SR-PMTU for SR Policy

draft-peng-spring-pmtu-sr-policy-00

Authors:
Shuping Peng, Huawei
Dhruv Dhody, Huawei
Ketan Talaulikar, Arrcus
Gyan Mishra, Verizon

Some history as to how did this draft came up in SPRING WG ??

- During the adoption of draft “draf-ietf-pce-pcep-pmtu-02”, a need for a document in SPRING was requested and confirmed by both PCE & SPRING WG chairs.
- IDR document “draft-ietf-idr-sr-policy-path-mtu-05” adoption call April 2020, a discussion was brought up by Ketan Talaulikar that the concept of “Path MTU” for SR Policy and its applicability should be first defined in SPRING WG before we introduce signaling aspects into BGP
- As a result of the PCE PMTU extension adoption call, the draft “draf-ietf-pce-pcep-pmtu-02” was to maintain only protocol extension specific details, while the SR Policy PMTU definition & framework was to be developed in SPRING WG as a Standards Track document to ensure vendor interoperability related to SR-PMTU concepts and computation details.
- This topic came up as a critical issue to be addressed during IDR WG 2020 BGP SR Policy PMTU adoption call, and now again during PCE WG 2022 PCE PMTU adoption call mentioned above due to the criticality of solving this topic related to handling of fragmentation and fragmentation avoidance procedures using SR-PMTU.

SR-PMTU (Path MTU) for SR Policy Overview

Motivation for handling of SR-PMTU for SR paths includes but not limited to:

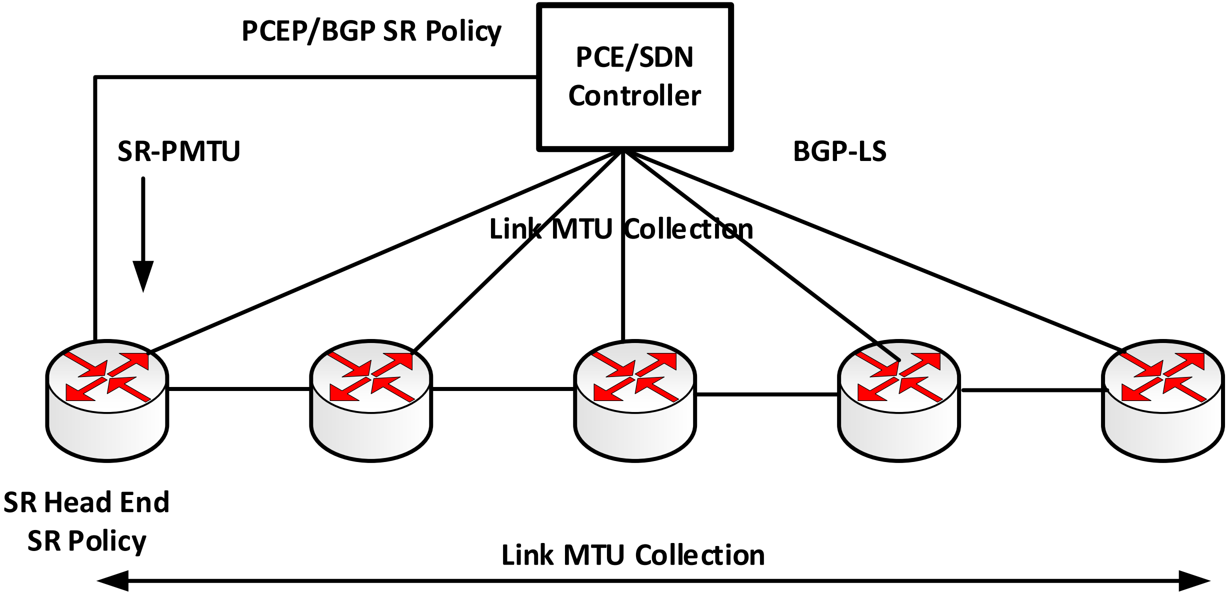
- **Fragmentation** avoidance by the SR headend by being aware of the SR-PMTU associated with all the SR paths and policies.
- Ability to generate **ICMP messages** at the head end.
- Follow correct fragmentation procedures.
- Ability to use SR-PMTU as a **path computation** constraint and optimization criteria at the headend or PCE/SDN Controller.

