Recent Advances in MPLS Traffic Engineering

Solutions to operational challenges in deploying RSVP-TE

SANOG27

Chandrasekar Ramachandran (csekar@juniper.net)

Agenda

- Why RSVP-TE?
- What are the operational challenges?
- Solutions

Why RSVP-TE?

Motivations for operators deploying RSVP-TE

- Bandwidth reservation
  - Motivation has been to increase n/w utilization
- Fast Re-route
  - Local repair: minimal traffic loss (of the order of milliseconds)
  - LSP re-optimization: hitless switch to optimal path (make-before-break)
- Prioritizing Traffic
  - During bandwidth contention after link degradation (AE member link failures)
- Class-based Forwarding (CBF)
  - Placing different types of traffic on different CBF paths

Operational challenges & Solutions

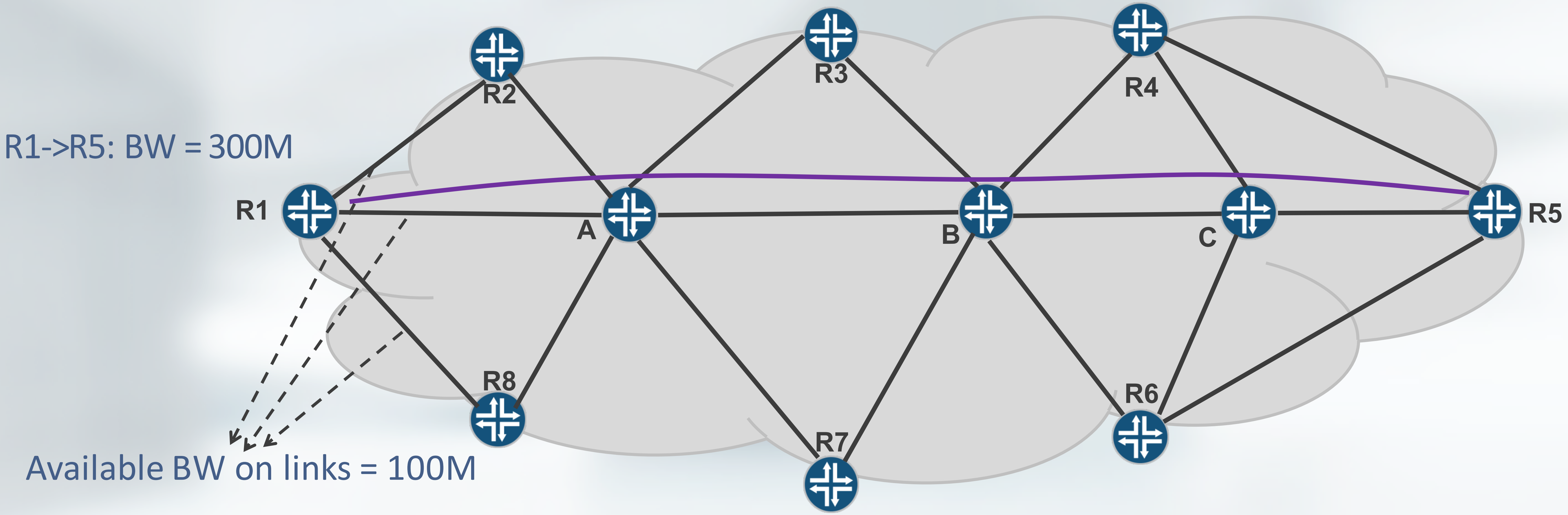
- *Observed in practice at scale*

#1: LSP Provisioning & Load Balancing

LSP Provisioning & Load Balancing

- Auto-bandwidth well deployed RSVP-TE feature
  - Operator configures initial bandwidth
  - Ingress LERs re-sizes LSP bandwidth based on traffic rate
    - Re-sizing decision based on LSP stats
    - Helps reduce operational overhead
    - Allows operators to increase utilization of links in the network

LSP Provisioning & Load Balancing (contd.)



