RSVP-TE Pop&Go: Using a shared MPLS forwarding plane

Harish Sitaraman Juniper Networks
Vishnu Pavan Beeram Juniper Networks
Mazen Khaddam Cox Communications

NANOG 72, Atlanta

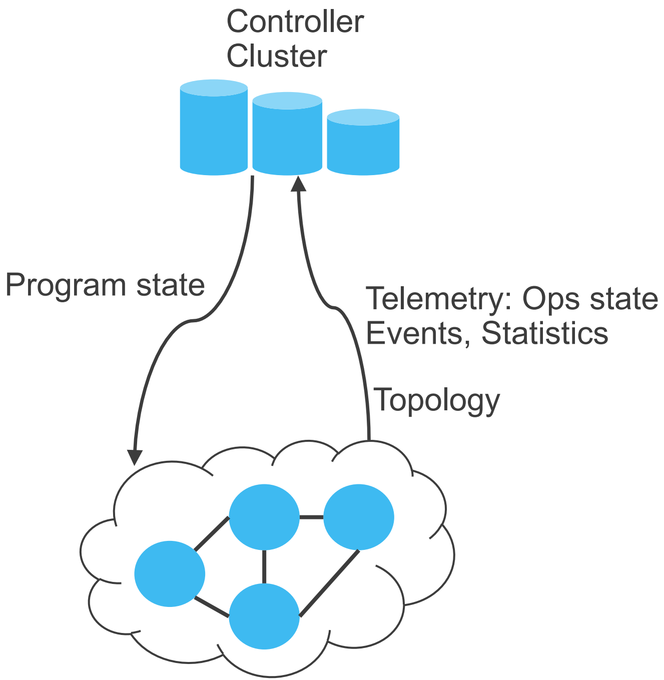
JUNIPER NETWORKS

Agenda

- Motivation
- Summary of RSVP-TE features
- Coupling RSVP-TE and shared MPLS forwarding plane
- Pop&Go traffic engineering details
- Reference implementation
- Summary

TE + SR: The infrastructure

- Controller is required for bandwidth accounting
  - Build or Buy? Deployment model…
- Learning the topology, events, capabilities
  - BGP-LS, IGP, LSDB streaming,…
- Path computation; Programming state…
  - PCEP, (BGP) SR-TE, programmable APIs,…
- Telemetry for traffic (feedback loop)
  - Link utilization, path optimization, where is my traffic?
- Further considerations: Label stack push depth, Readable label stack depth, Entropy, MPLS MTU
- Loose hop routing and ECMP
- Operate in a multi-vendor network



RSVP-TE: Feature Richness

- RSVP-TE at scale with distributed path computation
- Bandwidth admission control
- Fast Reroute (FRR)
- Auto-bandwidth for periodic bandwidth adjustments
- LSP priorities / preemption
- TE++ (Container LSP)
- Multicast (P2MP)

Widely deployed in many cloud/service provider backbone networks

RSVP-TE: The tale of two states

- Control plane state
  - RSVP-TE scaling work makes handling signaling state more efficient (https://datatracker.ietf.org/doc/draft-ietf-teas-rsvp-te-scaling-rec/)

- Forwarding plane state
  - Number of label routes proportional to the number of transit LSPs
  - Transit data plane churn
    - Label can change at every hop during make-before-break
  - Limited by platforms with smaller L-FIB space

RSVP-TE + SR MPLS Data Plane: Benefits

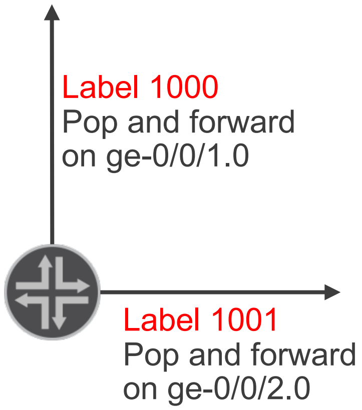
- *Coupling* feature richness of RSVP-TE control plane with Segment Routing MPLS forwarding plane (https://datatracker.ietf.org/doc/draft-ietf-mpls-rsvp-shared-labels/)
  - Reduce transit data plane state at LSR
  - Shared (*static*) forwarding plane across LSPs
  - No data plane programming at transit during LSP setup or teardown
  - Self-contained solution to automatically overcome label stack push and read limitations
  - Continue to use existing RSVP-TE control plane features
- Range of RSVP-TE data plane:
  - Current data plane (higher FIB state, granular LSP statistics at transit) to…
  - SR MPLS data plane (least FIB state, loss of granular LSP statistics at transit)

