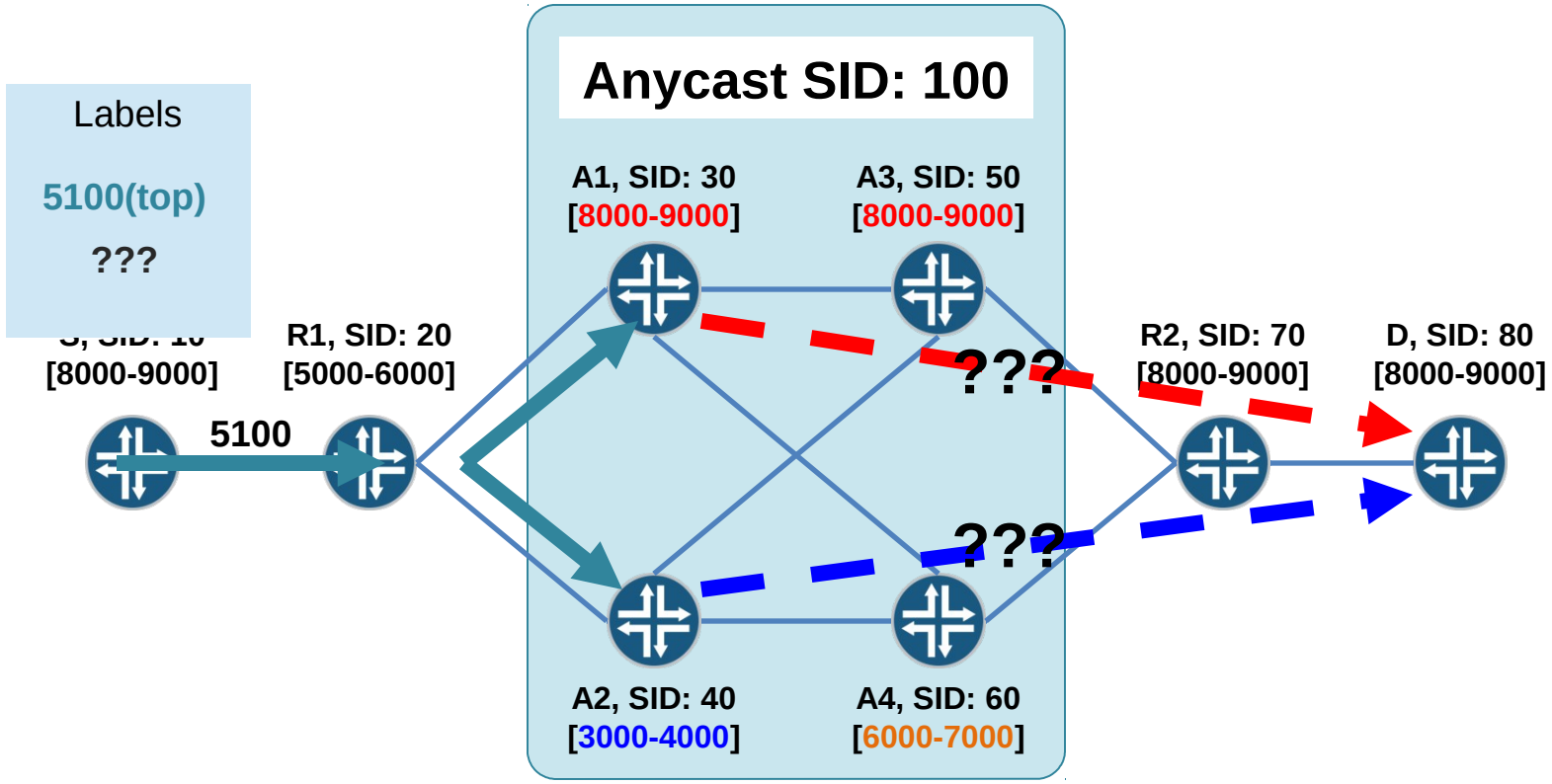
Bruno Decraene bruno.decraene@orange.com

Martin Horneffer Martin.Horneffer@telekom.de

More Definitions

- Common Anycast SRGB (CA-SRGB)
  - Identifies the **SRGB implemented by majority of the network devices** participating in one or more anycast group(s).
  - All devices in network MUST allow operator to set it
    - When set,
      - The **operator should set the same value on all devices**.
      - The device **need not allocate the same range for the local SRGB**. The CA-SRGB may or may not be same as the local SRGB.
    - If not set explicitly, the CA-SRGB should be assumed to be same as the local SRGB.

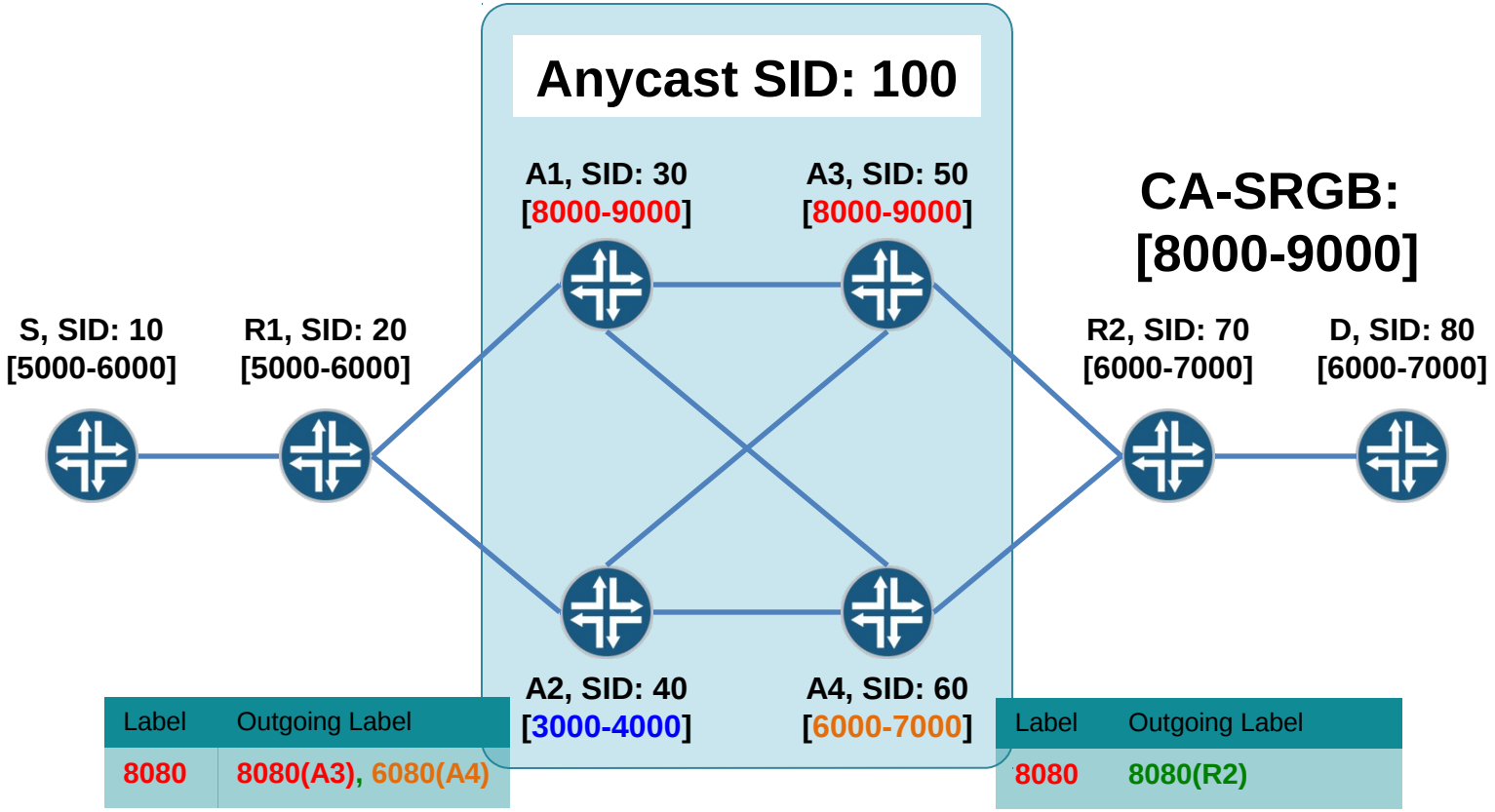
Problem Statement



How to compute the label that represents the next segment?

