



SPRING -Advanced Use Cases

Aditya Kaul – Solution Architect



LEGAL STATEMENT

The information contained in this presentation is confidential to Juniper Networks and is disclosed under conditions of confidentiality

This presentation contains information relating to Juniper Networks' development plans and plans for future products, features or enhancements ("SOPD"). SOPD information is subject to change at any time, without notice. Except as may be set forth in definitive agreements for the potential transaction, Juniper Networks provides no assurances, and assumes no responsibility, that such future products, features or enhancements will be introduced.

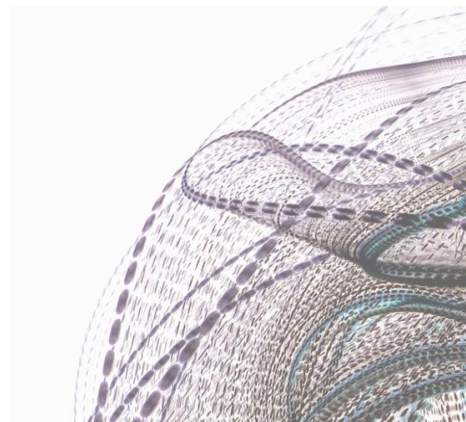
Customers must ensure that purchasing decisions:

- a) are not being made based upon reliance of timeframes or specifics outlined in the SOPD; and
- b) would not be affected if Juniper Networks delays or never introduces the future products, features or enhancements.



AGENDA

- Introduction
- SR Traffic Engineering (SR-TE)
- Flex-Algo
- References

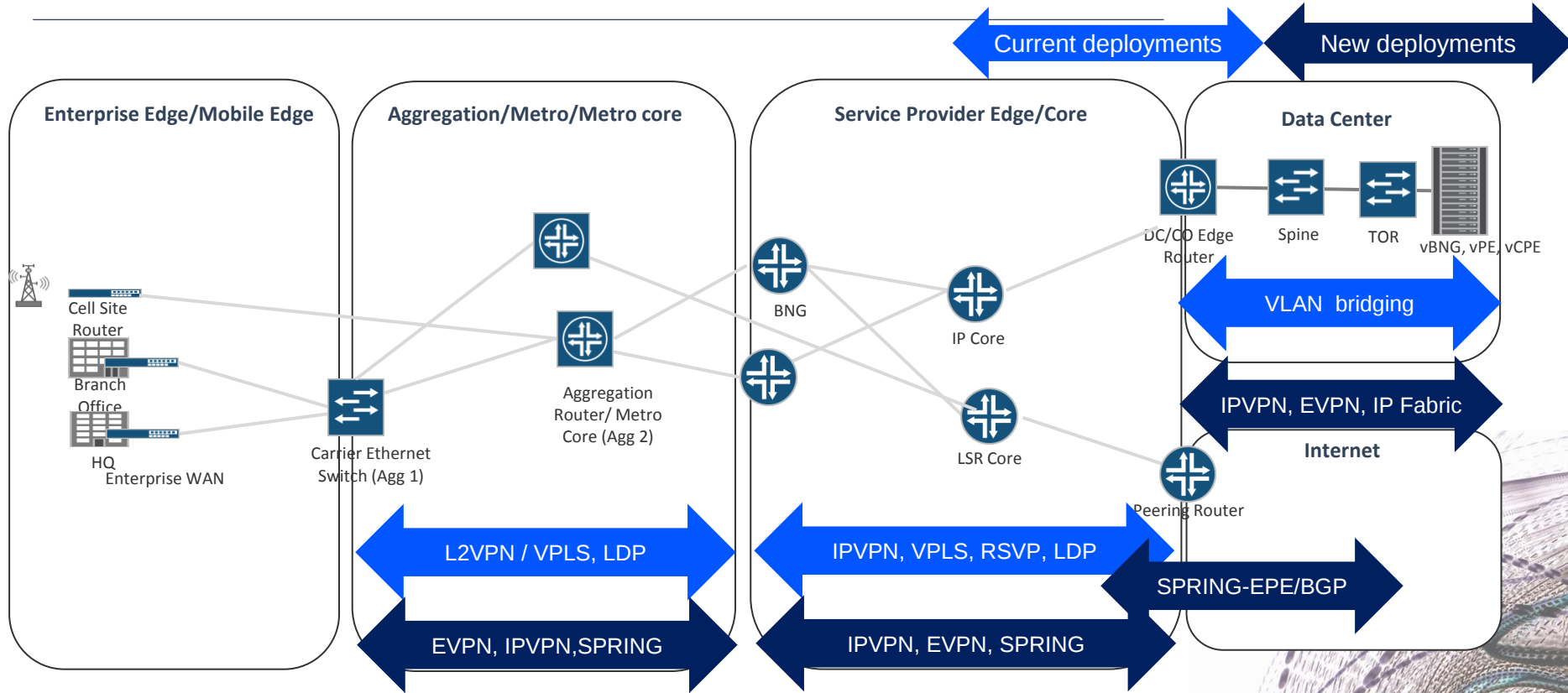


Assumptions

- Working knowledge of Segment Routing (SPRING/MPLS mostly)
 - Forwarding Mechanism
 - Segment ID (Node, Adj, Binding, Anycast)
- Working knowledge of MPLS based traffic engineering
- A basic understanding of Spring Controller and path computation (PCE, PCC)



ROUTING ARCHITECTURES EVOLUTION



Leverage single *Protocol* across all domains.