SR-PMTU Definition for SR Policy

A “Valid” candidate path is selected as the “Active” path once determined to be the best path for the SR Policy which can be dynamic/explicit composed of a set Segment-Lists or composite candidate path container grouping of SR policies.

- **SR-PMTU of a segment list** is defined as the minimum link MTU of all the links in the path from source to destination.
- **SR-PMTU of a candidate path** is the same as SR-PMTU of a Segment-List, for explicit/dynamic is expressed as a set of Segment-Lists and the SR-PMTU of the candidate path is defined as the minimum SR-PMTU of all the Segment-Lists in the set, for composite is the minimum SR-PMTU of all the constituent SR policies of the composite candidate path.
- **SR-PMTU of an SR Policy** is defined as the SR-PMTU of the selected/active candidate path

SR Policy SR-PMTU Framework



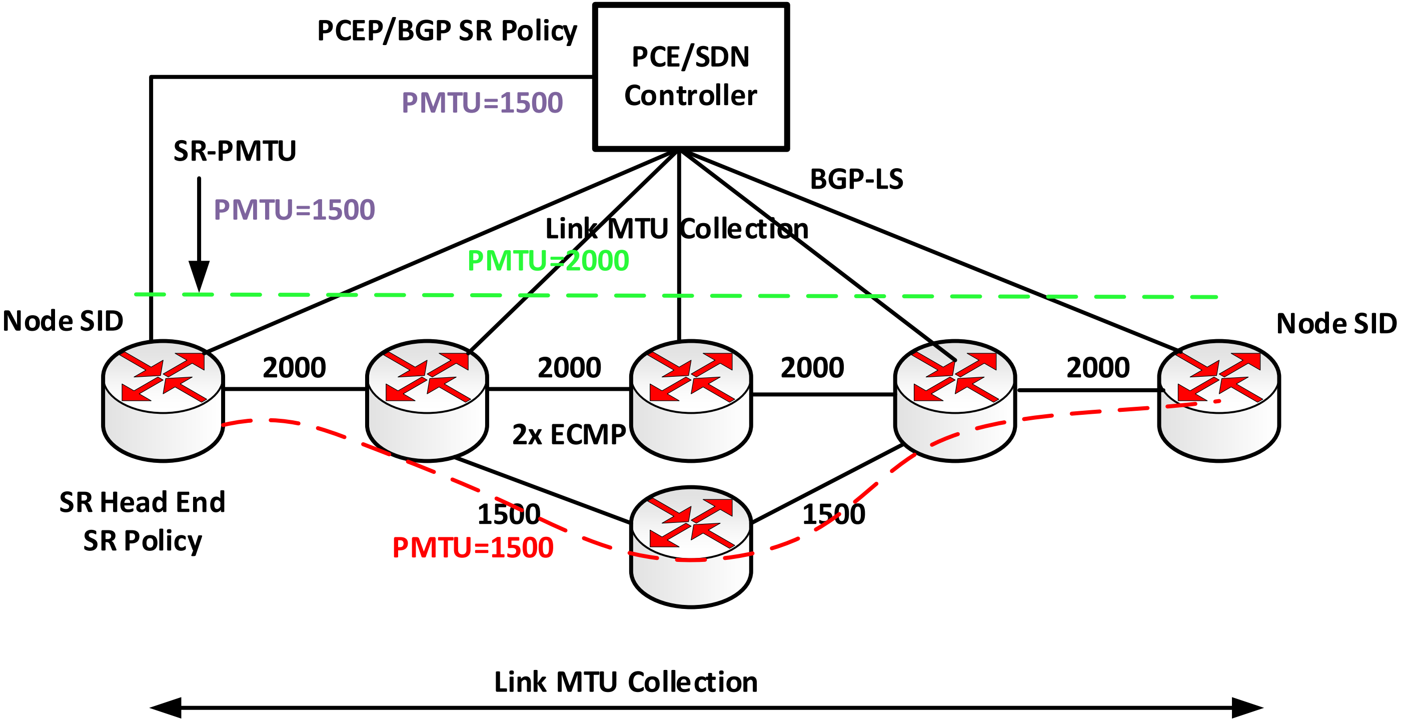
Framework of SR-PMTU for SR Policy

The framework of SR-PMTU for SR Policy includes 1st -link MTU collection, 2nd -SR-PMTU computation, 3rd -SR-PMTU Policy Enforcement as well as behavior handling on the headend.