Allocation of TE-link Labels

For each TE-link

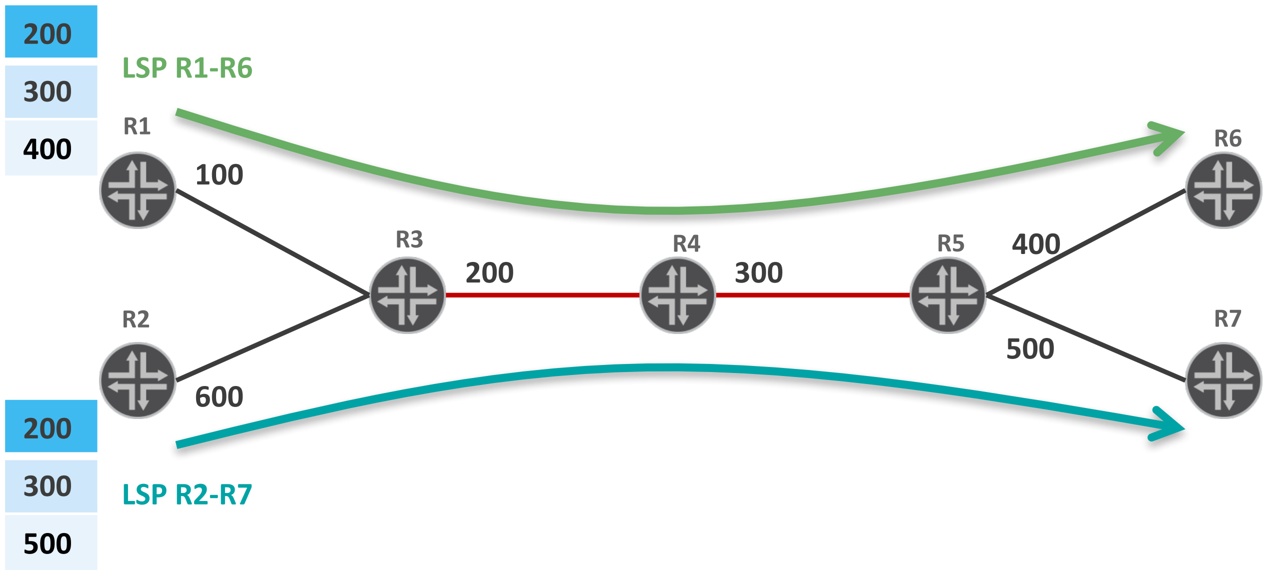
- Allocate a label
- Install the label route in LFIB with a MPLS next-hop action of *pop and forward* over that RSVP interface to the neighbor
- Separate labels for protected and unprotected LSPs

Number of TE-link labels ~= Number of RSVP neighbors



Shared Forwarding Plane Across LSPs

- Ingress can signal a LSP requesting TE-link labels
- Ingress pushes the received labels (recorded in Resv RRO) in a label stack
- Multiple LSPs at an ingress can have same label stack and account for statistics

