Segment Routing Traffic Engineering

Jose Liste
Technical Marketing Engineer
jliste@cisco.com

www.isocore.com/2015

SDN MPLS

CISCO

Agenda

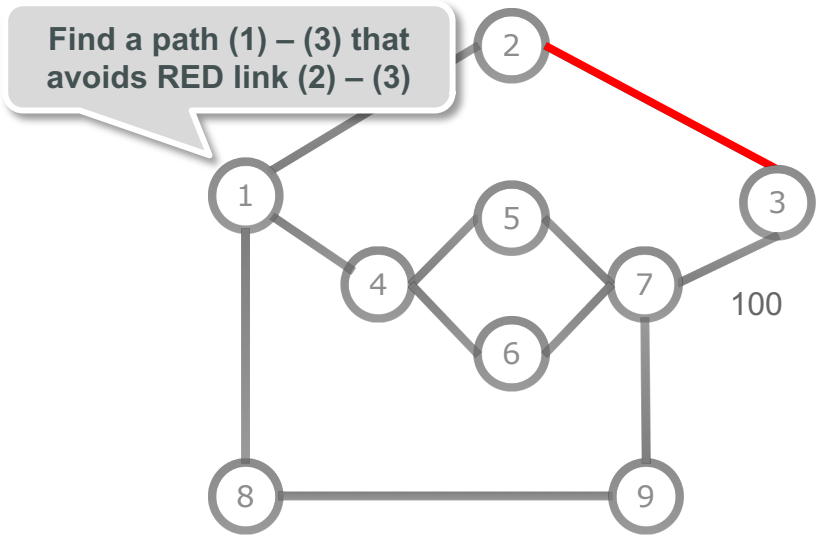
- What is Segment Routing Traffic Engineering?
- Use Case: Intra-Domain Latency Opt. + Constrains
- Use Case: Inter-Domain SRTE Disjointness
- Use Case: Dynamic SRTE policies to BGP NH
- Segment Routing Traffic Matrix

Segment Routing

- Source Routing
  - Source chooses a path and encodes it in the packet header as an ordered list of segments
  - Rest of the network executes the encoded instructions without any further per-flow state
- Segment: an identifier for any type of instruction
  - Forwarding or service
- Allows explicit routing / constraint-based routing – SR Traffic Engineering (SRTE)
- Strikes a balance between distributed and centralized intelligence

Traffic Engineering with Segment Routing

- SRTE brings innovative solutions to address TE problems
  - IP-centric (ECMP aware), Scale (no state / Inter-AS) and Simplicity
- Extensive scientific research backing new SRTE algorithms (1)
  - Applicable to both Controllers (centralized) and Routers (distributed)
- Uses existing ISIS / OSPF extensions to advertise link attributes
- No extra protocol to establish LSPs (no midpoint state)

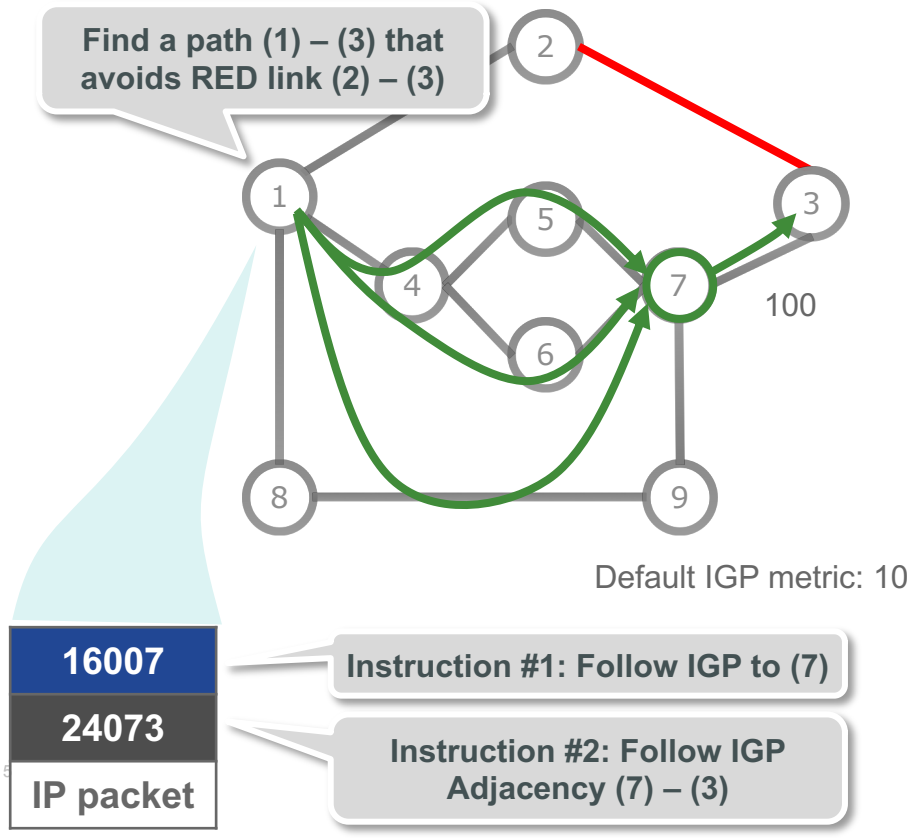


Default IGP metric: 10

(1) SIGCOMM 2015: http://conferences.sigcomm.org/sigcomm/2015/pdf/papers/p15.pdf

Traffic Engineering with Segment Routing

- SRTE uses a "Policy" (SID-list) to steer traffic through the network
- Policy path can be computed by router or controller
- SID list pushed by head-end
- Rest of network executes instructions embedded in the SID list