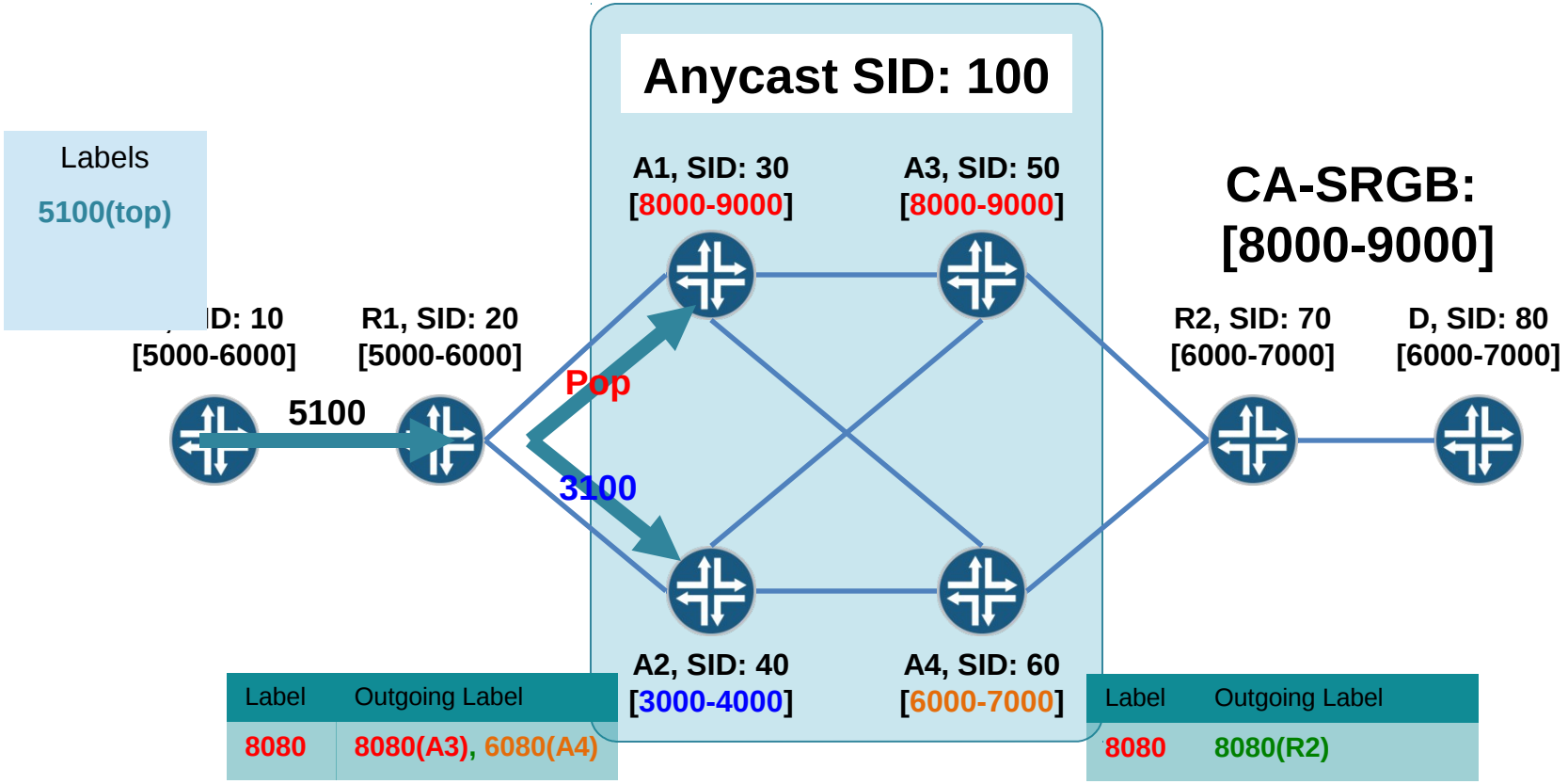
Proposed Solution

- Step 1: Devices originating any anycast prefix segments ***that does not have same local SRGB as the CA-SRGB***
  - Create a Virtual L-FIB lookup table
    - Map all remotely learnt node/anycast prefix segment index to corresponding downstream label and next-hop.



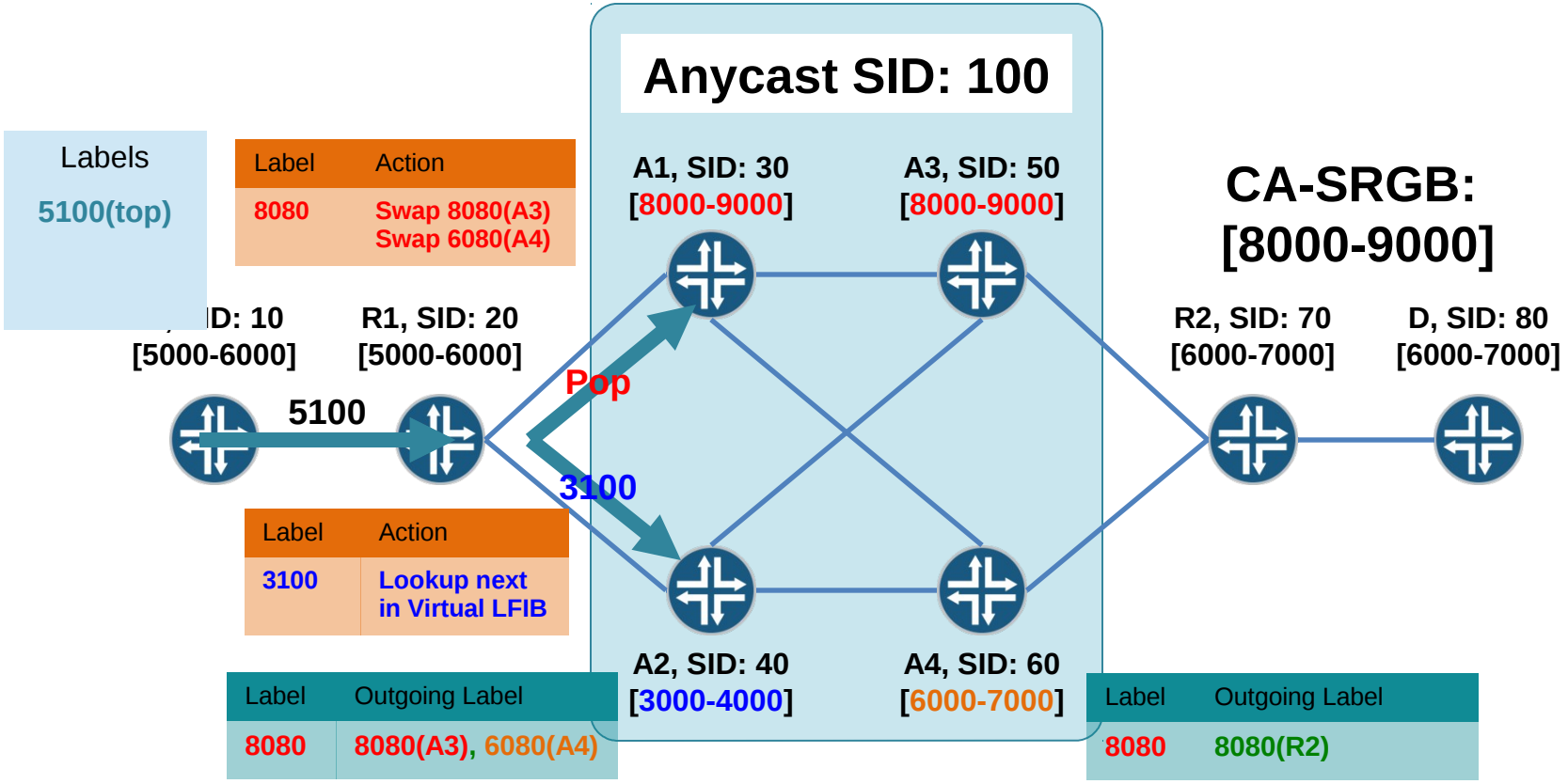
Proposed Solution

- Step 2: Devices originating any anycast prefix segments ***that does not have same local SRGB as the CA-SRGB***
  - **Originate IGP advertisement** for anycast prefix SID with (**No-PHP =1 and Exp-Null = 0**).
  - Ensures the packet arrives with anycast prefix segment label allocated for it. **Penultimate-hop does not POP** the label, but replaces it.