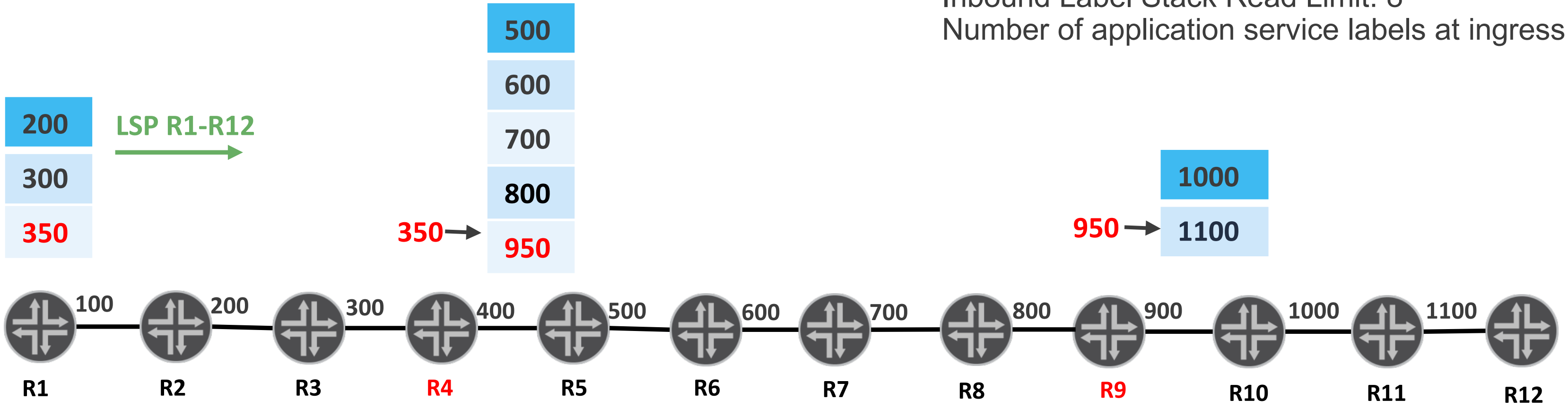


Delegating Label Push to Manage Stack Size

- Manage ingress label stack depth limit by offloading work to transit hop(s)

Labels in **RED** are **delegation** labels
Labels in **BLACK** are **TE-link** labels (for the next-hop neighbor)

Assumptions (for all nodes)
Outbound Label Stack Depth Limit: 5
Inbound Label Stack Read Limit: 8
Number of application service labels at ingress: 2



Delegation hops R4 and R9 allocate a delegation label to represent a set of labels that will be pushed

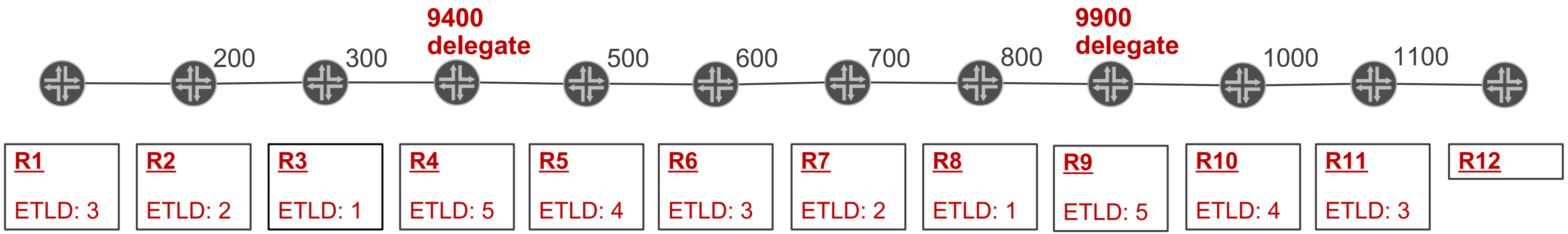
Transit LSR does not receive more labels than it can read and use for traffic hashing

Automatic / Explicit Delegation

- Automatic Delegation
  - Ingress LER lets the downstream LSRs *automatically* pick suitable delegation hops during the initial signaling sequence.
    - Ingress *does not need to be aware* up front of the label stack push and read limits of each of the transit LSRs
    - Delegation hops are picked based on a per-hop signaled attribute called the Effective Transport Label-Stack Depth (ETLD)
- Explicit Delegation
  - Ingress LER explicitly delegates one or more specific transit LSRs to handle pushing labels for a certain number of downstream hops
    - Ingress *needs to be aware* of the label stack push and read limits of each of the transit LSRs prior to initiating the signaling sequence

Automated delegation hop selection

Pop&Go tunnel R1-R12: Delegation hops R4 and R9 automatically chosen during Path signaling sequence