R1-R5 LSP needs BW > avail BW on all possible paths

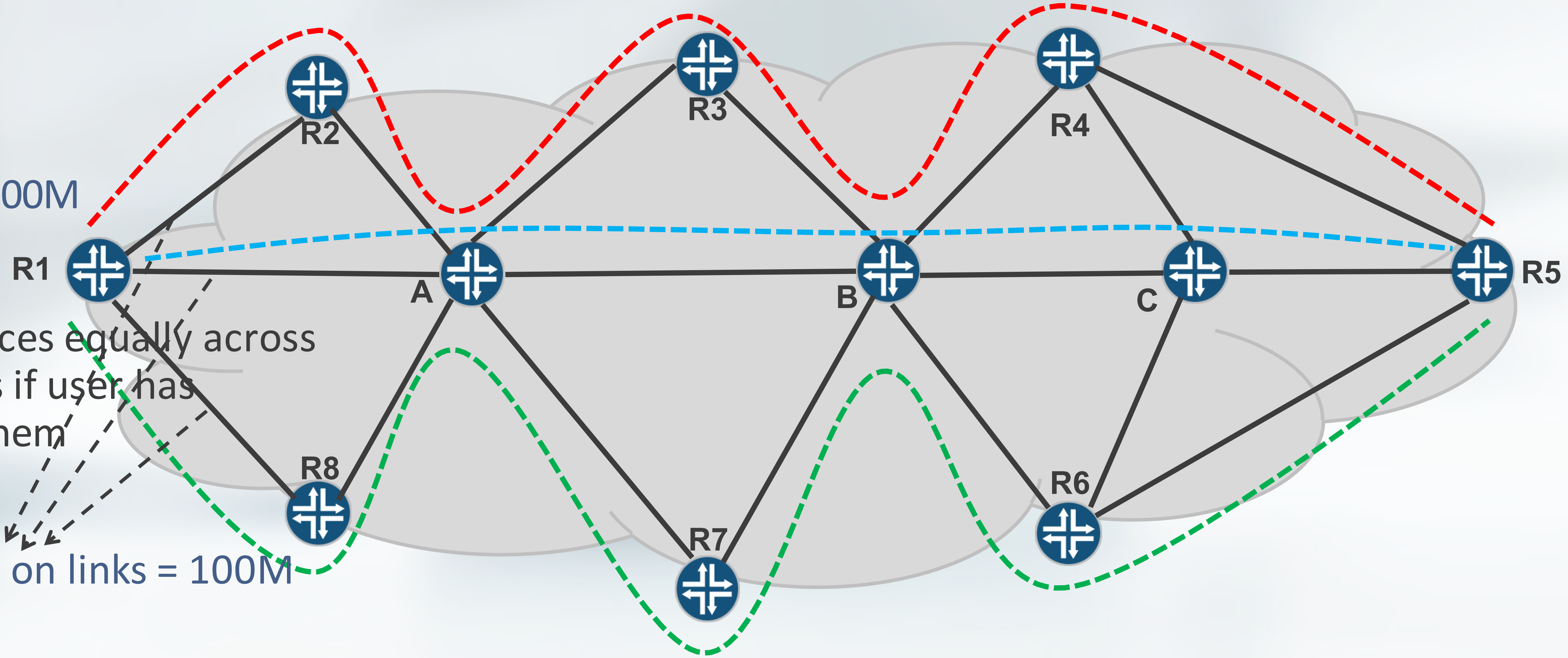
Operator intervention required: but how many LSPs to provision? When to de-provision them?

LSP Provisioning & Load Balancing (contd.)