Introduction

- SPRING a.k.a. Segment Routing
 - Source Packet Routing in Networking
 - Leverages the source routing paradigm; a node steers a packet through an ordered list of instructions, called segments
 - A segment can represent any instruction, topological or service-based
 - Allows to enforce a flow through any topological path and service chain while maintaining per-flow state only at the ingress node
 - A segment is referred to by its Segment Identifier (SID)



SEGMENT TYPES

- Link-State IGP Segments
 - Represent IGP prefixes or IGP adjacencies
 - Signaled by IS-IS or OSPF
- BGP Segments
 - Correspond to BGP peers
 - Signaled by BGP
- Binding Segments
 - Refer to SR policies

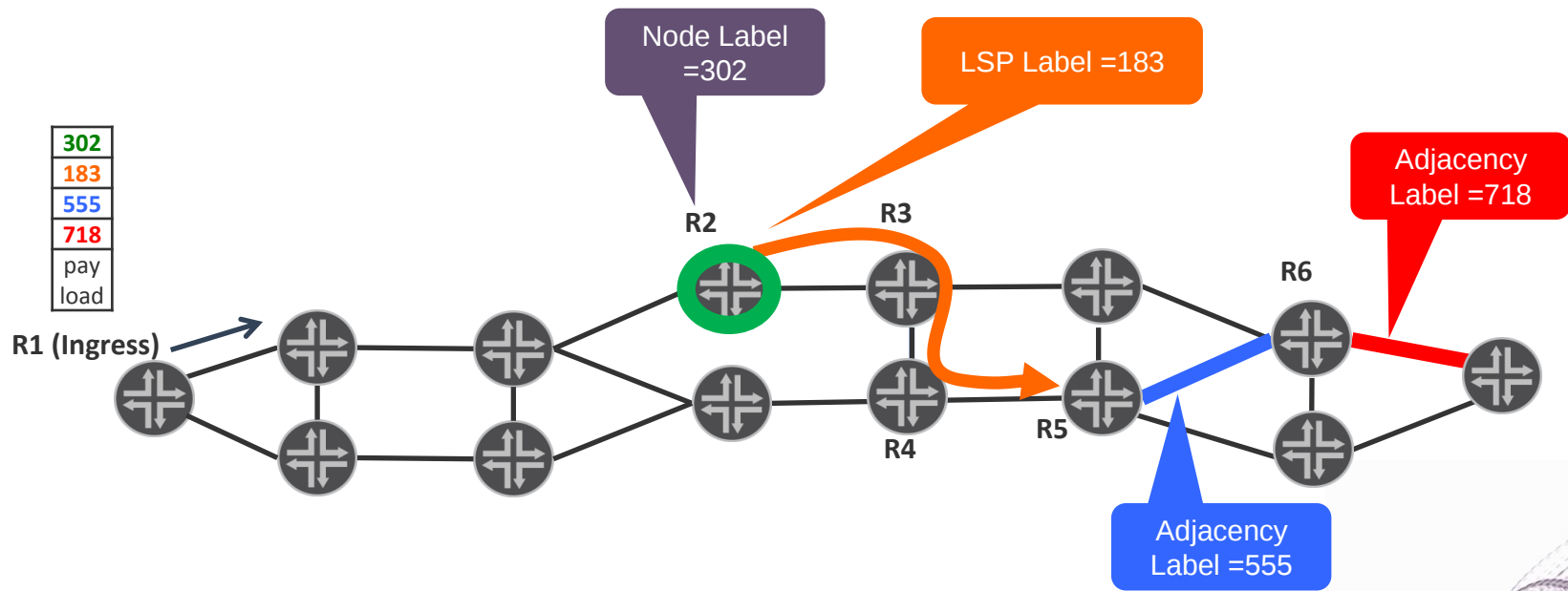


IGP SEGMENT TYPES

- **Adjacency Segments**
 - IGP Segment attached to an unidirectional adjacency or set of adjacencies
 - One-hop path to immediate neighbor
 - Local – only the originating router assigns a label to the local segment
- **Prefix Segments**
 - IGP segment attached to an IGP prefix
 - Identifies shortest-path computed by the IGP to the related prefix
 - Multi-hop path tunnels to all other nodes
 - Global – every router in the domain assigns a (local) label to the (global) segment
- **Node Segments**
 - IGP prefix segment which identifies a specific router (e.g. a loopback)
 - Enforces shortest-path forwarding to the related node
- **Anycast Segments**
 - IGP prefix segment which identifies a set of routers
 - Enforces shortest-path forwarding towards the closest node of the anycast set



SR - KEY CONCEPTS



MPLS label corresponding to a given segment is advertised in the IGP.
Currently being developed in IETF in the SPRING working group.