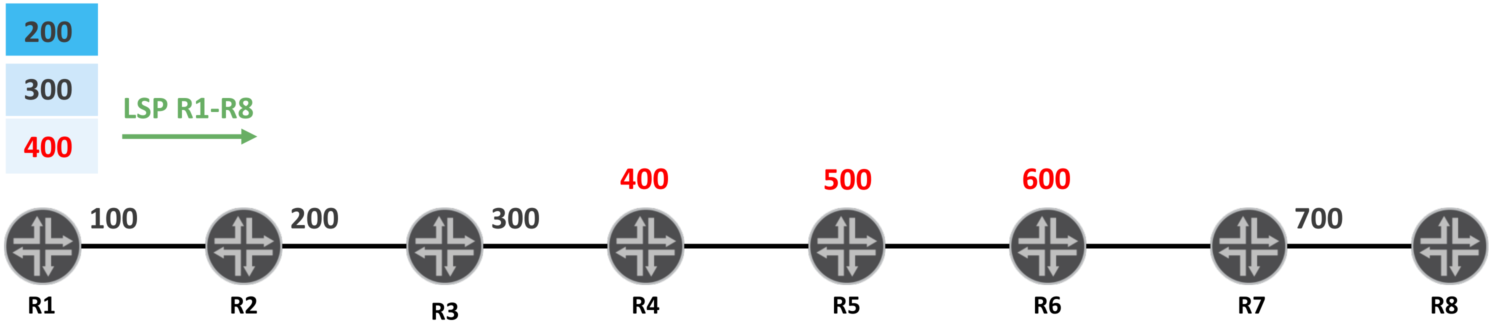


- ETLD (per-hop signaled attribute) in Path message:
  - Ingress populates ETLD with the maximum number of transport labels that it can *potentially send* to its downstream hop
  - Each successive hop decrements it by 1 (*or appropriately based on limitations at that hop e.g. inbound read limit depth*)
  - If a node is reached where the received ETLD is 1 (or no ETLD is received), then that node selects itself as delegation hop
  - Each *delegation hop* resets the ETLD to the maximum number of transport labels that it can potentially send to its downstream hop
  - *When the Path message reaches the egress, all delegation hops are elected.*

Backwards Compatibility

- Works if LSRs provide regular swap labels

  Labels in **RED** are conventional RSVP **swap** labels
  Labels in **BLACK** are RSVP **TE-link** labels (for the *next-hop neighbor)*

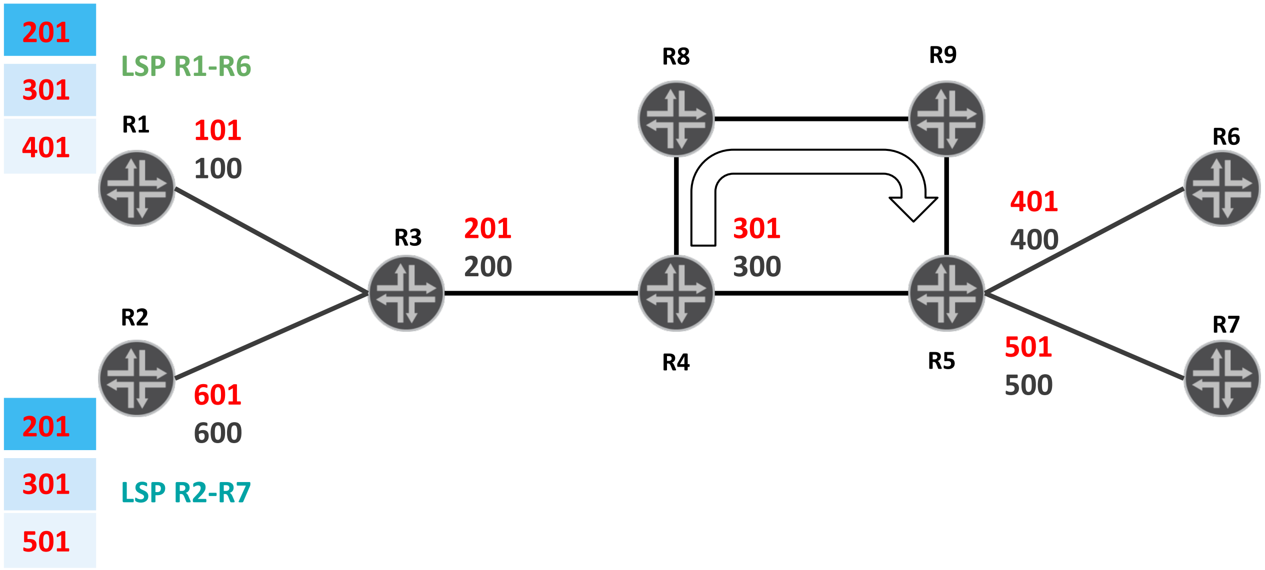


- Ingress constructs label stack by parsing RRO and checking type of each label

Link Protection

- Facility bypass link protection works naturally

Labels in **RED** are TE-link labels for link-protected LSPs
Labels in **BLACK** are TE-link labels for unprotected LSPs