Link MTU Collection: Minimum link MTU of all links along a path from source to destination.

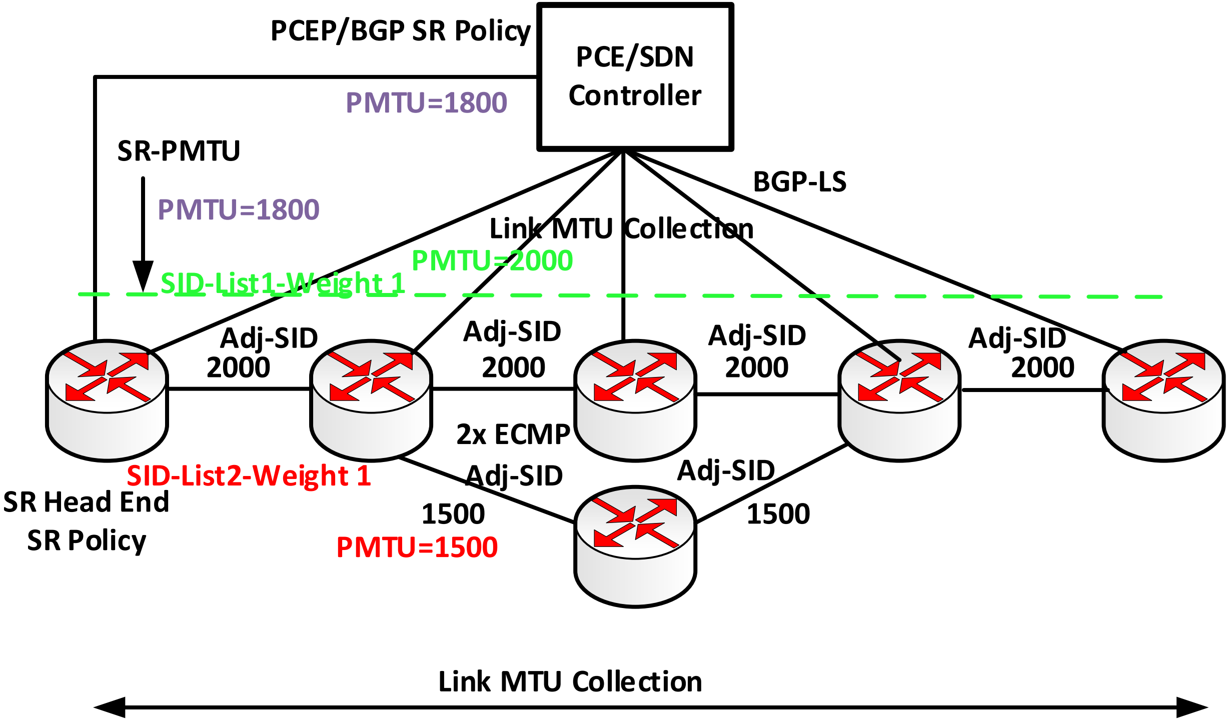
- SR-PMTU Computation:
  - **Loose TE path** – Minimum SR-PMTU of all ECMPs between two adjacent Node SIDs along SR-TE path
  - **Strict TE path** – Minimum link MTU of all the links along the Strict SR-TE Path
  - **Mixed TE Path** – Minimum SR-PMTU of all ECMPs between two adjacent Node SIDs and the ink MTU of all the links along the path indicated by the adjacency SID.
  - **Binding TE Path** – SR-PMTU of Binding path is same as that of SR Policy except that it includes the associated encapsulation overhead for SRv6 with outer IPv6 header & SRH, and SR-MPLS SID list label stack pushed onto the packet.
  - **TI-LFA TE Path** – SR-PMTU of the repair path at the PLR node to the merge point computed by the controller which then updates the head end with the new SR-PMTU.   FRR overhead can be taken into consideration for fragmentation avoidance as TI-LFA configurable PMTU value.