SPRING = Source Packet Routing in Networking

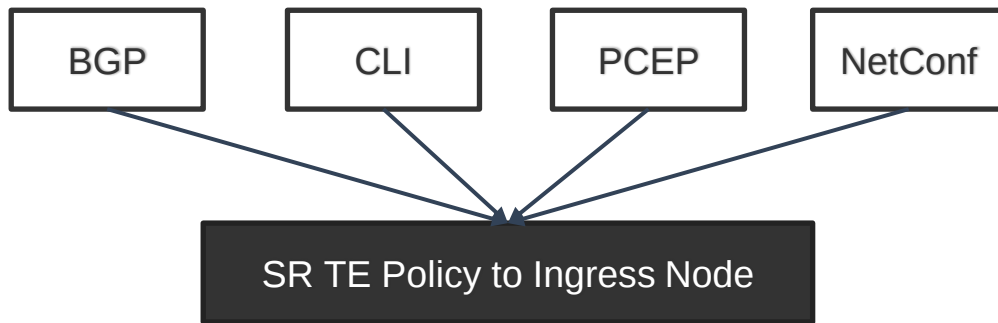
SRTE – INTRODUCTION

- Simple, Automated and Scalable
 - No Core State : state is in the packet header
 - No tunnel interface : “SR Policy”
 - On Demand policy instantiation
 - Automated Steering of packets
- Multi-Domain
 - SR Controller
 - Binding-SID for stitching multiple segments
- SRTE architecture applies to MPLS and IPv6 applications
- SRTE next-hop is a list or lists of SIDs that operator wants incoming traffic to use.



SRTE – INTRODUCTION [CONTD.]

- A head-end node can get to know about SR-TE path for a SR Policy by various means.
- SR-Policy is represented in FIB as a BSID-keyed entry
- A path is selected for a SR Policy when it is valid & its preference is the highest value
- The protocol source of the path does not matter in path selection logic
- When a new Cpath is learned or previous Cpath expires, selection process is re-executed
- SRTE policy maintains a SR-TE Database that is multi-domain capable



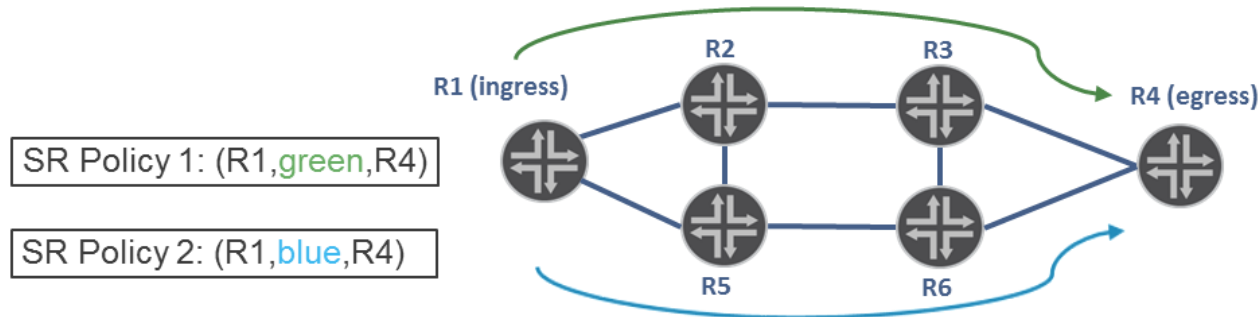
SR TRAFFIC ENGINEERING & SR POLICY

- SR Traffic Engineering / Traffic Steering
 - Headend node steers traffic through explicit paths
 - State is in the packet, no per-flow states in the core
- SR Policy
 - draft-filsfils-spring-segment-routing-policy
 - Set of explicit paths represented by ordered list of segment
 - Injected via BGP, local configuration, PCEP, or NETCONF
 - Headend node steers a flow into a SR Policy



SR POLICY – COLOR

- A SR Policy is identified by its tuple (headend, color, end-point)
 - Headend – node where the SR policy is instantiated
 - Color – numerical value that associates the SR policy with an intent
 - End-point – destination of the SR policy
- At a given head-end node, a SR policy is uniquely identified by its tuple (color, end-point)



COLOR EXTENDED COMMUNITY ATTRIBUTE

- The color extended community is specified in RFC 5512 and draft-ietf-idr-segment-routing-te-policy
- It is a Transitive Opaque Extended community
- The color value is a flat 32-bit number
- Remote prefixes advertised via BGP with Extended Color Community (Type 0x03 / Sub-type 0x0b) to steer traffic into a policy

```
protocols {
  bgp {
    group iBGP {
      type internal;
      neighbor 172.17.2.6 {
        family inet {
          unicast {
            extended-nextthop-color;
          }
        }
      }
    }
  }
}
```

* 11.0.1.11/32 (1 entry, 1 announced)