R1->R5: BW = 300M

R1 load balances equally across members – as if user has provisioned them

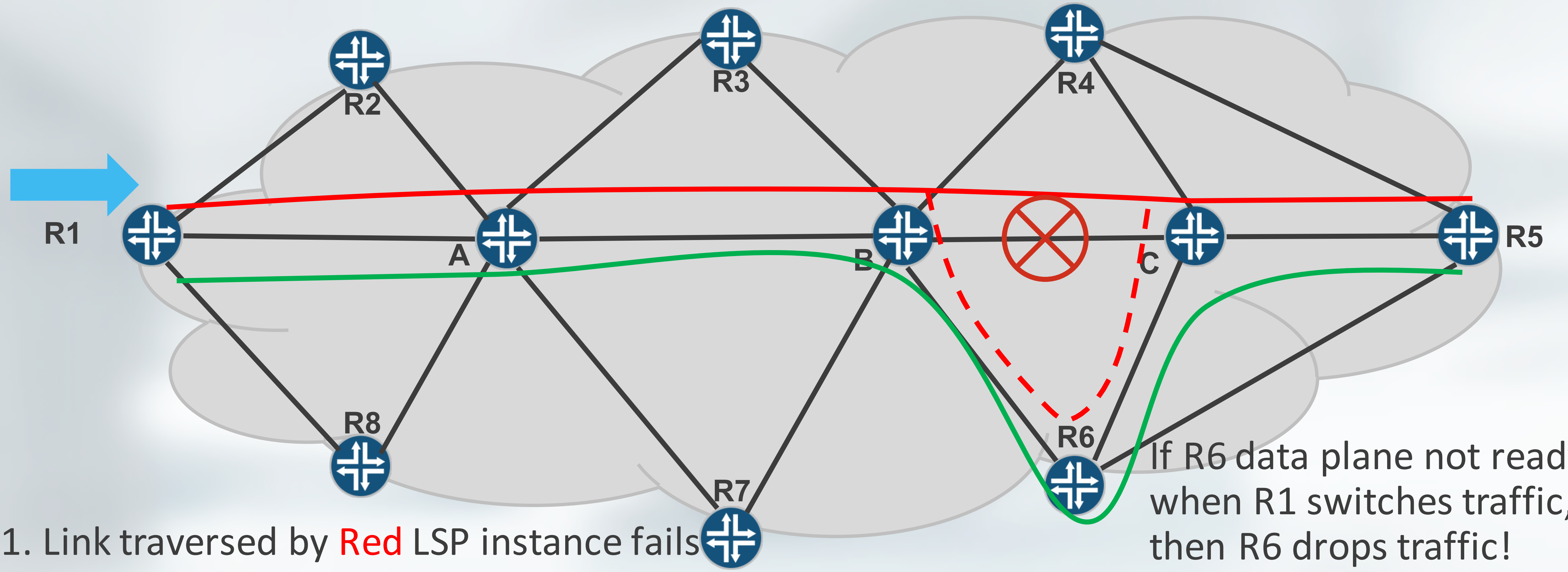
Available BW on links = 100M



- Operator provisions "container" LSP on R1 with R5 as LSP destination
- R1 automatically "splits" 300M request across 3 "member" LSPs (each 100M)
- R1 automatically "merges" to 2 member LSPs if LSP BW reduces to 200M