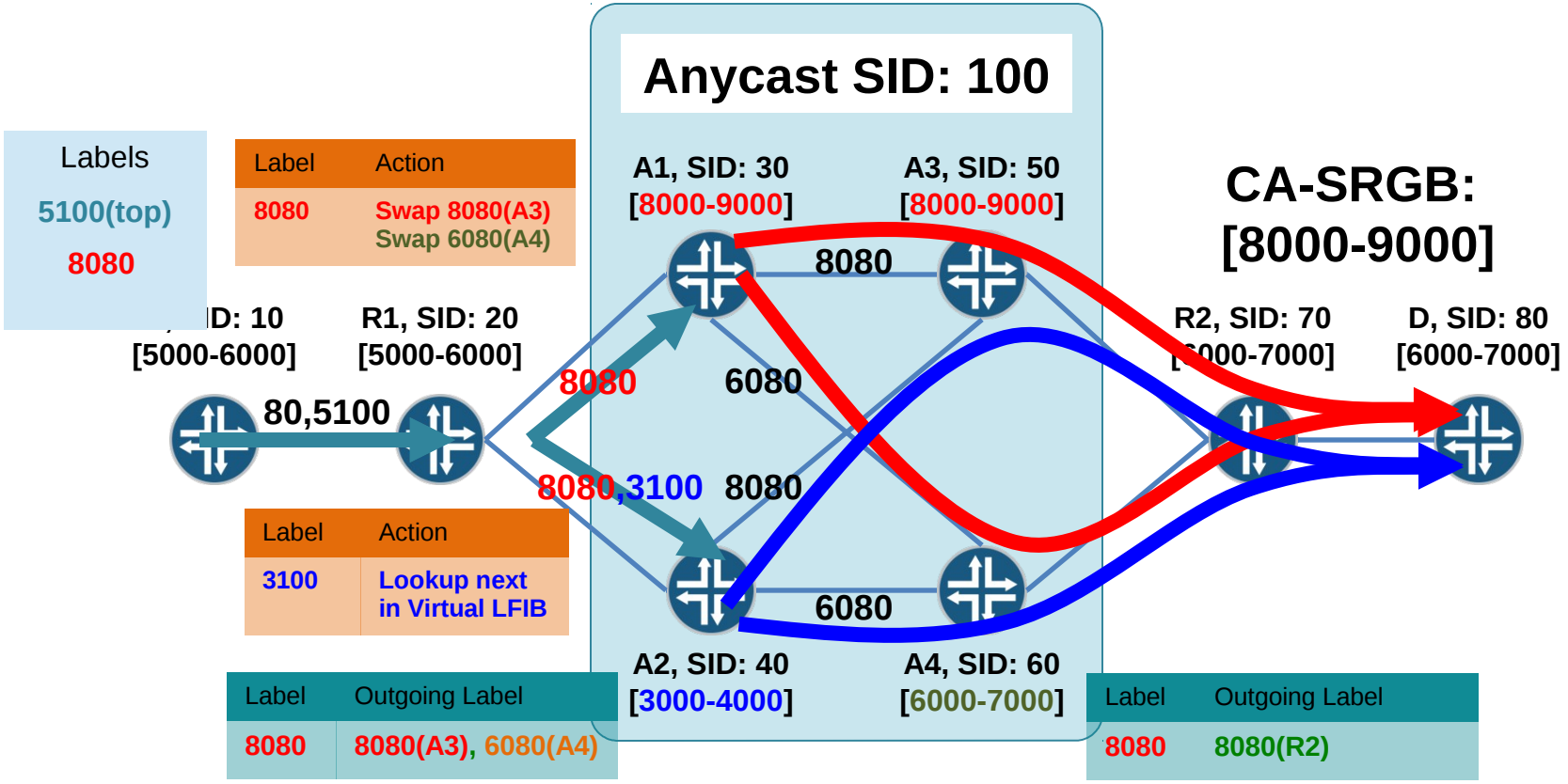
Proposed Solution

- Step 3: Devices originating any anycast prefix segments ***that does not have same local SRGB as the CA-SRGB***
  - **For the anycast segment label in the global LFIB table**.
    - Install a **Lookup into the Virtual LFIB** created in Step 1.



Proposed Solution

- Step 4: Ingress device using Anycast prefix segments
  - For the **prefix segment next that follows a anycast prefix segment**.
    - **Use the *prefix segment index as offset into CA-SRGB range*** to compute the label to be used.



Next Steps

- Comments/Questions/Suggestions ?
- WG Adoption.

THANK YOU

Segment Routing Conflict Resolution

draft-ginsberg-spring-conflict-resolution-00

Les Ginsberg (ginsberg@cisco.com)
Stefano Previdi (sprevidi@cisco.com)
Peter Psenak (ppsenak@cisco.com)

WHY?

For identifiers w global scope/usage conflicts may occur due to misconfiguration. This will cause forwarding issues (drops, loops).

Consistent (network-wide) and deterministic conflict resolution policy is needed to minimize the damage.

Prior discussions have not reached consensus.

Draft is the vehicle to drive discussions to consensus and document the agreed upon policies.

Problem is cross-protocol – therefore SPRING is the right working group.

Handling Invalid SRGB Entries

Example:

Range 1: (100, 199]

Range 2: (1000, 1099)

Range 3: (100, 599) !Overlaps w Range #1

Range 4: (2000, 2099)

Draft Proposal: Use ranges preceding the first conflicting range.

- **Presumes the most common case is misconfiguring a new range placed at the end of the advertisement.**

Alternate Proposal: Reject the entire advertisement

- **Places burden on the source to detect/prevent misconfigurations**
- **Represents a bug in the config validation – any sort of error is possible**

Mapping Entry

A generalized mapping entry can be represented using the following definitions:

Pi - Initial prefix
Pe - End prefix
L - Prefix length
Lx - Maximum prefix length (32 for IPv4, 128 for IPv6)
Si - Initial SID value
Se - End SID value
R - Range value

Mapping Entry is then the tuple: (Pi/L, Si, R)
Pe = (Pi + ((R-1) << (Lx-L))
Se = Si + (R-1)

Terminology: Conflict Types

PREFIX CONFLICT

When different SIDs are assigned to the same prefix

(192.0.2.120/32, 200, 1)

(192.0.2.120/32, 30, 1)

SID CONFLICT:

When the same SID has been assigned to multiple prefixes

(192.0.2.1/32, 200, 1)

(192.0.2.222/32, 200,1)

Mapping Entry Conflict Resolution

Policy	Advantages	Disadvantages
Ignore	Simple and predictable Easy to diagnose No unintended traffic flow	Delivery to all destinations in conflict is compromised
Preference Rule	Traffic to some of the destinations in conflict may continue to be forwarded successfully	Harder to diagnose based on forwarding behavior Introduction of new conflicts may cause other entries in conflict to be used

(Draft is currently agnostic)

Preference Rule Exclusions

Source (e.g. originating router-id) should NOT be used – not known consistently when advertisements are leaked between areas

Route type (e.g. intra-area vs inter-area) should NOT be used as it is not consistent network-wide

Prefix advertisements vs SRMS advertisements:

- **Equally vulnerable to misconfiguration**
- **Could be configured in a similar manner**

Context

When conflicts occur it is impossible to know which advertisement is the one intended by the operator.