SDN MPLS

SRTE Policy instantiated from configured tunnels

Instantiate SRTE Policy from configured tunnel

- SRTE Policies can be instantiated from an interface tunnel-te configuration
  - The desired characteristics of the SRTE Policy are specified in the tunnel-te interface configuration
  - A path-option, configured under tunnel-te interface, specifies how the path of the instantiated Policy is derived (explicit or dynamic)
- A simple example:

```
interface tunnel-te 1
 ipv4 unnumbered Loopback0
 destination 1.1.1.10
 path-option 1 dynamic segment-routing
```

tunnel-te source address

tunnel-te destination address

SRTE path-option

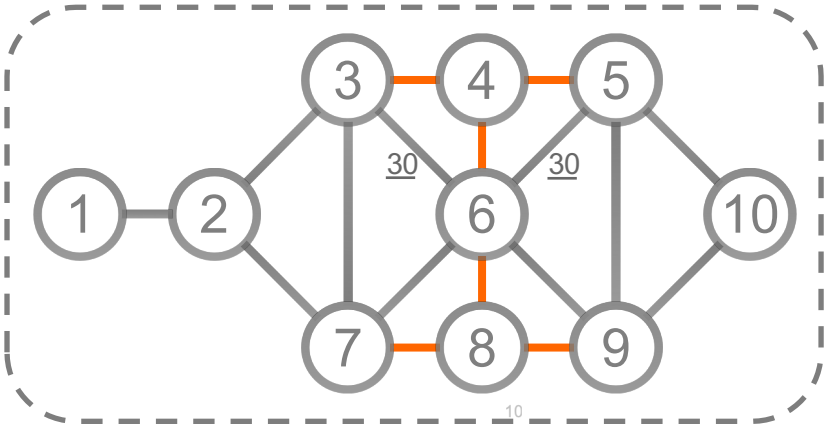
Use Case: Intra-Domain Latency Optimization with Constrains (Affinity)

Dynamic Path

- The path of an SRTE Policy instantiated from a configured tunnel-te can be dynamically calculated
- Allows to specify optimization objective and constraints

Use Case – Latency Optimization + Affinity

- The links to Node4 and Node8 are to be avoided
  - Operator marks these links with admin-group/color “ORANGE”
- Path of tunnel-te1, from Node1 to Node10, must avoid links with “ORANGE” color while optimizing for latency (based on TE metric)



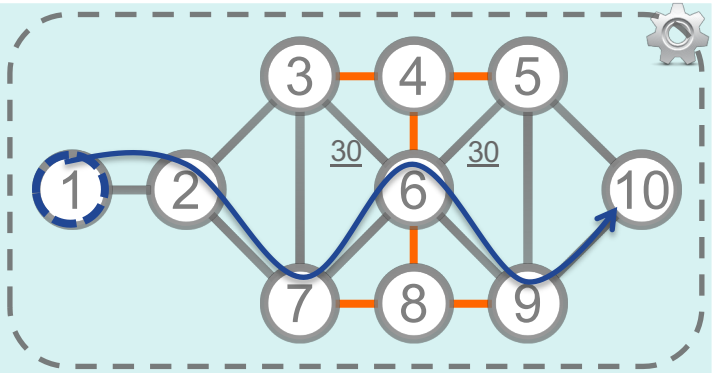
IGP metric: 10
Default TE metric: 10

Use Case – Latency Optimization + Affinity

Shipping functionality

- Node1 configuration under tunnel-te:

```
mpls traffic-eng
 affinity-map ORANGE bit-position 8
!
interface tunnel-te 1
 ipv4 unnumbered Loopback0
 destination 1.1.1.10
 path-selection
  metric te
 !
 affinity exclude ORANGE
 path-option 1 dynamic segment-routing
```



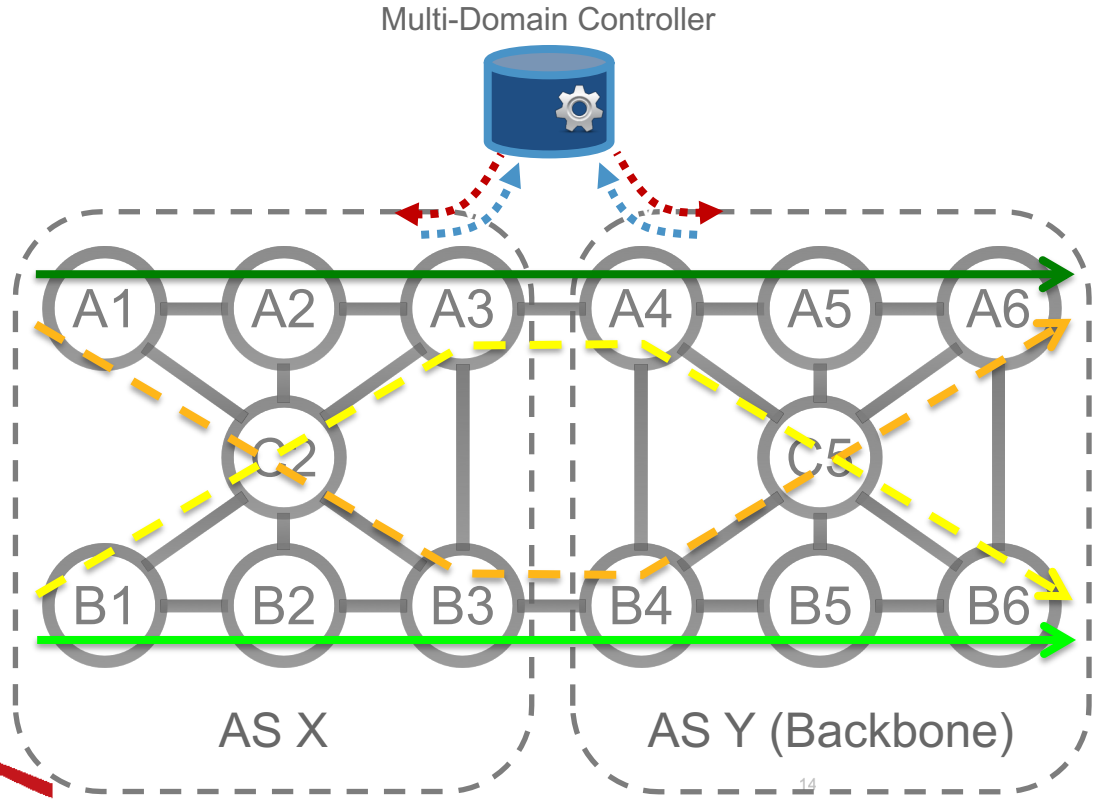
– The color "ORANGE" is represented by bit 8 in the affinity bitmap