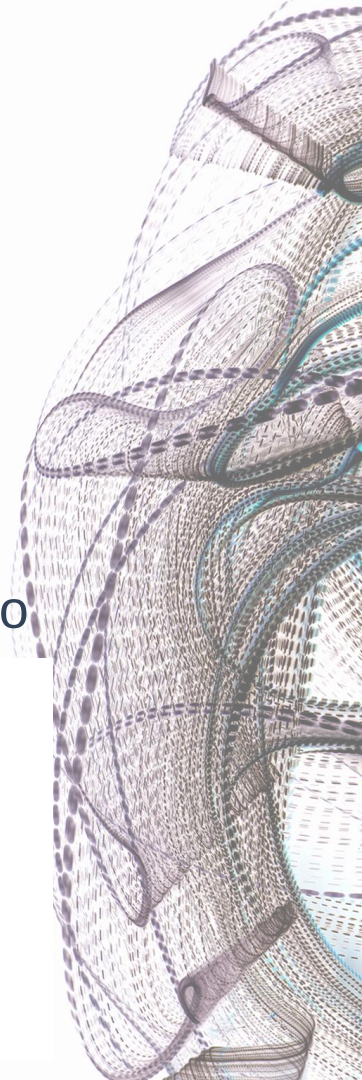
BGP group iBGP type Internal

Nexthop: Self

Localpref: 100

AS path: [3320] I

Communities: color:0:1111



SR POLICY – CANDIDATE PATH, SEGMENT LIST, AND BINDING SID (BSID)

- Candidate Path and Segment List
 - SR policy is associated with one or more candidate paths
 - Each candidate path is associated with a Segment-List (SID-List), or a set of SID-Lists with weights
- Binding SID
 - Each candidate path may be defined with a Binding SID
 - BSID of a SR policy represents its active candidate path
 - Binding SID label mapped to outgoing SR TE path(s)

```
SR policy NAME <headend, color, endpoint>
  Candidate-path PATH1 <protocol, origin, discriminator>
    Preference 200
    BSID 100
    Weight W1, SID-List1: <SID11...SID1i>
    Weight W2, SID-List2: <SID21...SID2j>
  Candidate-path PATH2 <protocol, origin, discriminator>
    Preference 100
    BSID 100
    Weight W3, SID-List3: <SID31...SID3i>
    Weight W4, SID-List4: <SID41...SID4j>
```

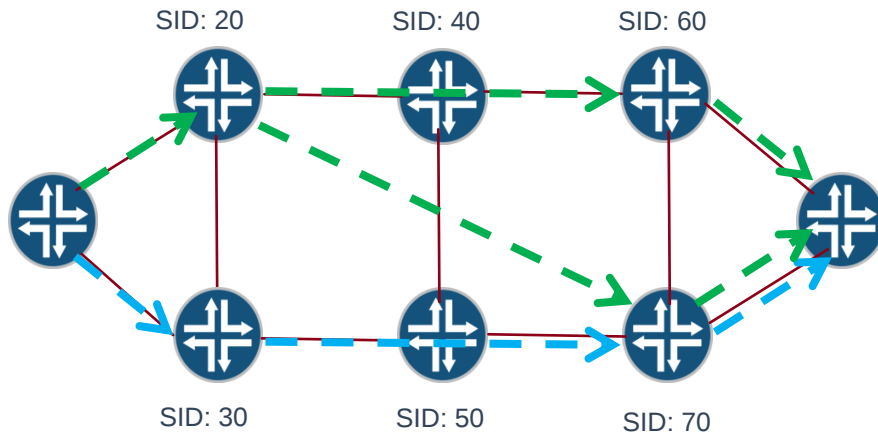


ADVERTISING SR TE POLICIES IN BGP (DRAFT-IETF-IDR-SEGMENT-ROUTING-TE-POLICY)

- New SR TE Policy SAFI (73) & NRLI:
 - <Distinguisher, Policy Color, Endpoint>
 - <Color, Endpoint> tuple – identifies policy, matches destination prefix
 - Distinguisher – makes SR TE policy unique across multiple originators
- Tunnel Encapsulation Attribute (23)
 - Tunnel Type: SR TE Policy (15) – new sub-TLVs:
 - Binding SID – instruction to allocate e.g. a binding label to SR TE policy
 - Preference – preference among multiple SR TE policy originators
 - Segment List – defines explicit TE path
 - Weight – used for weighted UCMF
 - Segment – single element of explicit path
 - Segment