No matter what policy is chosen we cannot guarantee all traffic will be delivered correctly.

Caused by a misconfiguration – network is broken!!

Any choice will be optimal in some deployments and sub-optimal in other deployments

Any choice will be optimal in some types of misconfigurations and sub-optimal in other types of misconfigurations

Priorities

Detect the problem

Report the problem

Define consistent behavior

Don’t overengineer

Choose the resolution behavior

Next Steps

Open discussion

Come to consensus quickly ☺

WG adoption Requested

Advertising Per-Topology and Per-Algorithm Label Blocks

draft-bowers-spring-adv-per-algorithm-label-blocks-02

Chris Bowers cbowers@juniper.net
Pushpasis Sarkar psarkar@juniper.net
Hannes Gredler hannes@juniper.net
Uma Chunduri uma.chunduri@ericsson.com

SPRING Working Group
IETF94 Yokohama

Computing locally significant labels for shortest path next-hops

SPF_Label(X,D) = Label_Block(X) + Node_Index(D)

- Label_Block(X) is the label block advertised by X
- D is the destination node
- SPF_Label(X,D) is the value of the label that neighbors need to apply to a packet so that X will forward the packet along the shortest path next-hop to D.

Computing locally significant labels for next-hops corresponding to other topologies and algorithms

Option 1: per-topology / per-algorithm node index

Label(X,D,T,A) = Label_Block(X) + Node_Index(D,T,A)

Option 2: per-topology / per-algorithm label block

Label(X,D,T,A) = Label_Block(X,T,A) + Node_Index(D)

- T is the topology
- A is the algorithm for computing destination-based forwarding next-hops
- D is the destination node
- X is the next hop along the path to D that is determined by algorithm A for topology T
- Label(X,D,T A) is the value of the label that neighbors need to apply to a packet so that X will forward the packet along the next-hop to D determined by algorithm A for topology T.

Option 2: per-topology / per-algorithm label block

label blocks

100-109	0	1	2	3	4	5	6			
110-119	0	1	2	3	4	5	6			
120-129	0	1	2	3	4	5	6			
130-139	0	1	2	3	4	5	6			
140-149										

With option 2, label values must be logically related to the topology/algorithm pair and the node SID.

Option 1: per-topology / per-algorithm node index

label block

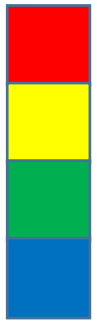
100-149	0	1	2	3	4	5	6			
	0		2	4		5			2	
	0	1	3	5	4	3				5
	1		6		4		6			
		1	2	3	6		0			

T=0,A=0

T=0,A=4

T=0,A=5

T=7,A=1



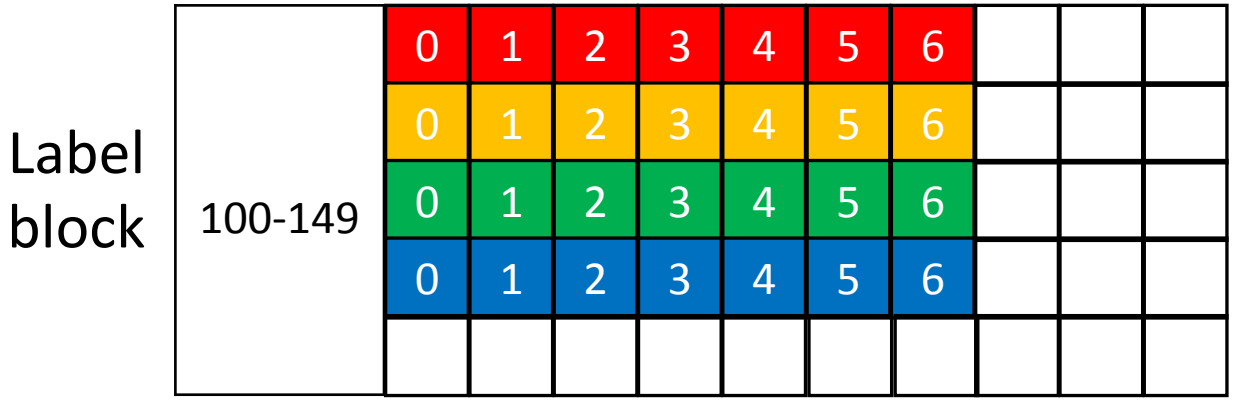
With option1, label values do not need to have any logical relationship to the topology/algorithm pair and the default topology/algorithm node SID.

node#

#

Organization of label values needs to be imposed externally.

Option 1: per-topology / per-algorithm node index plus the configured offset mapping method



Config on node#3
base_node_index=3

label_block_size=50
topology=0 algorithm=0 offset=0
topology=0 algorithm=4 offset=10
topology=0 algorithm=5 offset=20
topology=7 algorithm=1 offset=30

Config on node#4
base_node_index=4

label_block_size=50
topology=0 algorithm=0 offset=0
topology=0 algorithm=4 offset=10
topology=0 algorithm=5 offset=20
topology=7 algorithm=1 offset=30

Config on node#4
base_node_index=5

label_block_size=50
topology=0 algorithm=0 offset=0
topology=0 algorithm=4 offset=10
topology=0 algorithm=5 offset=20
topology=7 algorithm=1 offset=30

Identically configured offset mappings via CLI (or management interface) can be used to impose order on the label assignment

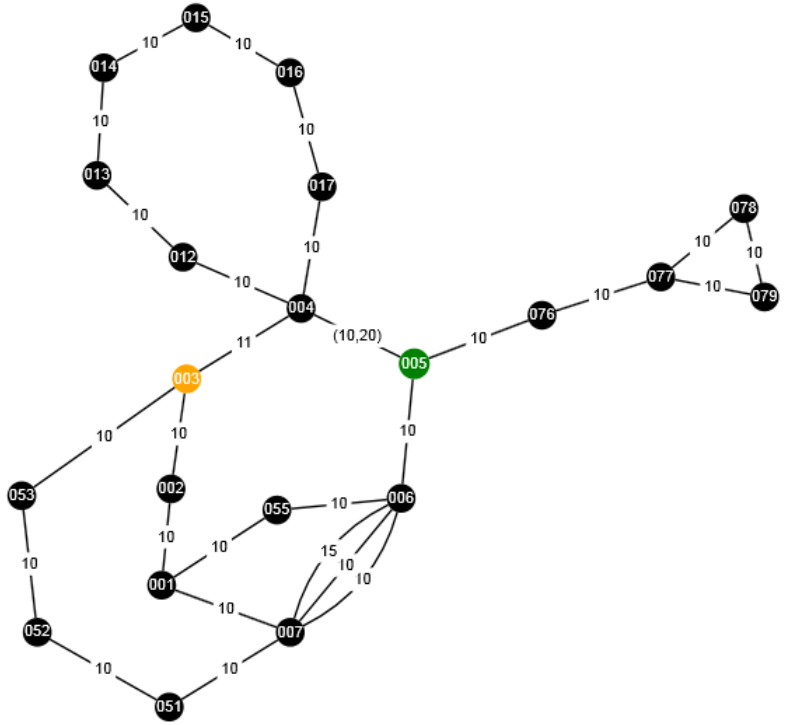
Why is option 2 (per-topo/per-algo label blocks) better than option 1 (per-topo/per-algo node indexes)?