Use Case: Inter-Domain SRTE Disjointness

Use Case Highlights

- Operator Requirements
  - Service Disjointness from pair of sources to pair of destinations in a Multi-Domain network
  - Stringent 150msec end-to-end path protection
- Solution
  - Centralized multi-domain topology discovery and path calculation
  - End-to-end SRTE disjoint policies with explicit path-options
  - Primary path and secondary path per policy
  - TI-LFA for local protection and BFD (over SR-TE) for path protection

Inter-AS SRTE Disjointness



A Primary SRTE Policy
A' Secondary SRTE Policy

B Primary SRTE Policy
B' Secondary SRTE Policy

These path pairs are disjoint:

A Primary SRTE Policy
B Primary SRTE Policy

A Primary SRTE Policy
A' Secondary SRTE Policy

B Primary SRTE Policy
B' Secondary SRTE Policy

SDN MPLS

CISCO

Inter-AS SRTE Disjointness

```
interface tunnel-te1
 ipv4 unnumbered Loopback0
 bfd
  fast-detect sbfd
  multiplier 3
  minimum-interval 50
 !
 destination 1.1.1.2
 path-protection
 path-option 10 explicit name PRIMARY segment-routing protected-by 20
 path-option 20 explicit name SECONDARY segment-routing protected-by 10
```

BFD over SR-TE

Policy with two (2) paths

Inter-AS SRTE Disjointness

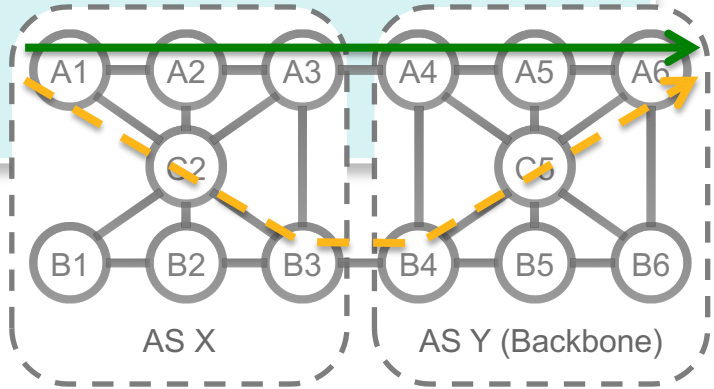
Ships in Jan. 2016

```
explicit-path name PRIMARY
 index 10 next-address strict ipv4 unicast 1.1.1.A2
 index 20 next-address strict ipv4 unicast 1.1.1.A3
 index 30 next-label A3A4
 index 40 next-label 160A5
 index 50 next-label 160A6
!
explicit-path name SECONDARY
 index 10 next-address strict ipv4 unicast 1.1.1.C2
 index 20 next-address strict ipv4 unicast 1.1.1.B3
 index 30 next-label B3B4
 index 40 next-label 160C5
 index 50 next-label 160A6
```

Loopback IP (local AS)

NNI label – MPLS static

Prefix SID (remote AS)



1.1.1.A2: loopback of A2
A3A4: mpls static label A3→A4
160A5: prefix-SID of A5

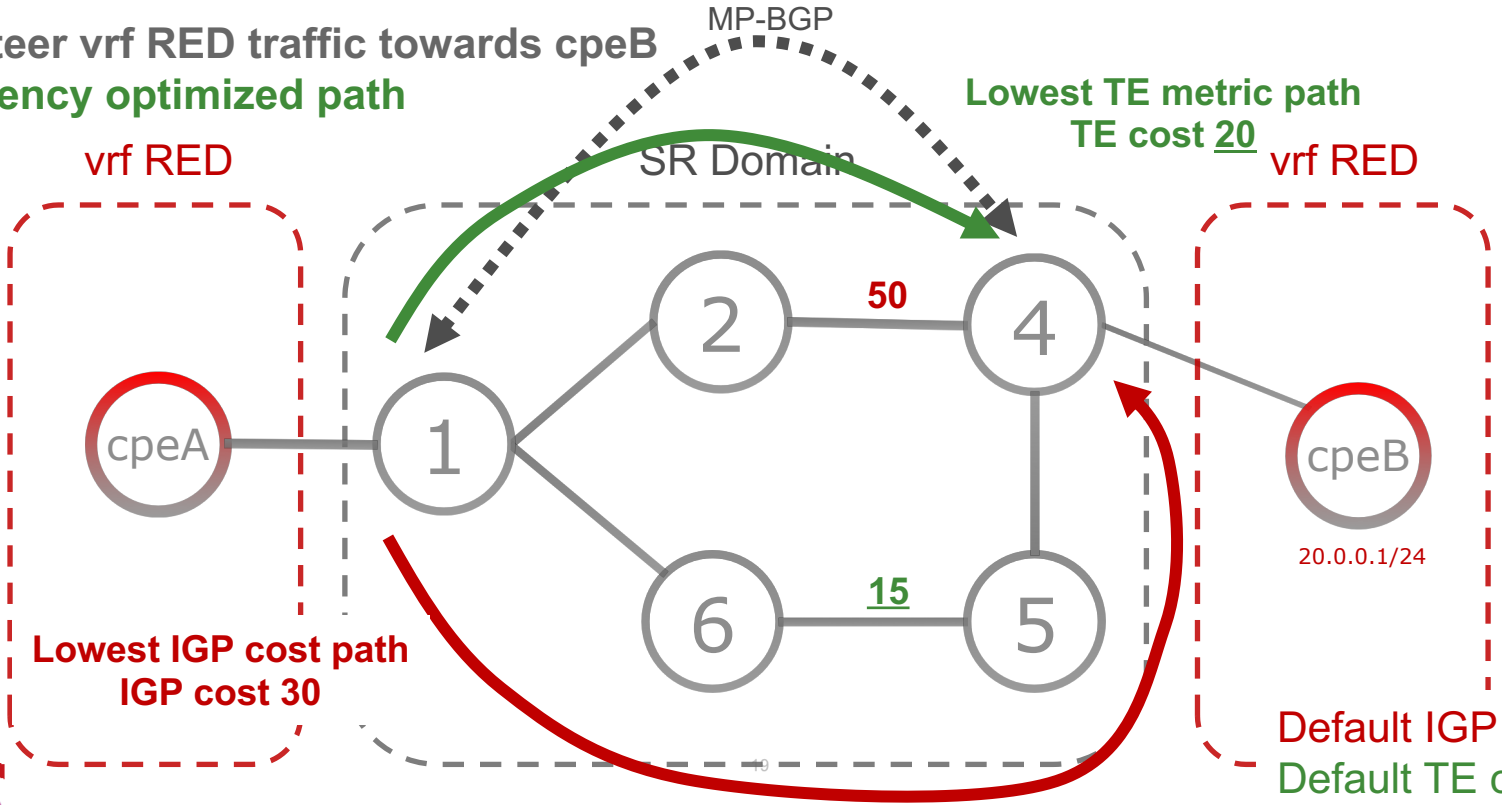
Dynamic SRTE policies to BGP next-hops (aka BGP SRTE)