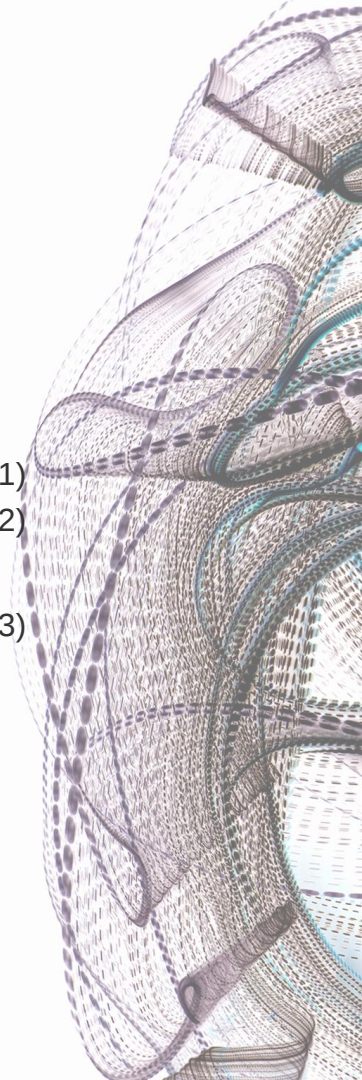
STATIC WECMP SPRING TE CONFIGURATION



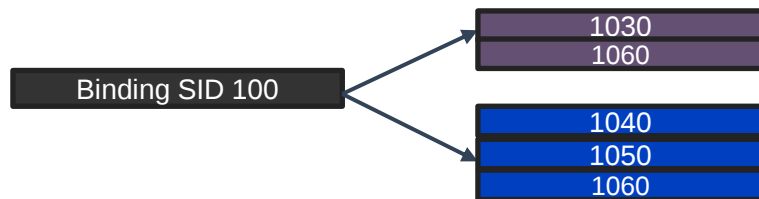
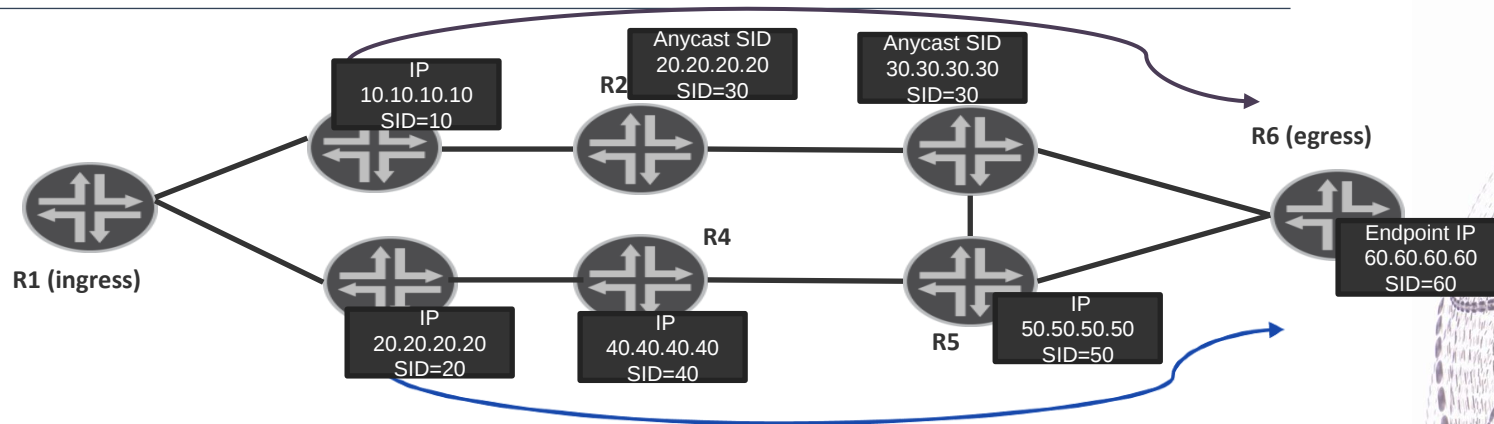
```
source-routing-path sr_lsp1 {  
  to 10.0.0.8  
  color 10  
  binding-sid 1000 (optional)  
  primary P1 weight 2  
  primary P2 weight 1  
}
```

Policy: (Endpoint, Color)

- Endpoint: 10.0.0.8
- Color:
 - Green (10)
 - Segment List: 20, 60, 80 (P1)
 - Segment List: 20, 70, 80 (P2)
 - Blue (11)
 - Segment List: 30, 70, 80 (P3)
- Weight
 - Load balancing



SRTE – FORWARDING USING SIDS



IP addr of top label : 1010 <-> 10.10.10.10 is used for resolution / Label Swap
SR-TE Label Stack 1 (weight = x)

IP addr of top label: 1020 <-> 20.20.20.20 is used for resolution / Label Swap
SR-TE Label Stack 2 (weight = y)

SPRING- FLEX-ALGO

Simplified Traffic Engineering



FLEXIBLE ALGORITHM

- Mechanism to create paths for different Algorithms
- Algorithms could define
 - Different computation algorithms
 - Different metric-type ex: latency
 - Different constraints ex: link color
- Separate Routing-tables for each algorithm
- Mechanism to advertise separate SR-SIDs
- MPLS paths corresponding to the algorithm
- Sub-second FRR convergence with TI-LFA
- ECMP and Load-balancing by default



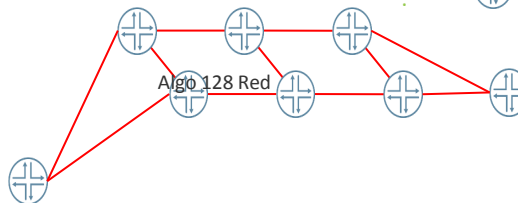
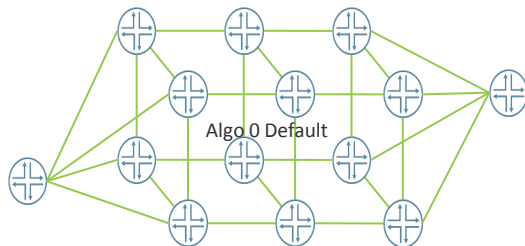
FLEXIBLE ALGORITHM (CONTD.)