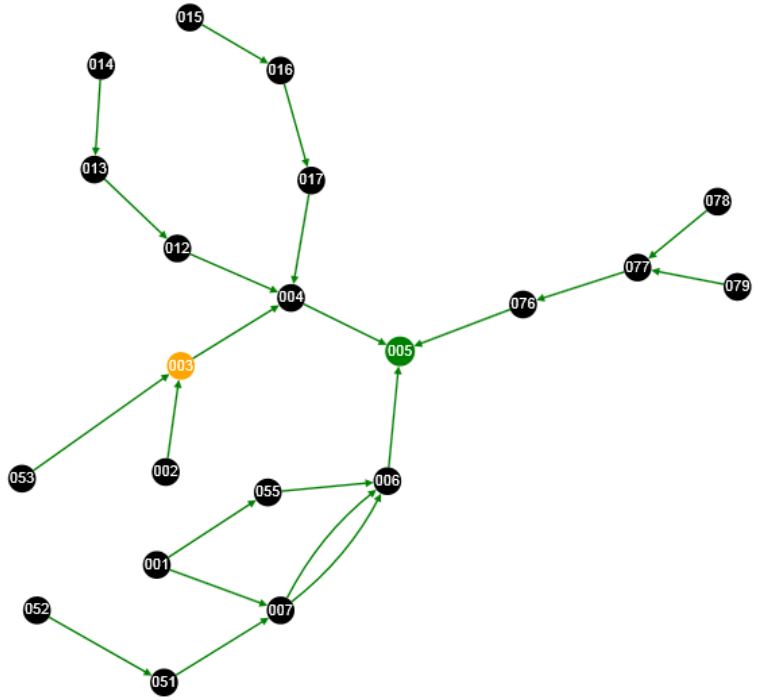
- Using SR to distribute labels for shortest path routes
- Advantage over LDP that any node (not just neighbors of X) can determine the FEC-label bindings distributed by node X.
  - **Good thing**: no need for targeted LDP sessions
  - **Bad thing** : need to assign and maintain tightly packed, domain-unique node index values

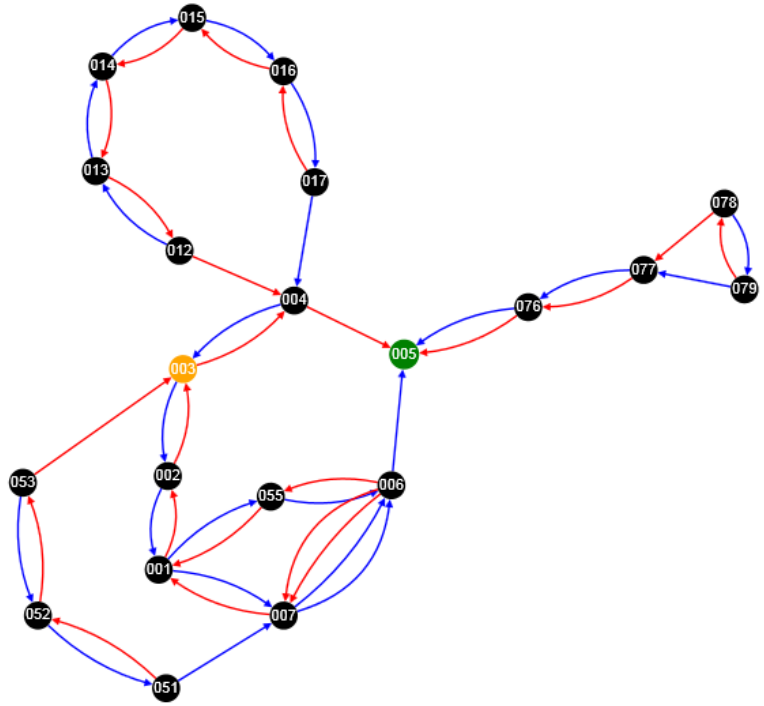
- Generalizing SR to distribute labels for other topologies and algorithms
  - Option 1 allows the **Bad Thing** to propagate.
    - Configured offset mappings required to manage the badness
    - Extra configuration adding no value and error prone
  - Option 2 puts the **Bad Thing** in a box.

Algorithms and topologies:

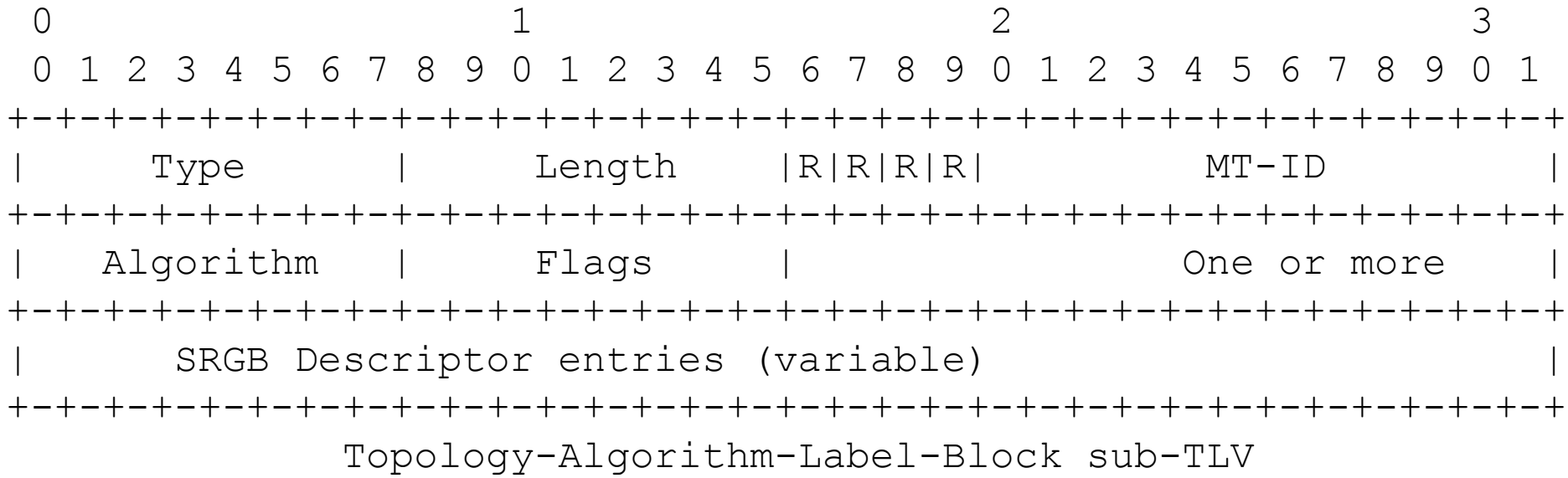
- Simple example in draft uses shortest path based on latency as alternative algorithm.
- SR algorithm mechanism seems like an elegant method for distributing labels for maximally redundant trees (MRT)
  - eg. MRT-Red next-hops = algo#4 , MRT-Blue next-hops = algo#5
  - Single label signifying destination prefix and algorithm (label stack depth = 1)
- Complexity of managing per-algorithm node-SID assignment makes it less attractive.







Proposed ISIS extension to support per-topology/per-algorithm label blocks



Topology-Algorithm-Label-Block sub-TLV

- Same format as SR-Capabilities sub-TLV for specifying label block, plus topology and algorithm field.
- Backward compatible:
  - MT-ID = 0, Algorithm=0 label block is only carried by the SR Capabilities sub-TLV.
  - Topology-Algorithm-Label-Block sub-TLV only carries MT-ID and algorithm value pairs where at least one is non-zero

Interconnecting Millions Of Endpoints with Segment Routing

draft-filsfils-spring-large-scale-interconnect-01

C. Filsfils, D. Cai , S. Previdi, Cisco
W. Henderickx, Alcatel-Lucent
R. Shakir, BT
D. Cooper, F. Ferguson, Level3
T. LaBerge, S. Lin, Microsoft
B. Decraene, Orange
L. Jalil, Verizon
J. Tantsura, Ericsson

IETF 94 SPRING, November 2015, Yokohama

What’s new since version 00?