Objective

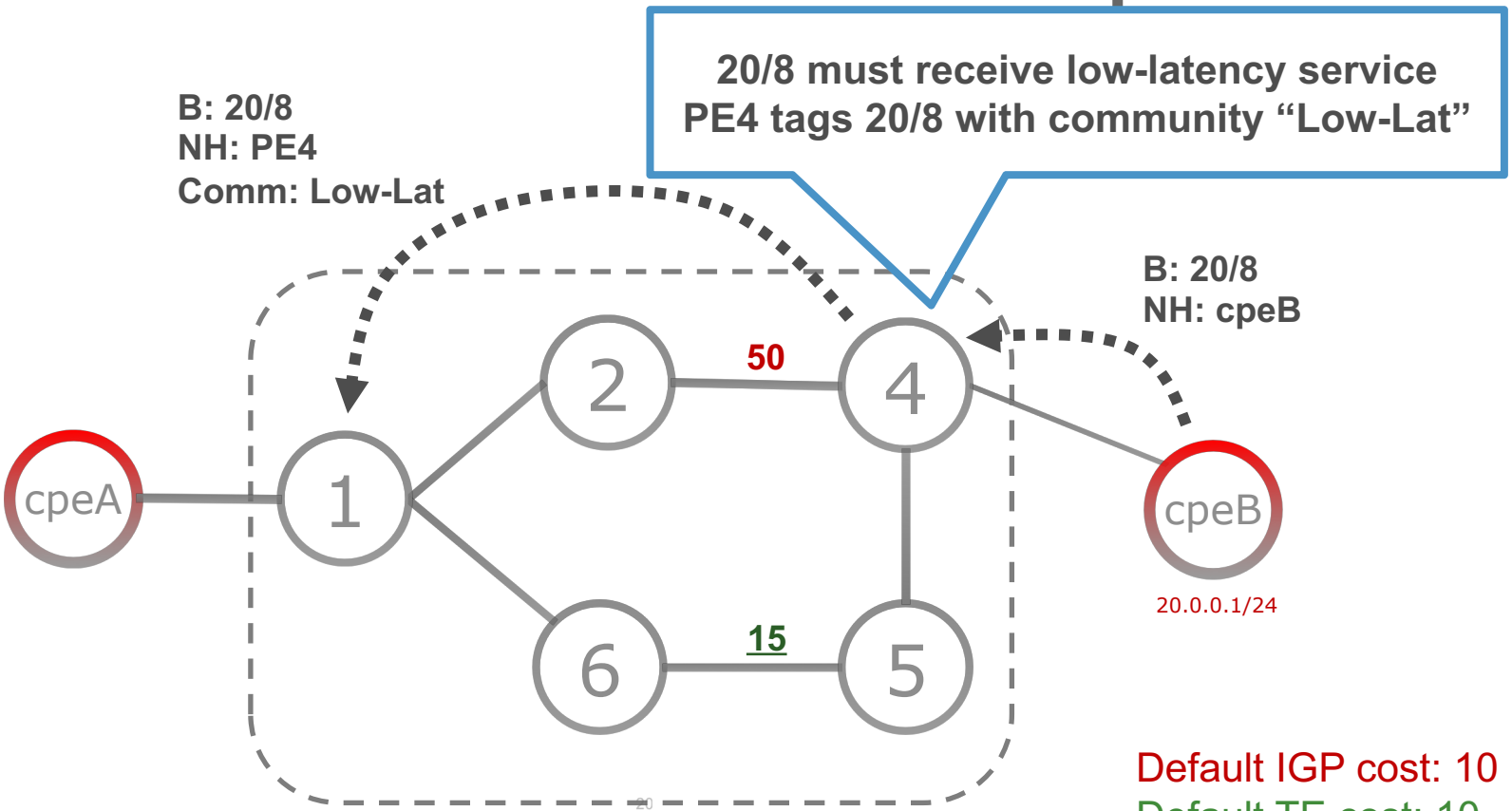
- Trigger automatic TE policies for traffic to VPN destinations
  - Policies that meet customer / application SLA (e.g. latency optimized)
- Without any pre-configured TE tunnel at ingress PE
- Without typical PBR performance tax