- IGP Flexible Algorithm draft-ietf-lsr-flex-algo-01
- We can achieve Traffic Engineering using only IGP
- Algorithm can be defined by users = Flexible Algorithm = Flex Algo
- Flex Algo Definition (Combination of the following attributes)
 - Metric-type : (IGP cost / Delay / TE cost)
 - Constraints : MPLS Admin Groups a.k.a. Link color
 - Numeric identifier = 128 - 255
- Router has separated RIB each Flex algo and calculate SPF each Flex Algo



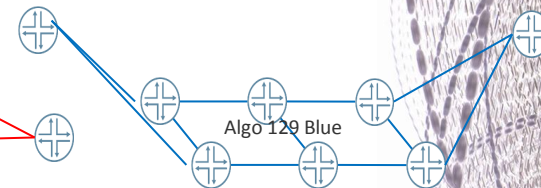
FLEX-ALGO with SR

Single Loopback address can be used for each Flex Algo
We can use Individual Node SID (Prefix SID) for each Flex Algo
The following SR features are usable for each Flex Algo
FRR using TI-LFA (within 50msec in general)
ECMP aware using L-ISIS/L-OSPF route

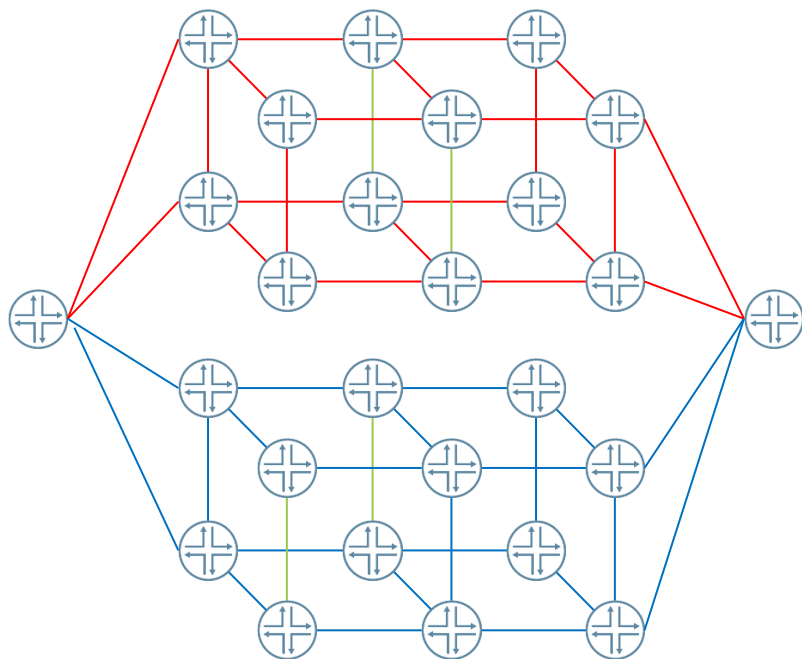


Router 1 Lo 1.1.1.1
Algo 0 SID 16001
Algo 128 SID 16101
Algo 129 SID 16201

Router 2 Lo 2.2.2.2
Algo 0 SID 16002
Algo 128 SID 16102
Algo 129 SID 16202



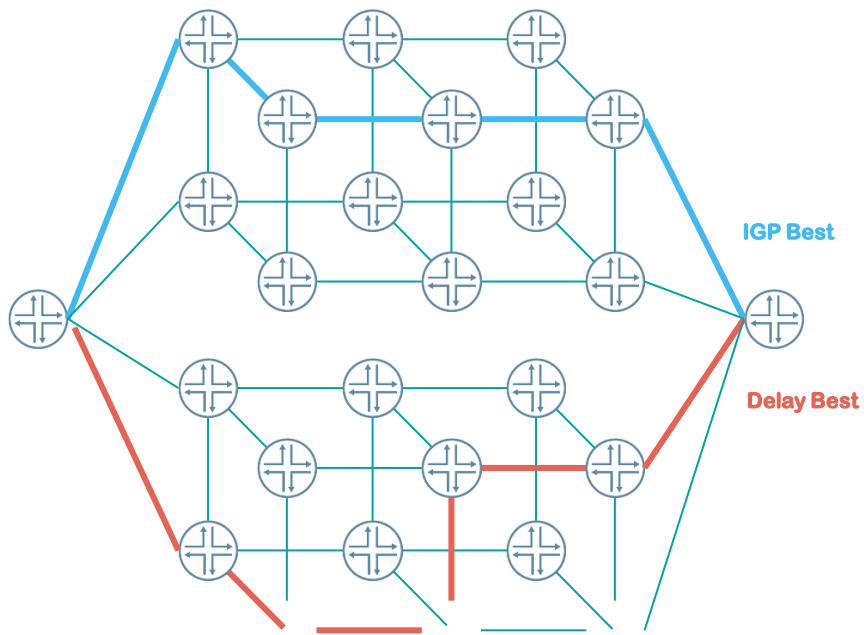
Use-Case: Dual Plane Core



- Algo 128 for Core 1
 - ID 128, metric IGP, link **Red**
- Algo 129 for Core 2
 - ID 129, metric IGP, link **Blue**
- Service Traffic A = Core 1
- Service Traffic B = Core 2
- Calculation of SPF, advertising SIDs and TI-LFA are executed each Core Plane separately
- Core Plane 1&2 is completely separated in logical