- Very simple, no much change
- Add new co-author
  - Jeff Tantsura, Ericsson

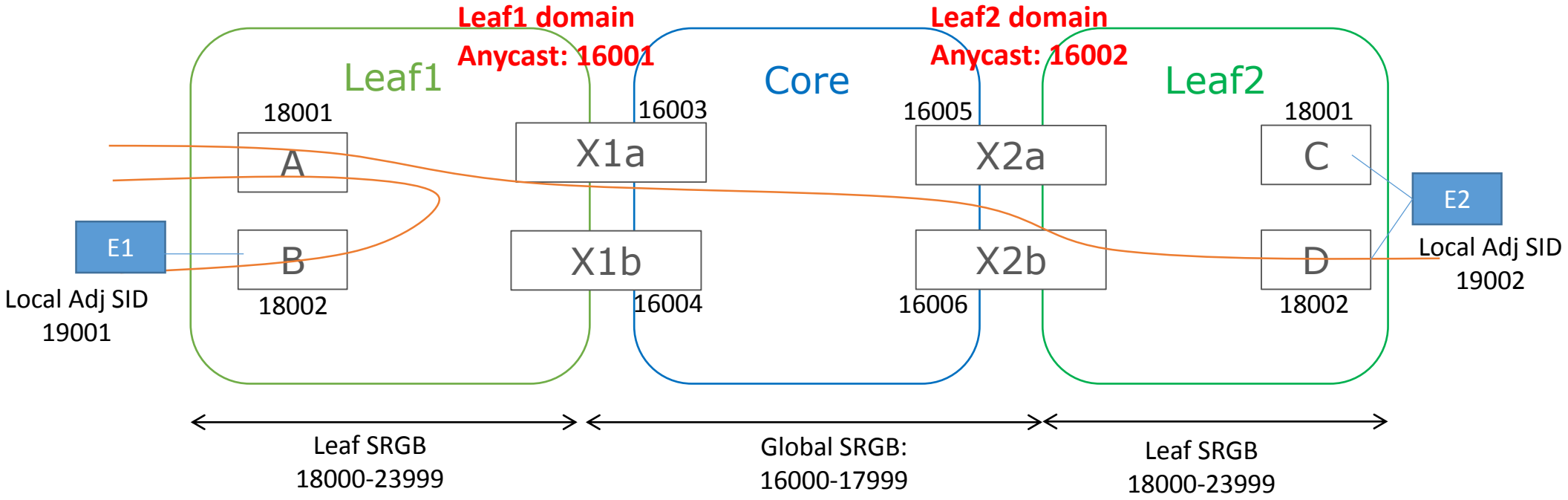
Problem Statement Re-cap

- Not like IP, there is no such concept of the label summarization and default label
- For the MPLS network, each node need specific label for the forwarding
- Millions of nodes/endpoints means millions of RIB/FIB, which is not desirable for the low cost DC switches or SP metro access nodes

The Principle and Reference Design

Using hierarchical label stack to solve large scale MPLS network

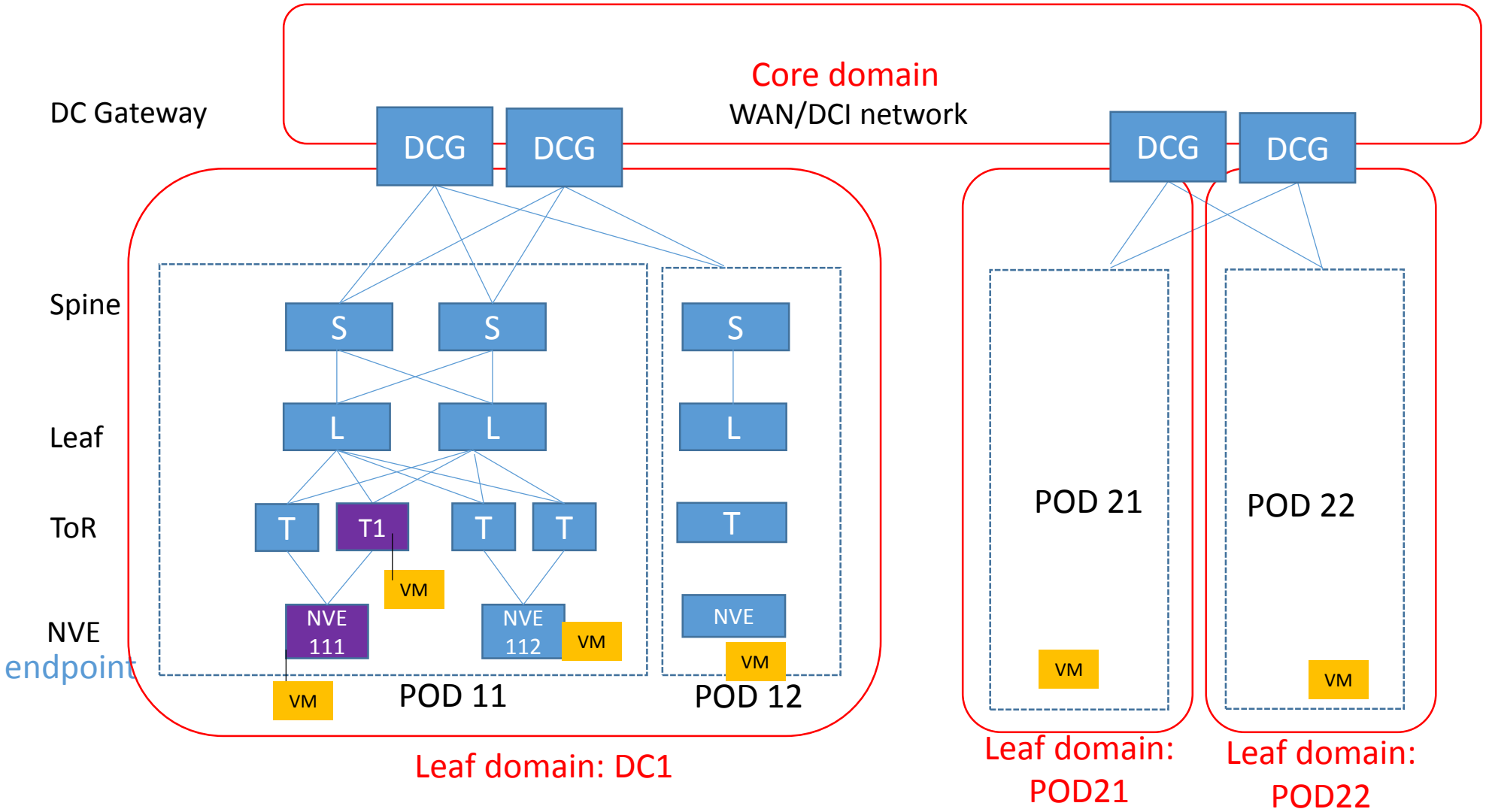
- Network is divided into 2 or 3 layers: core, leaf and sub-leaf (or local endpoint) optionally
- Each leaf domain is reachable via domain label (thinking zip code). Domain label is anycast SID which is advertised by the leaf border routers
- Endpoint use local adj SID which is behind one or multiple leaf nodes