- Regular bypass at transit R4
- At LSR R4
  - Primary: Pop label 301 and forward via R4-R5 link
  - Backup: Pop label 301 and send via bypass to R5

Show output: Shared MPLS Forwarding Plane

```
user@D# set protocols mpls label-switched-path <name> pop-and-forward
```

```
user@D> show mpls lsp terse

...
Transit LSP: 64000 sessions
Total 64000 displayed, Up 64000, Down 0
```

```
user@D> show rsvp pop-and-forward [label|detail|extensive]
RSVP pop-and-forward: 1 shared labels
Label-in      Hop-count   Next-segment- Protection     Session-
                          label                        count

300           1                         unprotected    64000
```

```
user@D> show route table mpls.0
mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 00:07:56, metric 1
                      Receive
1                  *[MPLS/0] 00:07:56, metric 1
                      Receive
2                  *[MPLS/0] 00:07:56, metric 1
                      Receive
13                 *[MPLS/0] 00:07:56, metric 1
                      Receive
300                *[RSVP/7/1] 00:03:35, metric 1
                    > to 56.56.56.2 via ge-0/0/1.0, Pop
300(S=0)           *[RSVP/7/1] 00:03:35, metric 1
                    > to 56.56.56.2 via ge-0/0/1.0, Pop
```

Outputs are subject to change

Show output: Automatic Delegation

Outputs are subject to change

LSP Path: 7-hop path auto-delegating to the 3rd hop

```
Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 70)
80.1.1.2 S 50.1.1.2 S 70.1.1.2 S 92.1.1.1 S 93.1.1.2 S 102.1.1.2 S 100.1.1.2 S
Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
             (Labels: P=Pop D=Delegation)
3.3.3.3(flag=0x20) 80.1.1.2(Label=299776, P) 4.4.4.4(flag=0x20) 50.1.1.2(Label=299792, P)
5.5.5.5(flag=0x20) 70.1.1.2(Label=299856, D) 6.6.6.6(flag=0x20) 92.1.1.1(Label=299936, P)
7.7.7.7(flag=0x20) 93.1.1.2(Label=299872, P) 8.8.8.8(flag=0x20) 102.1.1.2(Label=299792, P)
9.9.9.9(flag=0x20) 100.1.1.2(Label=3)
```

RIB entry at ingress …

```
9.9.9.9/32 (1 entry, 1 announced)
  Label-switched-path t1
  Label operation: Push 299856, Push 299792, Push 299776(top)
```

LFIB entry at transit (delegation hop) 5.5.5.5...

```
299856 (1 entry, 1 announced)TSI:KRT in-kernel 299856 /52
*RSVP   Preference: 7/1
  Next hop: 92.1.1.1 via ge-0/0/2.0, selected
  Label operation: Swap 299792, Push 299872, Push 299936(top)
```

Summary

- Traffic Engineering is not a best effort activity
  - Planning, bandwidth and resource management, efficient link utilization
  - Predictable network behavior
- RSVP-TE scalability and usability has improved significantly over the years
- Pop&Go offers a pragmatic path to leverage the RSVP-TE control plane coupled with the minimal state of the shared forwarding plane
- Request for your support

References

- IETF MPLS-WG Draft
  https://datatracker.ietf.org/doc/draft-ietf-mpls-rsvp-shared-labels/
- Blog
  https://forums.juniper.net/t5/Industry-Solutions-and-Trends/RSVP-TE-Pop-amp-Go-Using-a-shared-MPLS-forwarding-plane/ba-p/312918

Thank you

hsitaraman@juniper.net
vbeeram@juniper.net