#2: Fast Re-route: Traffic loss during LSP Re-optimization

FRR: Traffic Loss during Re-optimization

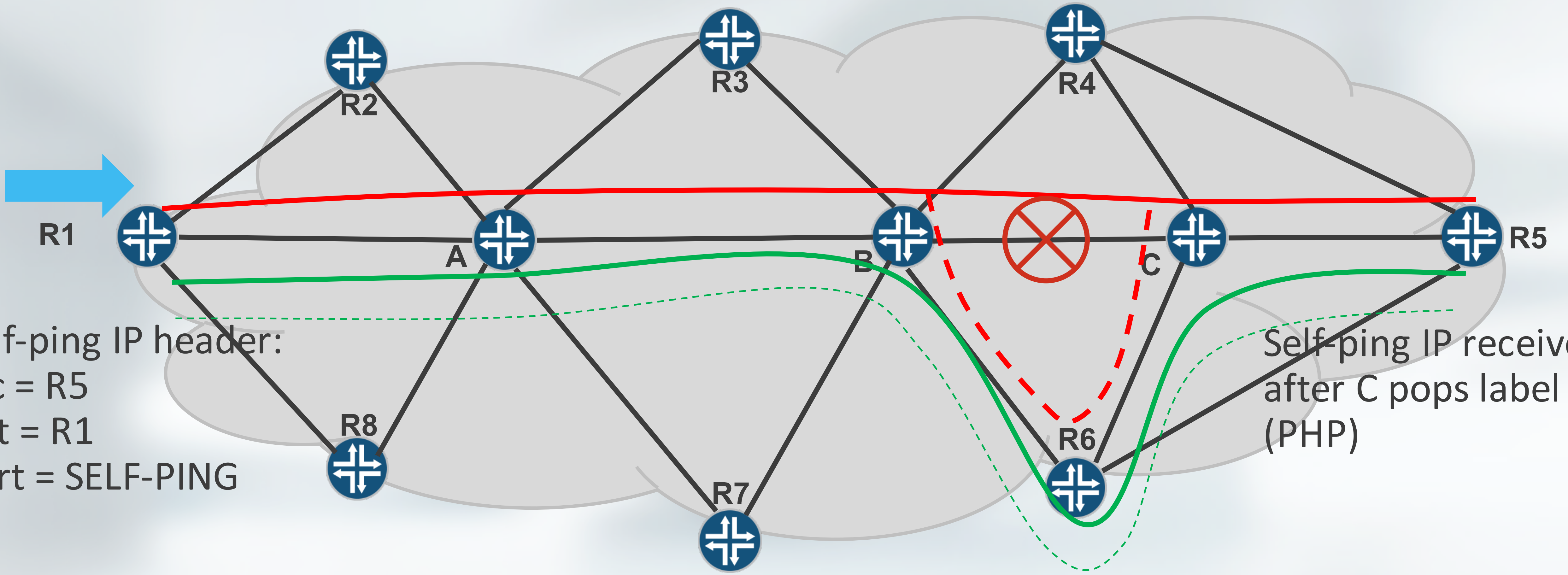


1. Link traversed by Red LSP instance fails
2. Red LSP instance is locally repaired
3. R1 computes & signals Green LSP instance in “make-before-break” fashion
4. R1 switches traffic to Green LSP instance

If R6 data plane not ready when R1 switches traffic, then R6 drops traffic!

- Configuring switchover delay
  - scale dependent (operational challenge)
- Send LSP ping probes before switching
  - S/w on R5 must send LSP ping replies

FRR: Traffic Loss during Re-optimization (contd.)



Self-ping IP header:
Src = R5
Dst = R1
Port = SELF-PING

Self-ping IP received after C pops label (PHP)

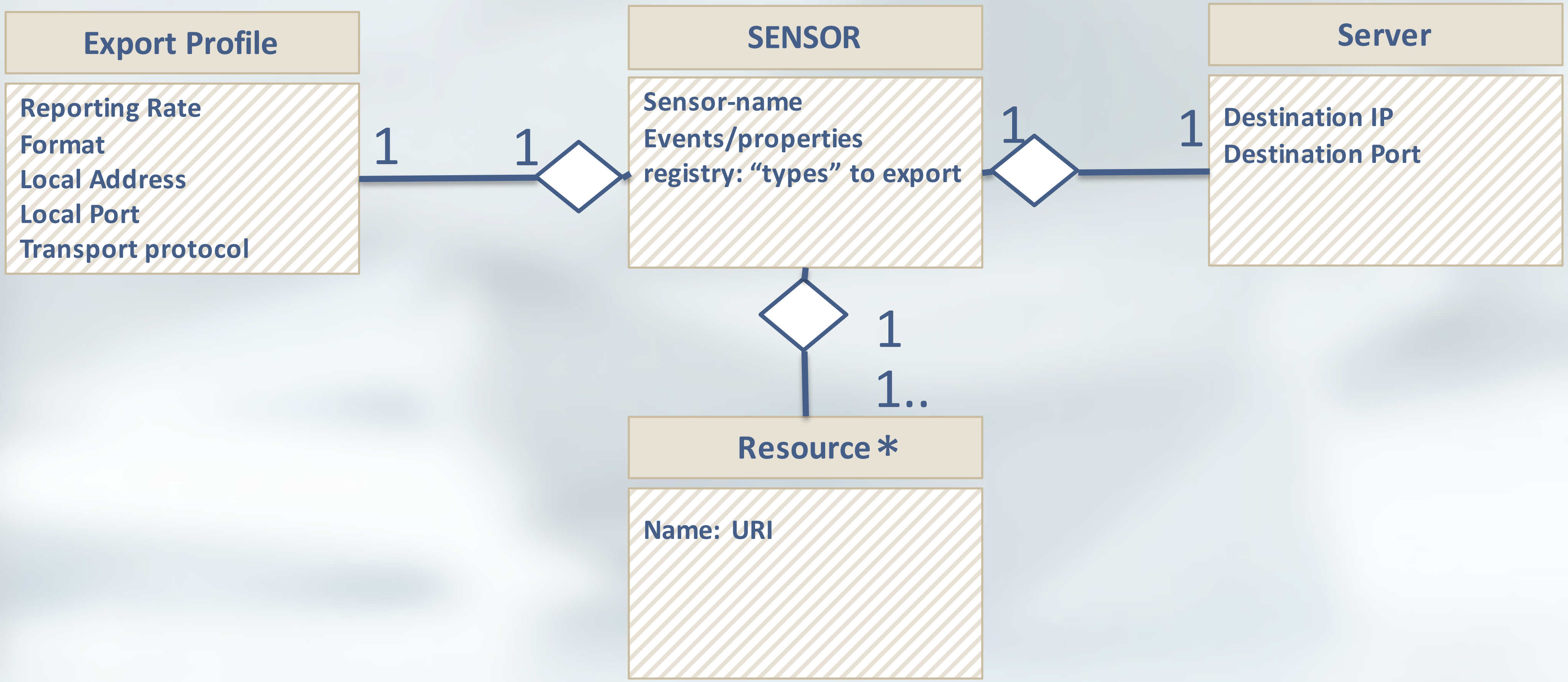
4. Ingress LER sends IP self-ping (UDP packet) over Green LSP instance
5. When data plane is ready, Egress LER receives self-ping packet
6. Egress LER ***hardware*** IP forwards the packet back to Ingress LER
7. R1 switches traffic to Green LSP instance