SR-PMTU ⬄ SR Policy Loose Path

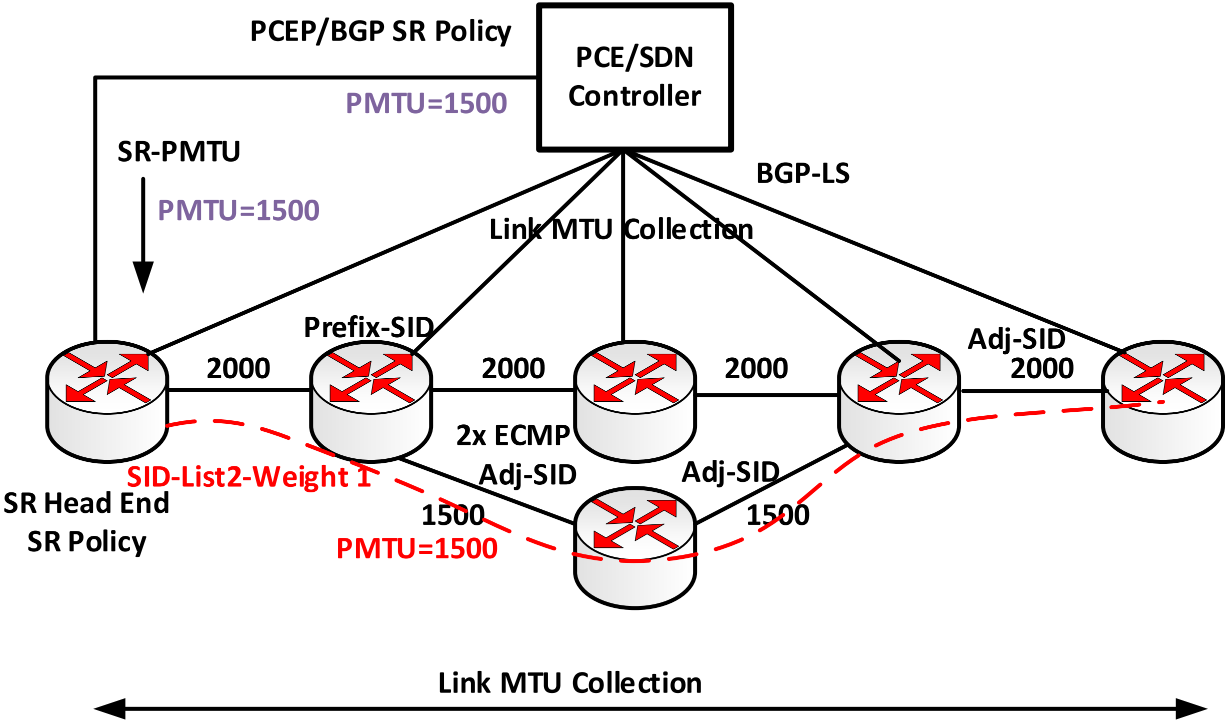


SR PMTU ⇔ SR Policy Strict Path

- SR policy can be configured with the min SR-PMTU constraint value as 1800 to make sure that the path chosen by the controller meets this PMTU criteria.
- The PCE/controller would then pick the strict path SID-List1 with Adj-SID corresponding to green as shown in the figure
- The other path does not meet the SR-PMTU constraints!



SR-PMTU ⬄ SR Policy Mix Path (Strict / Loose)



SR-PMTU Key Procedures

SR-PMTU Enforcement, Handling, Optimizations etc.

- **SR-PMTU Enforcement** – SR-PMTU can be encoded in the SR policy, after computation the SR Policy -SR-PMTU is enforced.
- **Headend Behavior Handling** – After SR-PMTU is enforced, the headend will perform handling behaviors, encapsulation, fragmentation etc.
- **Constraints & Optimizations** – SR-PMTU constraint is set considering both carried services and encapsulation overhead. Once SR-PMTU is set at the headend, it is supposed to be the lowest bound of the SR-PMTUs of all paths being computed locally or enforced by the controller to avoid fragmentation.
- **Fragmentation processing** – If SR-PMTU of all paths being computed locally or enforced by the controller are smaller than the SR-PMTU constraint set on the headend, fragmentation will be handled, and if not possible, headend will generate ICMP to notify the traffic source. Over this selected path, packets are fragmented in order to guarantee the size of the encapsulated packets are smaller then the PMTU of the selected path.
- **SRv6 Specific Handling** – SRH is included in the calculation of the link MTU and SR-PMTU.

Next Steps & Questions to be discussed?

- SR-PMTU TI-LFA computation & fragmentation caveats?
- SR-PMTU SRv6 source node encapsulation and fragmentation caveats?
- SR-PMTU Binding SID path computation caveats?

Spring Working Group Review & Feedback ??

Thank You!