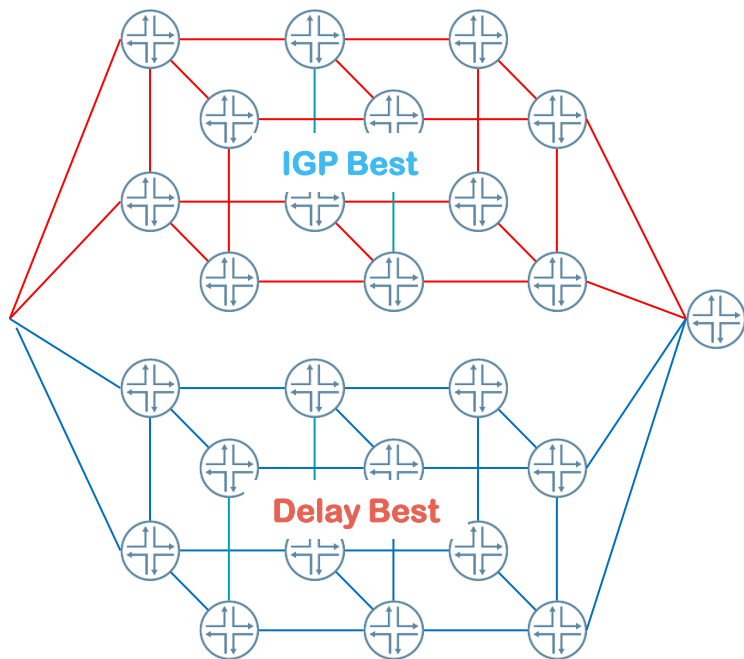
Use Case: Low latency path



- Algo 128
 - ID 128, metric **IGP**
- Algo 129
 - ID 129, metric **Delay**
- Normal Service Traffic is dealt with Algo 128
- Delay sensitive Service Traffic is dealt with Algo 129
- All of links in domain are usable



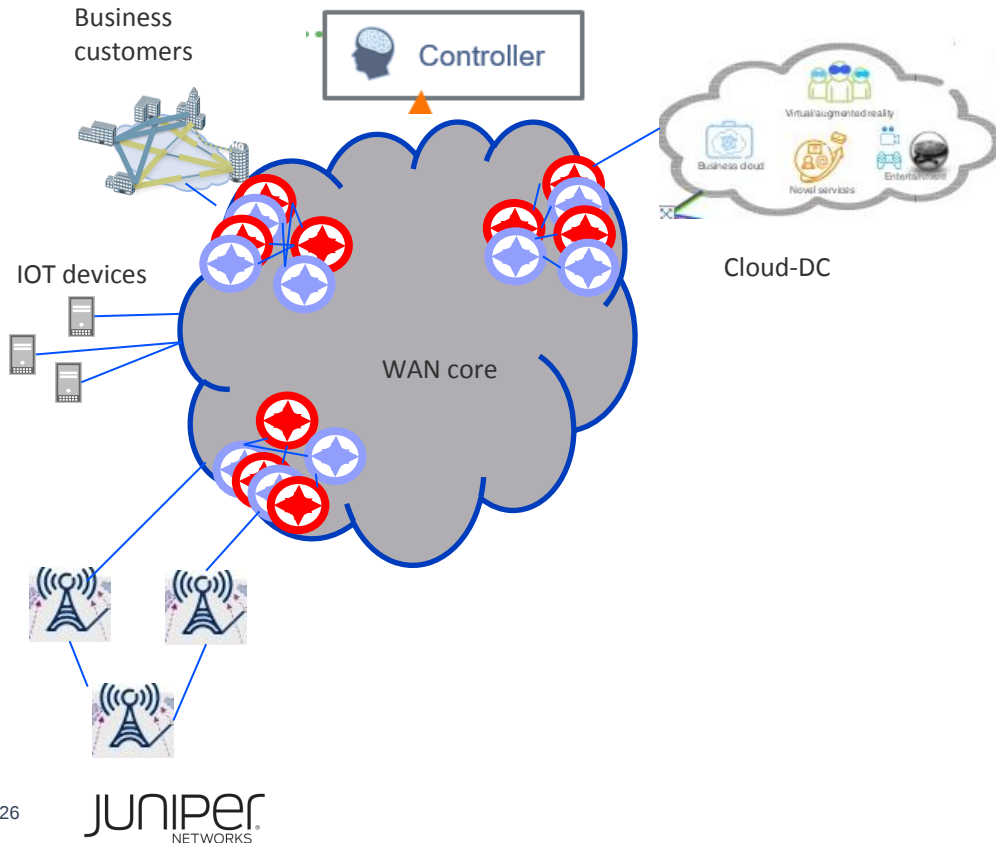
Use Case: Dual Plane + Different Metric



- Algo 128 for Core 1
 - ID 128, metric **IGP**, link **Red**
- Algo 129 for Core 2
 - ID 129, metric **Delay**, link **Blue**
- Normal Service Traffic is dealt with at Core 1 (IGP Best domain)
- Delay sensitive Service Traffic is dealt with at Core 2 (Delay Best)
- Calculation of SPF, advertising SIDs and TI-LFA are executed each Core Plane separately
- Core Plane 1&2 is completely separated in logical

