January 2020

SEGMENT ROUTING POLICIES WITH BUILT-IN RESILIENCY

Clarence Filsfils

Zafar Ali

Swadesh Agrawal

Francois Clad

Jose Liste

See next page for additional authors

Inventor(s)
Clarence Filsfils, Zafar Ali, Swadesh Agrawal, Francois Clad, Jose Liste, and Jiri Chaloupka

SEGMENT ROUTING POLICIES WITH BUILT-IN RESILIENCY

AUTHORS:
Clarence Filsfils
Zafar Ali
Swadesh Agrawal
Francois Clad
Jose Liste
Jiri Chaloupka

ABSTRACT

Techniques are described herein to establish hierarchical Segment Routing (SR) policies with built-in resiliency. These techniques increase the scalability of the SR Path Computation Element (PCE) and SR network. Specifically, once such policies are installed by the SR PCE, the SR PCE does not have to take any action in the event of failure. This is an improvement over current approaches, which employ the SR head-end node as a single point of failure.

DETAILED DESCRIPTION

One of reasons why Segment Routing (SR) is considered a replacement for GTTP is its ability to integrate service chaining (which allows for Network Functions Virtualization (NFV) for security, billing, etc.) with traffic engineering. In an SR network, SR policies are used to steer a packet through a set of service or topological segments. Another advantage of SR over GTTP is that multiple sessions can be aggregated into a single SR policy (shared Segment Identifier (SID) lists of service and topological segments).

Each SR policy has a Binding SID (BSID) that can be used to steer traffic into the policy. The BSID can be used to stitch SR policies without requiring specific configuration on the stitching border nodes. An ingress node can include the BSID of a remote SR policy in the SID list of its local SR policy to create hierarchical policies. Typical use-cases for hierarchical policies include SID list length reduction (to address platform encapsulation restrictions), scalability, domain isolation, and opacity and transport interworking for SR Multiprotocol Label Switching (SR-MPLS) and Segment Routing over Internet Protocol version 6 (IPv6) (SRv6) policies.

However, as illustrated in Figure 1 below, the SR head-end node is currently a single point of failure. A hierarchical SR policy P1 is using a service policy P2 to steer the packet over a service operated at node 4 before exiting domain 2. The policy P2 in domain 2 may be used for SRv6/MPLS interworking when domains 1 and 3 are SRv6 and domain 2 is SR-MPLS (or vice versa). One of the main issues with node 3 being a single point of failure is that the SR Path Computation Element (PCE) needs to recompute the transit SR policies. This process is very slow and not scalable.

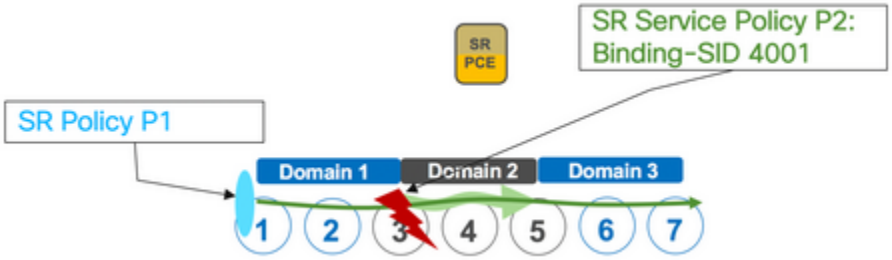

Figure 1

Accordingly, techniques are described herein to address the aforementioned requirements without relying on Topology Independent Loop Free Alternate (TI-LFA) or path protection. These techniques may rely on Interior Gateway Protocol (IGP) fast convergence and hence may be simpler and scale better than alternative approaches. They also enable the SR PCE to compute a multi-domain path that has built-in end-to-end resiliency. In one example, techniques described herein introduce the notion of an anycast BSID. The use of an anycast BSID provides built-in resiliency that protects the end-to-end policy from failure of the head-end node of an SR policy. This may also be used when the anycast SID is the entry point in a service chain.