Dynamic VPN instantiation of SRTE policies

Goal: steer vrf RED traffic towards cpeB over latency optimized path



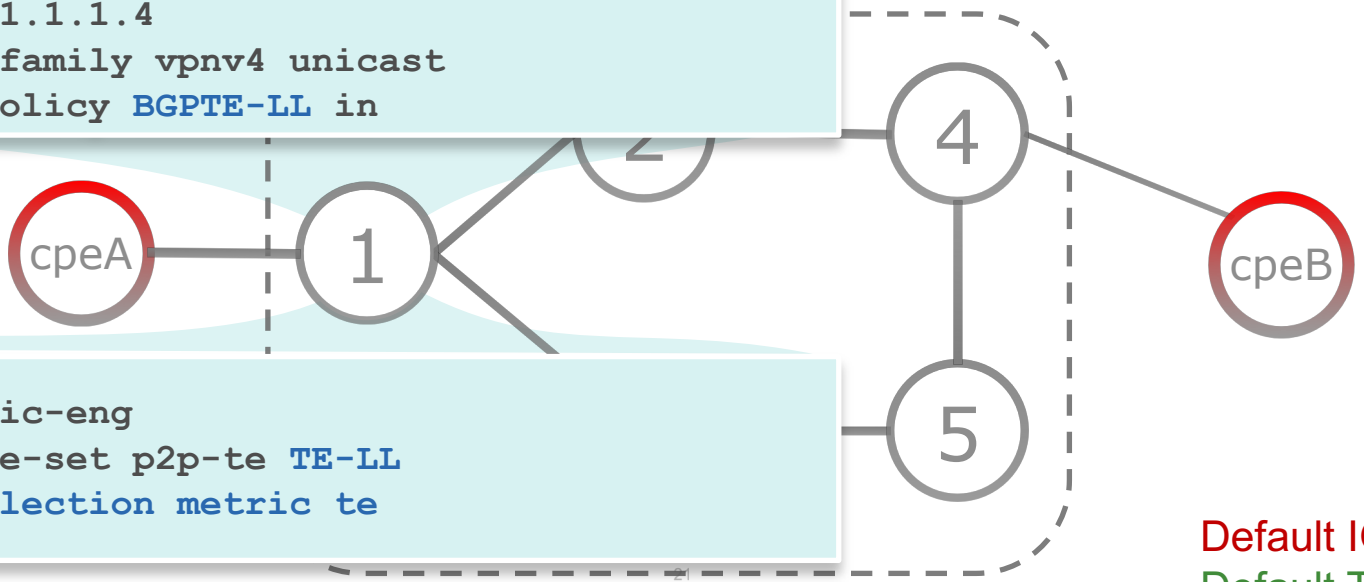
Dynamic VPN instantiation of SRTE policies



Dynamic VPN instantiation of SRTE policies

```
route-policy BGPTE-LL
   if community matches-every (100:1) then
      set mpls traffic-eng attribute-set TE-LL

router bgp 1
 neighbor 1.1.1.4
  address-family vpnv4 unicast
   route-policy BGPTE-LL in
```



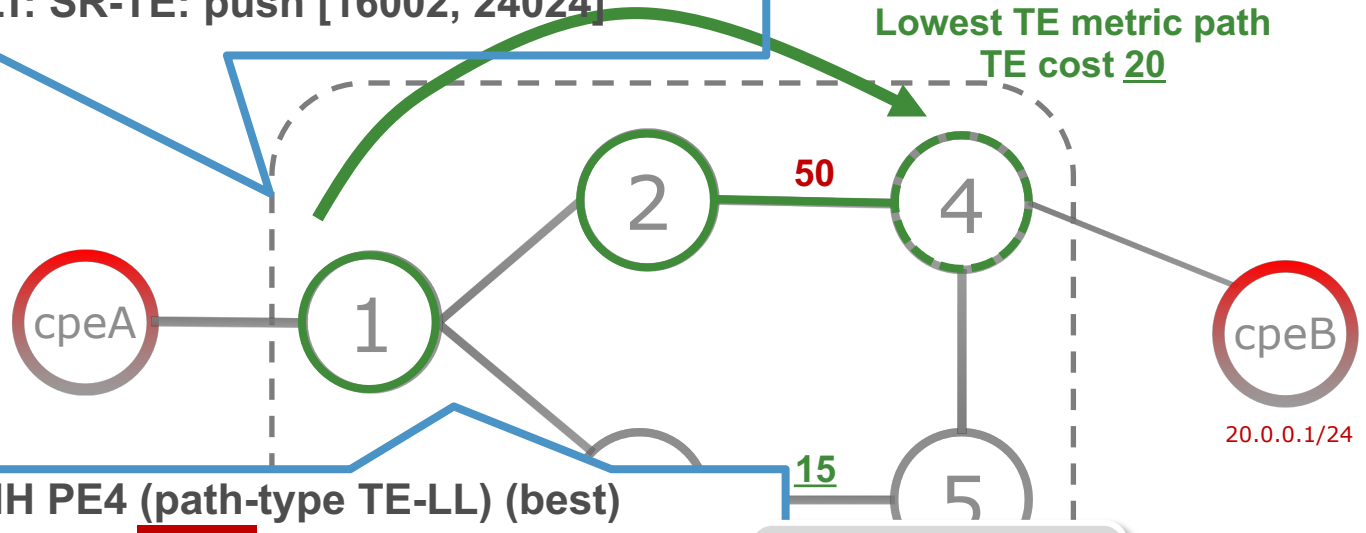

```
mpls traffic-eng
  attribute-set p2p-te TE-LL
   path-selection metric te
```

Default IGP cost: 10
Default TE cost: 10

Dynamic VPN instantiation of SRTE policies

Ships in Nov. 2015

MAP: Low-Lat means “minimize TE metric”
COMPUTE: minimize TE metric to node 4
RESULT: SR-TE: push [16002, 24024]