#3: Monitoring LSPs at scale

Monitoring LSPs at scale

- What events are happening on the LSPs? Which ones need operator attention? What properties of the LSPs are changing?
  - But, without relying on a polling mechanism (like SNMP)
- A push-based approach to export LSP events/properties to an off-router client:
  - Transmitter and Collector rely on the same message-template to <u>generate s/w code that transmits or parses the message</u>.
  - Transmitter: implemented by the router vendor
  - Collector: implemented by the router vendor or by the operator

LSP Telemetry: Object model



LSP **sensor:** basic unit of LSP- telemetry.
Tracks events/properties to export.
Export as per the **Export Profile**.

#4: Protocol Traffic load

Protocol Traffic load

- RSVP protocol relies on periodic refreshes to:
  - Synchronize new states along the LSP path
  - Recover from lost messages
    - Path message refreshes achieve state synchronization after message loss
    - Time out states to clean up states upon lost PathTear messages
- Periodic refreshes are problematic at scale
- Solution: RFC 2961
  - Increase refresh interval from default 30 seconds to a long interval

Protocol Traffic load – Solution

- What were missing in implementations preventing operators from setting arbitrarily long refresh intervals?
  - Reliable delivery of tear down & error messages (RFC 2961)
  - Bind the fate of LSP state to the state of Hello sessions
  - Utilize acknowledgements (or lack of them) for flow control

Conclusions

- It is possible to deploy RSVP MPLS-TE at scale that:
  - Are easy to provision
  - Can resize automatically
  - Can utilize network more effectively
  - Are easy to monitor
  - Are well-behaved and not chatty

- Current status of the solutions
  - Implementation already shipping
  - Already under trials for operator deployments

References

- RSVP-multipath/TE++:
  - http://www.juniper.net/assets/us/en/local/pdf/whitepapers/2000587-en.pdf
- Entropy Labels:
  - https://tools.ietf.org/html/rfc6790
- Self-ping:
  - https://tools.ietf.org/html/draft-ietf-mpls-self-ping-06
- RSVP-TE scaling best current practices:
  - https://tools.ietf.org/html/draft-ietf-teas-rsvp-te-scaling-rec-00
  - https://tools.ietf.org/html/draft-chandra-mpls-ri-rsvp-frr-02

Thank You!