Figure 2 below illustrates an example use case implementing the techniques described herein. Here, a low latency policy P1 needs to proceed through a service S5 in SR-MPLS domain 2. The end-to-end low latency path needs to proceed through node 2 in domain 1 and TE node 8 in domain 3 while applying service S5 in domain 2.

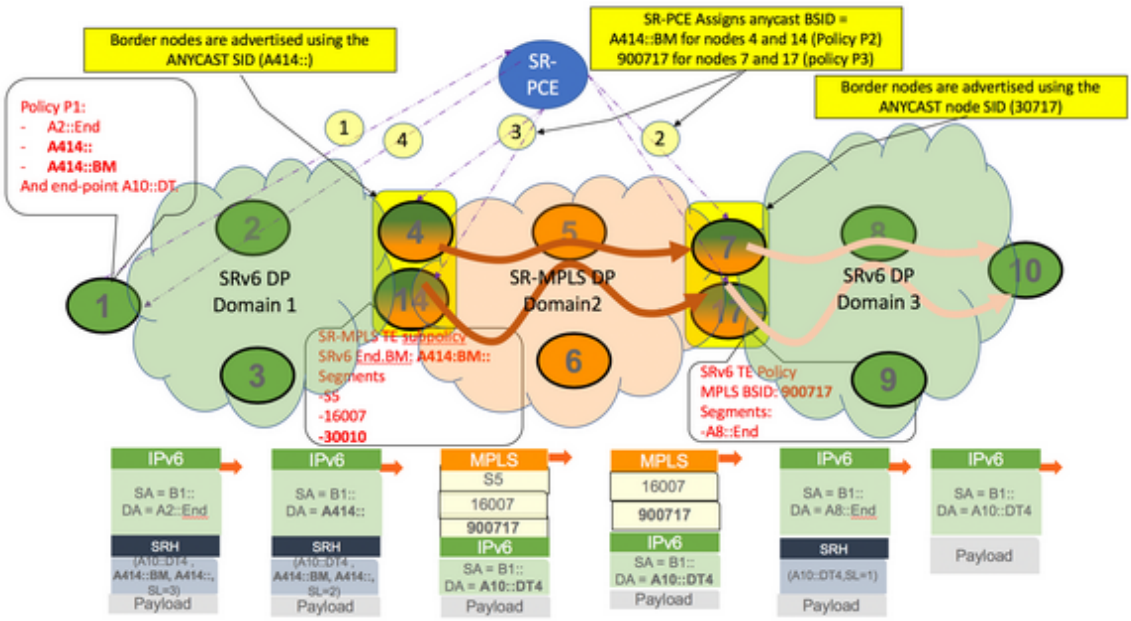

Figure 2

Node 1 cannot compute an end-to-end low latency path that proceeds through service S5 in SR-MPLS domain 2. As such, node 1 requests a Software Defined Networking (SDN) controller (e.g., SR PCE) to compute an end-to-end path, providing the necessary information (e.g., low latency requirement, transit service requirement, resiliency requirement, constraints, etc.).

The SDN controller computes the end-to-end path and determines that the end-to-end path requires transit through both SR-MPLS domain 2 and SRv6 domain 3. Accordingly, the SDN controller instantiates two transit policies for traversing the service domain 3: one on node 7 and another one on node 17. The SDN controller assigns the same BSID value to both of these (transit) policies (BSID = 900717). If the transit anycast BSID policies are already instantiated as part of pre-planning of the network, this step may be skipped.

Similarly, the SDN controller may instantiate two transit policies for traversing service domain 2: one on node 4 and another one on node 14. The SDN controller assigns the same BSID value to both (transit) policies (BSID = A414:BM::). The binding SID is the END.BM function. If the transit anycast BSID policies are already instantiated as part of pre-planning of the network, this step may be skipped.