BGP: 20/8 NH PE4 (path-type TE-LL) (best)
FIB: BGP: 20/8 via 30001
FIB: SRTE: (PE4 –TE-LL)
Binding SID: 30001: Push [16002, 24024]

SRTE policy’s Binding SID

Default IGP cost: 10
Default TE cost: 10

BGP SRTE Dynamic - Benefits

Ships in Nov. 2015

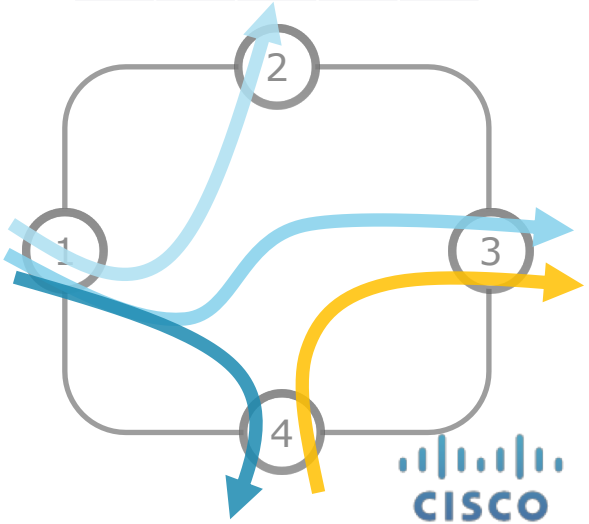
- Significant configuration simplification
  - ZERO tunnel configuration !!!
  - A few optimization templates are configured, the same across all the ingress PE's, each optimization template is identified by a BGP policy based on communities
- Automated steering of BGP routes on the right path
  - Dataplane performant
- BGP PIC FRR dataplane protection is preserved
- BGP NHT fast control plane convergence is preserved

SR Traffic Matrix (TM) Collection

Automated Traffic Matrix Collection

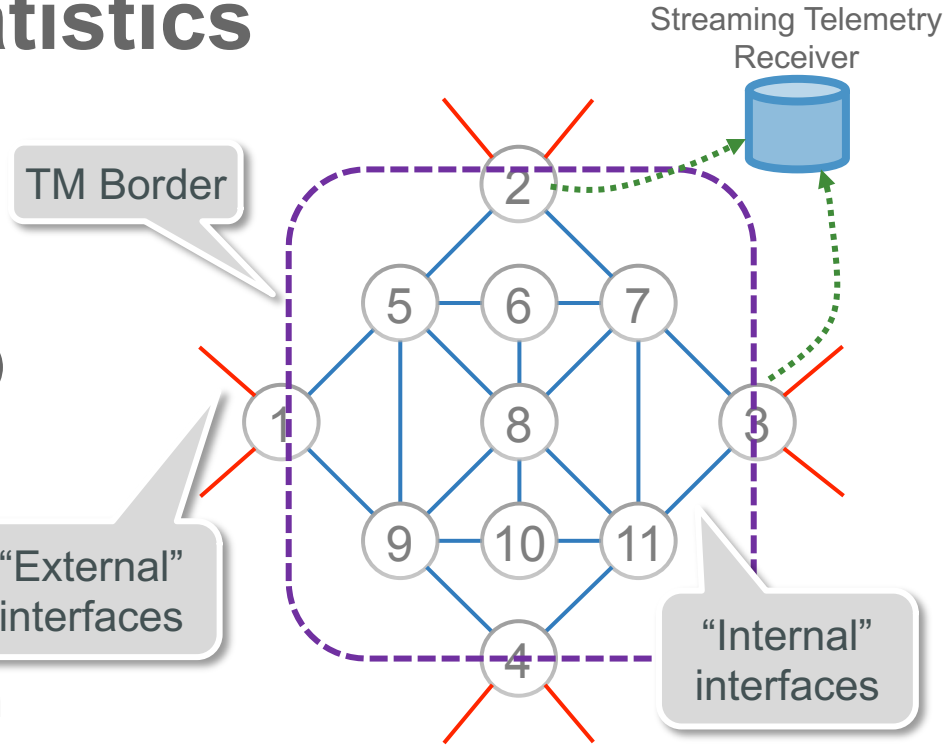
- Traffic Matrix is fundamental for
  - Capacity planning
  - Centralized traffic engineering
  - IP/Optical optimization
- Provides volume of traffic $T_{i,j}$ from *i* to *j* over a time interval, for every ingress point *i* and every egress point *j*
- Most operators do not have an accurate traffic matrix
- With Segment Routing, the traffic matrix collection is automated

	1	2	3	4
1				
2				
3				
4				



SR Traffic Matrix Statistics

- Measures traffic entering TM border from external interfaces towards destination prefix-SIDs
- Base Pcounter[1] for Prefix-SID(N)
  - Accounts any packet switched on the Prefix-SID(N) FIB entry
- TM Pcounter[1] for Prefix-SID(N)
  - Accounts any packet from an external interface and switched on the Prefix-SID(N) FIB entry
- TM stats collected by Streaming Telemetry