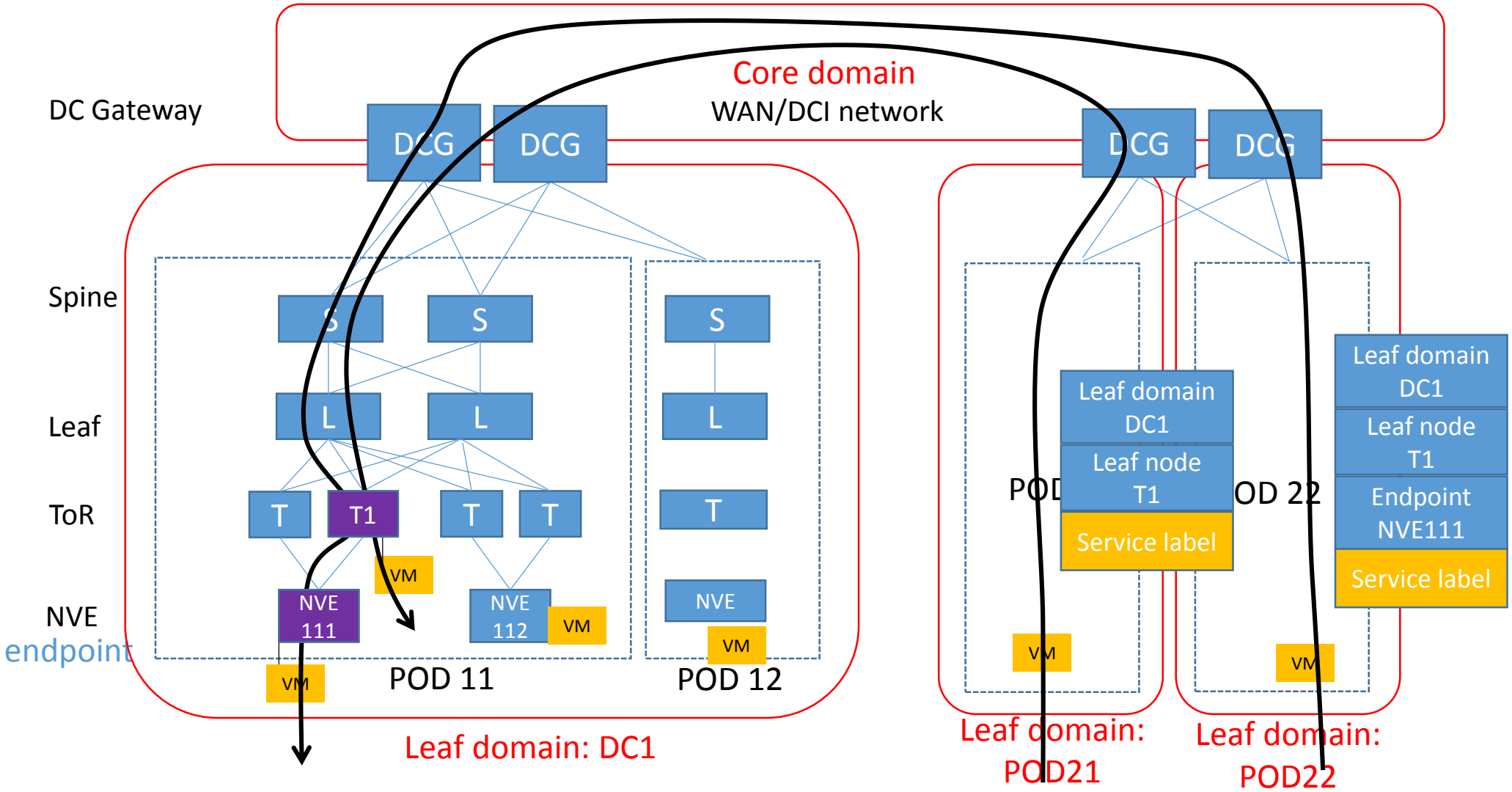
Intra-leaf (A→B): shortest-path {18002}
Inter-leaf (A →D): shortest-path via X {16002,18002}

Inter-leaf (E1→E2): shortest-path via X {16002,18002, 19002}

Large-scale DC Network Use Case

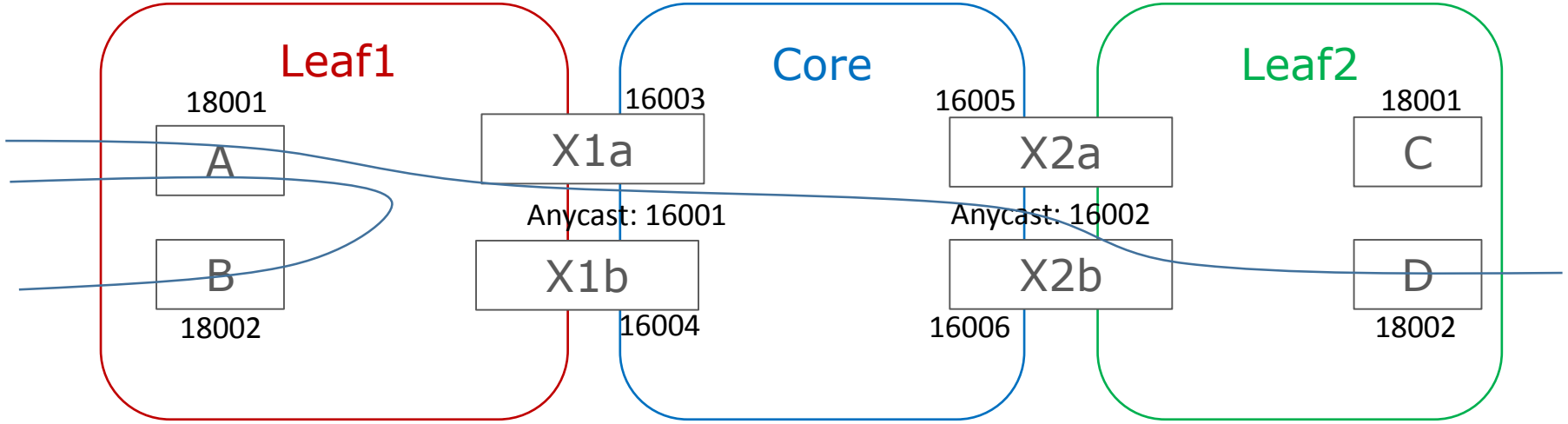


Large-scale DC Network Use Case



Benefit

- A simple way to scale MPLS network using existing SR, no protocol changes
- Fully leverage the distributed SR in each domain: ECMPs, TI-FRR and SR-TE
- Simple “X” node (border node) redundancy using anycast SID
- Fully inter-operate with existing network protocols and design: LDP, seamless MPLS



Questions/Comments?

Tunnel Segment in Segment Routing

draft-li-spring-tunnel-segment-00

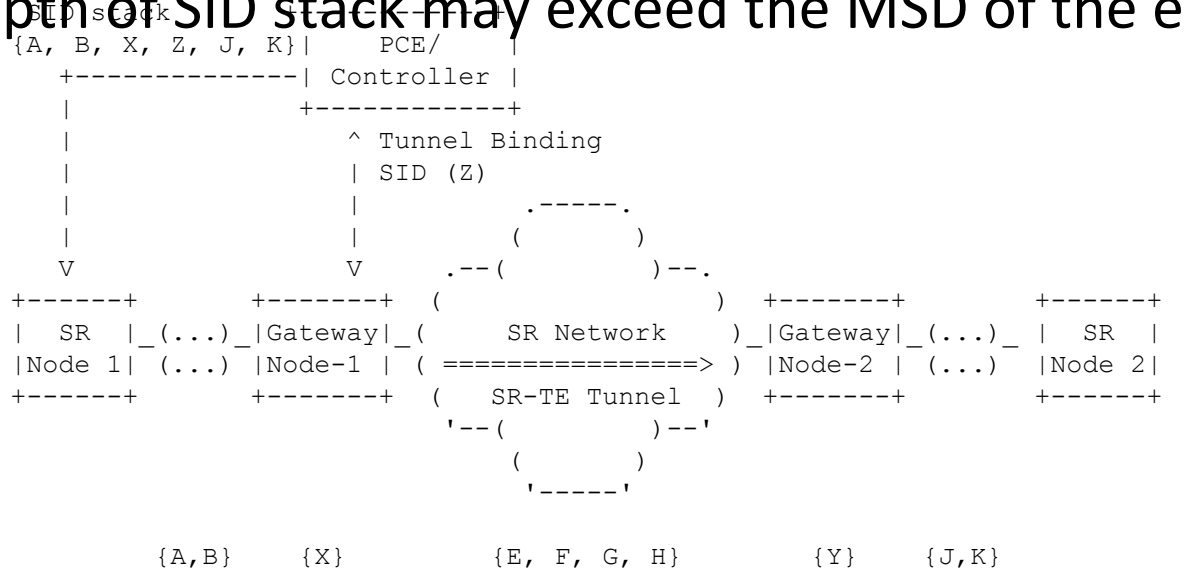
Robin Li(lizhenbin@huawei.com)
Eric Wu(eric.wu@huawei.com)