The SDN controller builds the end-to-end SID list that contains the anycast SID shared by nodes 4 and 14 (A414::), directly followed by the transit policy anycast BSID A414:BM:: (transit SR policies). This SID list ends with a Virtual Private Network (VPN) SID on endpoint node 10 and may include other intermediate segments in any of the edge domains in order to meet the optimization objective or constraints indicated by node 1 (in this example, Node A2::End). The SDN controller then provides node 1 with a complete list of SRv6 segment <A2::End, A414::, A414::BM, A10:DT>, and end-point A10::DT.

Figure 3 also illustrates the path traveled by the packet through the network. This example involves an SR-MPLS/SRv6 interworking scenario, but it will be appreciated that the techniques described herein may be applicable to any suitable multi-domain scenario.

The combination of anycast SIDs with a shared BSID for transit policies protects the end-to-end SR policy against the failure of any domain border routers, thereby providing built-in resiliency for the system. Failure within a domain may be addressed through TI-LFA. As illustrated in Figure 3 below, should one of the border nodes fail or becomes unreachable, IGP convergence automatically reroutes the traffic to the other border router in the anycast group, which may steer the traffic into an equivalent transit policy.

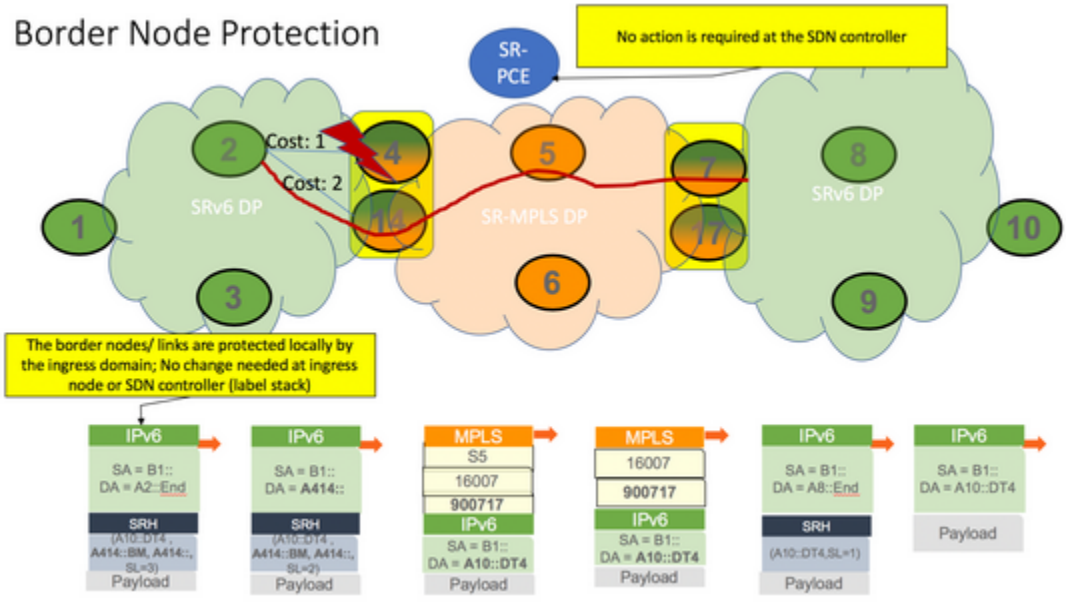

Figure 3

The same concept is applicable when several transit policies are stitched together in an end-to-end SR policy. No change is required in the SID list imposed on the packet by

the original head-end node. Once the controller sets up the policies, it does not have to reinstall them (e.g., in the event of a failure in the network). There is no need for implementing reoptimization procedures at the SR policy level, either. The procedure also works naturally for one-to-N protection (unlike path protection, which is mostly limited to one-to-one). The network itself handles all resiliency aspects in a distributed fashion, which improves the scalability.

In summary, techniques are described herein to setup hierarchical SR policies with built-in resiliency. These techniques increase the scalability of the SR PCE and SR network. Specifically, once such policies are installed by the SR PCE, the SR PCE does not have to take any action in the event of failure. This is an improvement over current approaches, which employ the SR head-end node as a single point of failure.