Note 1: Pcounter: pair of (packet, byte) counters

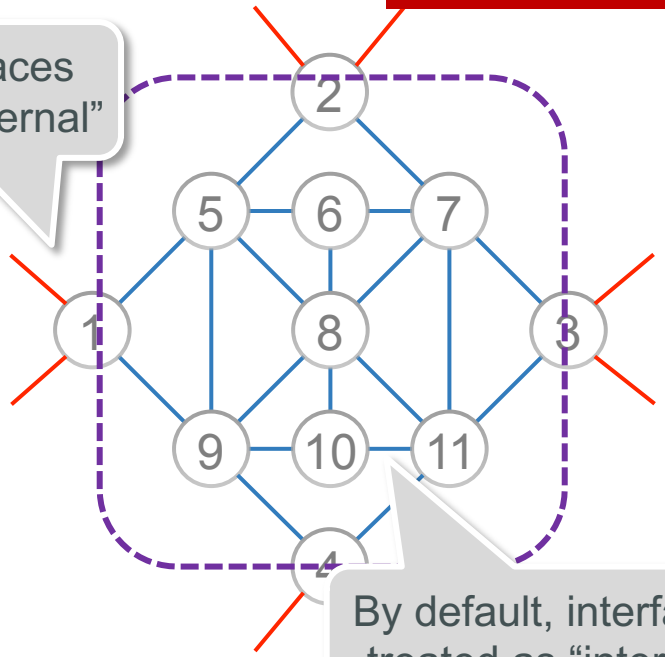
Enable Statistics Collection

Ships in Nov. 2015

- Minimal Traffic Collector (TC) configuration

```
traffic-collector
```

- With this configuration, TC periodically collects Prefix-SID Base and TM Pcounters and Tunnel-te Pcounters and keeps their the history



TM and Base Counter History Database

```
RP/0/RSP0/CPU0:R1#show traffic-collector ipv4 counters prefix 1.1.1.3/32 detail
Prefix: 1.1.1.3/32  Label: 16003 State: Active
Base:
    Average over the last 5 collection intervals:
        Packet rate: 9496937 pps, Byte rate: 9363979882 Bps

    History of counters:
        23:01 - 23:02: Packets 9379529, Bytes: 9248215594
        23:00 - 23:01: Packets 9687124, Bytes: 9551504264
        22:59 - 23:00: Packets 9539200, Bytes: 9405651200
        22:58 - 22:59: Packets 9845278, Bytes: 9707444108
        22:57 - 22:58: Packets 9033554, Bytes: 8907084244
TM Counters:
    Average over the last 5 collection intervals:
        Packet rate: 9528754 pps, Byte rate: 9357236821 Bps

    History of counters:
        23:01 - 23:02: Packets 9400815, Bytes: 9231600330
        23:00 - 23:01: Packets 9699455, Bytes: 9524864810
        22:59 - 23:00: Packets 9579889, Bytes: 9407450998
        22:58 - 22:59: Packets 9911734, Bytes: 9733322788
        22:57 - 22:58: Packets 9051879, Bytes: 8888945178
```

Base Pcounters

Average packet / byte rates

TM Pcounters

Pcounter history Packet / byte count

Conclusion

- SRTE brings a fundamentally different way to look at TE
- New algorithms backed by research
- Applicable to both centralized and distributed path computation scenarios
- SRTE is now REAL with live deployments to occur in CY2016

References / Contact us

- This presentation covered only a subset of the committed projects … More is coming
- Like to share your usecase / questions / concerns: ask-segment-routing@cisco.com
- Detailed SR tutorials at: http://www.segment-routing.net/

THANK YOU

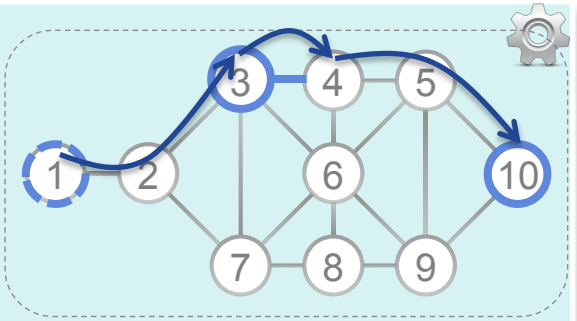
Use Case: Intra-Domain Explicit Path

Path-option explicit segment-routing

Shipping functionality

- SRTE Policy path can be explicitly specified by configuring a segment list as an ordered list of IP addresses and/or label values
- Each of the entries in the ordered list represents a segment
- Example configuration on Node1:

```
explicit-path name PATH1
 index 10 next-address 1.1.1.3    !! Prefix-SID(3)
 index 20 next-address 99.3.4.4   !! Adj-SID(3-4)
 index 30 next-address 1.1.1.10   !! Prefix-SID(10)
!
interface tunnel-te1
 ipv4 unnumbered Loopback0
 destination 1.1.1.10
 path-option 1 explicit name PATH1 segment-routing
```



Router-id of NodeX: 1.1.1.X
Prefix-SID index of NodeX: X
Link address XY: 99.X.Y.X/24 with X<Y
Adj-SID XY: 240XY