IETF94, Yokohama

Use cases (1)

☐Reducing SID Stack Depth

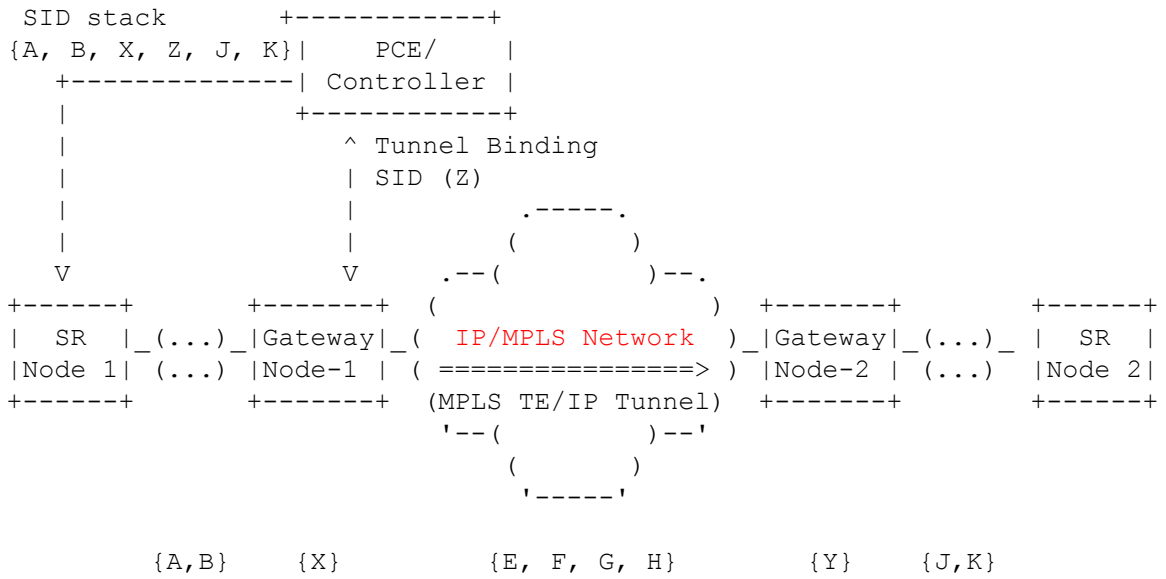
- An explicit path expressed in Binding-SID may require multiple TLV instances since no guarantee for continuous IP addresses.
- The depth of SID stack may exceed the MSD of the explicit path.



Use cases (2)

☐Passing through Non-SR Domain

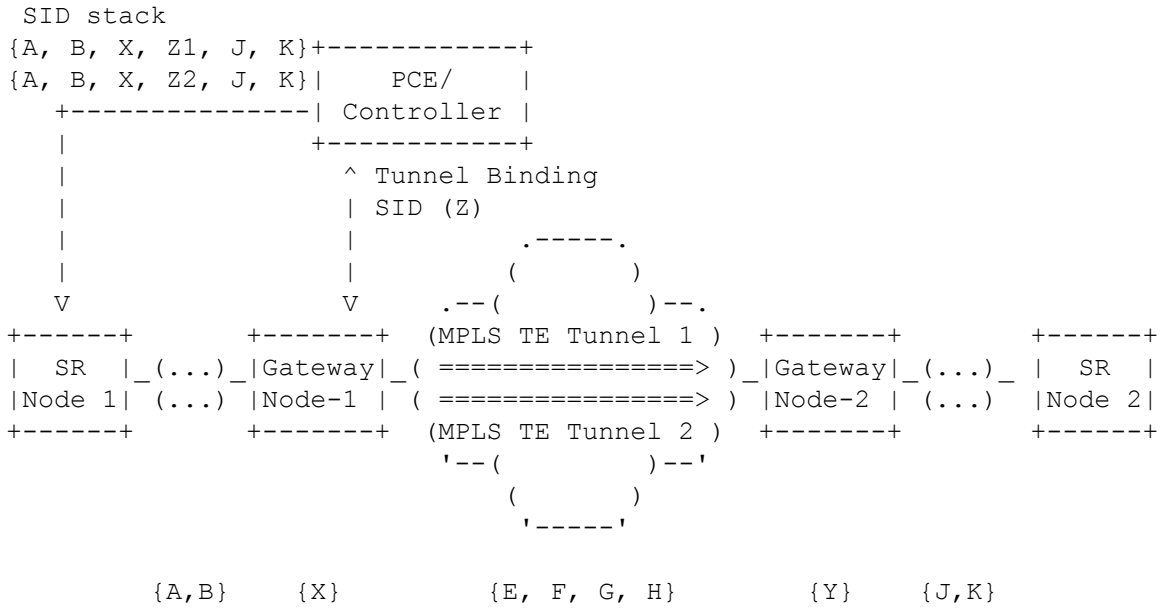
- Even MSD is not an issue, the network a tunnel passed through can be non-SR-capable, so steering by SIDs is not going to happen.



Use cases (3)

☐Differentiated Services

- Multiple tunnels between the same pair of gateway nodes to support different services though the explicit path is same.



Creating and binding

Manual

- Creating and binding a tunnel to its SID manually. Then several signaling ways can be used to propagate binding information: IGP/PCEP/BGP-LS

Centralized

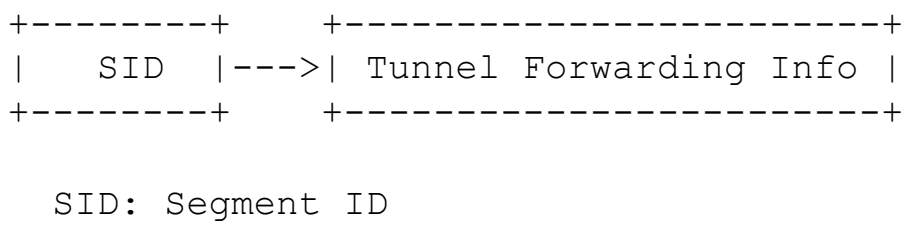
- A tunnel initialized and propagate binding relationship through PCEP extensions (Refer to draft-ietf-pce-pce-initiated-lsp and draft-zhao-pce-central-controller-user-cases).

Comparison with Adjacency Segment

- It may be necessary to differentiate a tunnel segment from other adjacency segment in some scenarios since there are more attributes attached to a tunnel.
- Not only to inform the binding relationship between a tunnel and a SID but also to learn tunnel information as much as possible
- IGP Adjacency will need an IP(a borrowed one at least) while a Tunnel-SID won't.

Forwarding Mechanism

- In the gateway node, when received the packet with the tunnel segment SID as the topmost SID, it will use the forwarding mechanism shown in the following figure to steering the traffic to the corresponding tunnel



Requirements of Control Plane and Yang Models

- REQ 01/02/03: IGP/BGP-LS/PCE extensions SHOULD be introduced to advertise the binding relationship between a SID/label and the corresponding tunnel. Attributes of the tunnel MAY be carried optionally.

- REQ 04: PCE SHOULD support initiating IP tunnel.

- REQ 05: PCE SHOULD support to allocate SID/label for the corresponding tunnel dynamically.

- REQ 06: PCEP extensions SHOULD be introduced to distribute the binding relationship between a SID/label and the corresponding tunnel from a PCE to a PCC. Attributes of the tunnel MAY be carried optionally.

- REQ 07: An I2RS interface SHOULD be available for allocating SID/label to the corresponding tunnel. And augmentation on segment routing YANG models SHOULD be introduced.

Next step

- ☐ Collect feedback and comments.
- ☐ Refine this draft according to comments.