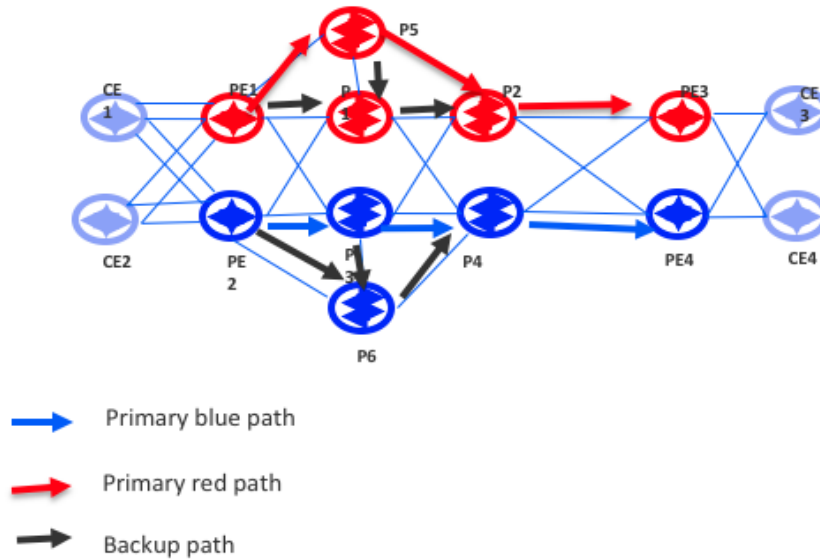


USE CASE : NETWORK SLICING



- Reducing Capex and Opex is the goal for many service providers, Providers want to operate one network for all kinds of traffic
- Network Slicing provides a means to run different logical networks on top of same physical infrastructure
- Flex-Algo provides efficient means for logical separation
- Easier to configure and maintain
- User defined algorithm and constraints
- Automated traffic steering

TI-LFA PROTECTION WITHIN TOPOLOGY



- TI-LFA protection and local-FRR per-each algorithm
- Protection paths follow constraints of the algorithm



IETF Drafts

[draft-ietf-spring-segment-routing](#)

Segment Routing architecture

[draft-ietf-spring-segment-routing-mpls](#)

Segment Routing details with MPLS forwarding

[draft-ietf-isis-segment-routing-extensions](#)

ISIS extensions to distribute SR segments

[draft-ietf-isis-prefix-attributes](#)

Advertising ISIS prefix attributes for SR

[draft-ietf-ospf-segment-routing-extensions](#)

OSPF extensions to distribute SR segments

[RFC 7684](#)

OSPFv2 Prefix/Link Attribute Advertisement

[draft-hegde-rtgwg-microloop-avoidance-using-spring-01](#)

Micro-loop avoidance using SPRING

[draft-francois-rtgwg-segment-routing-ti-lfa](#)

FRR using TI-LFA.



IETF Drafts

[draft-rosen-idr-rfc3107bis-00](#)

Advertise a label stack using BGP-LU

[draft-gredler-idr-bgp-ls-segment-routing-ext](#)

BGP LS extensions for exporting SR topology to a controller (north bound)

[draft-ietf-pce-segment-routing](#)

PCE extensions to setup a SR path different from shortest path (SR-TE) from the controller (south bound)

[draft-sreekantiah-idr-segment-routing-te](#)

Advertise SR-TE policies via BGP

[draft-sivabalan-pce-binding-label-sid](#)

Advertise binding label SID to steer traffic through a TE LSP

[draft-tantsura-bgp-ls-segment-routing-msd-00](#)

[draft-tantsura-ospf-segment-routing-msd-00](#)

[draft-tantsura-isis-segment-routing-msd-00](#)

Exposing Max Label stack depth supported by a node

[draft-hegde-spring-node-protection-for-sr-te-paths-00](#)

Node Protection for SR-TE Paths



TECHNICAL DOCUMENTATION

ISIS Spring

- https://www.juniper.net/documentation/en_US/junos/topics/concept/source-packet-routing.html

Static SR LSPs

- https://www.juniper.net/documentation/en_US/junos/topics/topic-map/static-segment-routing-lsp.html

BGP SR TE Policy

- https://www.juniper.net/documentation/en_US/junos/topics/concept/bgp-sr-te.html
- https://www.juniper.net/documentation/en_US/junos/topics/task/configuration/bgp-sr-te-taskconf.html

NorthStar

- https://www.juniper.net/documentation/en_US/northstar4.0.0/topics/concept/northstar-spring.html

PCEP

- https://www.juniper.net/documentation/en_US/junos/topics/concept/spring-te-for-pcep-overview.html

BGP-LS

- https://www.juniper.net/documentation/en_US/junos/topics/concept/spring-te-for-pcep-overview.html



THANK YOU.



Aditya Kaul
Solution Architect
Professional Services-APAC
mobile +60 17 664 4381
email kaula@juniper.net
twitter  @Kaul_Addie

www.juniper.net