Dynamic SRTE policies to BGP next-hops

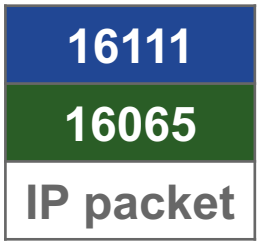
Prepended Segments

BGP SRTE – Prepended (anycast) Segments

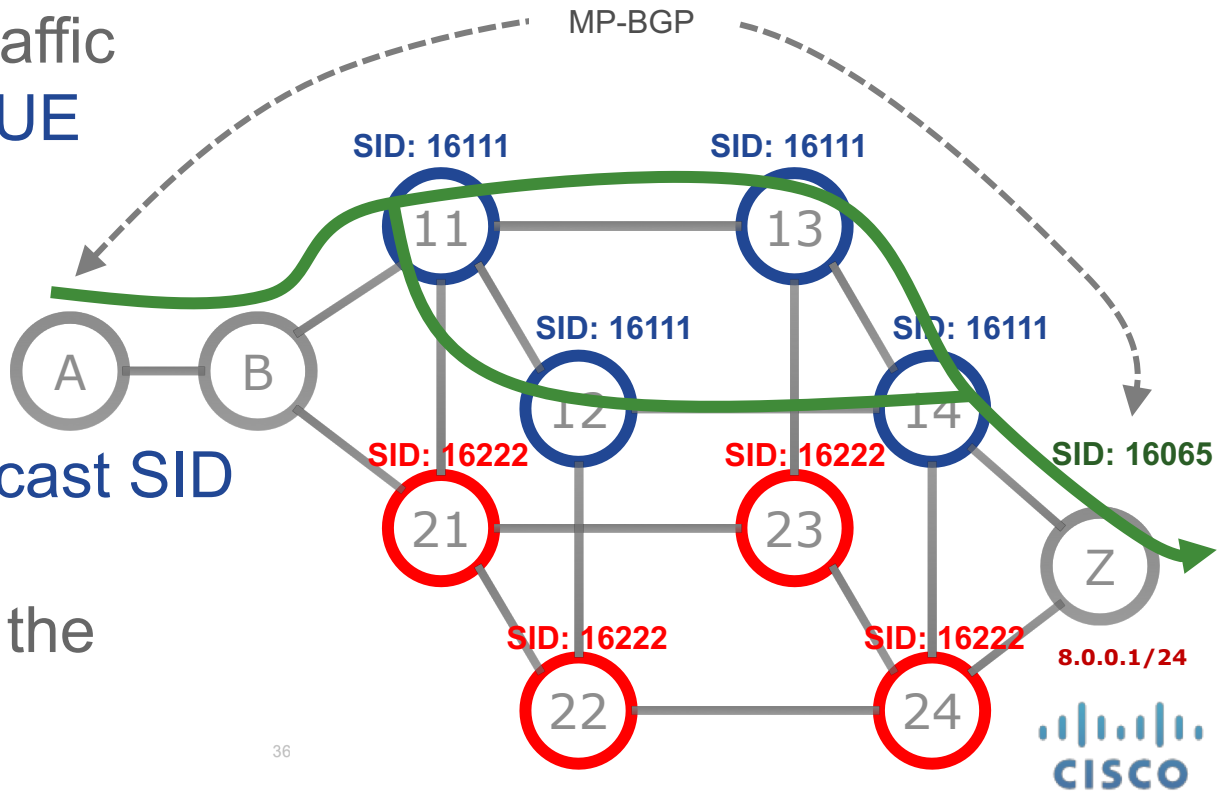
- BGP SRTE can also be used to force traffic to destination to step via prepended (Anycast) segments
- Use Cases:
  - Dual-plane disjointness (e.g. go via plane 1's Anycast SID)
  - Low-latency in international topology (e.g. go via country XYZ Anycast SID)

Prepend – Plane Disjointness

Goal: steer VPN xyz traffic towards Z over the BLUE plane



Solution: prepend Anycast SID of the target plane before the segment of the BGP nhop

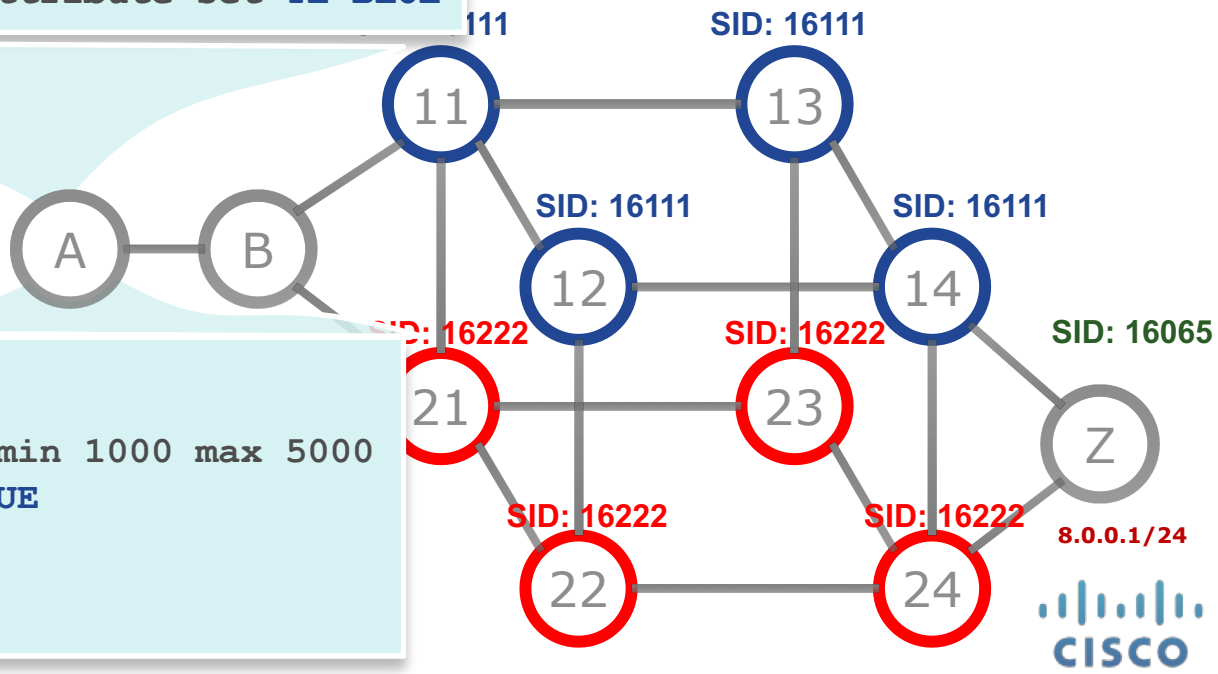


Prepend – Plane Disjointness

Ships in Nov. 2015

```
route-policy BGPTE-PLANE-BLUE
   if community matches-every (100:1) then
      set mpls traffic-eng attribute-set TE-BLUE
```

```
mpls traffic-eng
  auto-tunnel p2p tunnel-id min 1000 max 5000
  attribute-set p2p-te TE-BLUE
   index 1 mpls label 16111
   index 2 bgp-nhop
```



Prepend – Plane Disjointness

Ships in Nov. 2015

FIB:
BGP: 8/8 via 30001
SRTE: (Z, TE-BLUE): BSID 30001: push [16111, 16065]
ISIS: Z/32: 16065 to 16065 via B