

SPRING Working Group
Internet-Draft
Intended status: Informational
Expires: 29 December 2025

A. Karboubi, Ed.
C. Alaettinoglu
H. Shah, Ed.
S. Sivalaban
T. Defillipi
Ciena
P. Maheshwari
Airtel India
27 June 2025

Circuit Style Segment Routing Policies with Optimized SID List Depth
draft-karboubi-spring-sidlist-optimized-cs-sr-03

Abstract

Service providers require delivery of circuit-style transport services in a segment routing based IP network. This document introduces a solution that supports circuit style segment routing policies that allows usage of optimized SID lists (i.e. SID List that may contain non-contiguous node SIDs as instructions) and describes mechanisms that would allow such encoding to still honor all the requirements of the circuit style policies notably traffic engineering and bandwidth requirements.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 29 December 2025.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Terminology	3
3. Problem statement: Issues with SID list compression	4
3.1. Deviation due to failures	5
3.2. Deviation due to repairs	6
4. Dealing with deviation due to failures	7
4.1. Head end node procedure	8
4.2. Controller/PCE component procedure	9
4.3. Eligibility control attribute	9
5. Dealing with deviation due to repairs/changes	9
6. Protocol and model changes	10
6.1. Active candidate path selection	10
6.2. PCEP extensions	10
6.3. SR policy Yang changes	11
7. IANA considerations	11
8. Security considerations	11
9. References	11
9.1. Normative References	11
9.2. Informative References	11
Acknowledgements	12
Authors' Addresses	12

1. Introduction

Service providers require delivery of circuit-style transport services in a segment routing based IP network.

[I-D.ietf-spring-cs-sr-policy] introduces a solution that supports circuit style SR policies. However, the solution mandates using a fully specified SID list where the path is encoded using persistent or manually configured adjacency SIDs. Using a fully specified SID list causes a very large segment stack that may not be feasible for low-end edge devices often found in access networks.

This document presents a solution that removes the fully specified SID list requirement while still maintaining the key features presented in [I-D.ietf-spring-cs-sr-policy]. It enables use of compressed SID list (i.e. allows the use of node SIDs) in circuit-style SR policies.

[I-D.ietf-spring-cs-sr-policy] defines circuit-style SR as an SR policy with the following characteristics:

- * Persistent end-to-end traffic engineered paths that provide predictable and identical latency in both directions
- * Strict bandwidth commitment per path to ensure no impact on the Service Level Agreement (SLA) due to changing network load from other services
- * End-to-end protection (<50msec protection switching) and restoration mechanisms
- * Monitoring and maintenance of path integrity
- * Data plane remaining up while control plane is down

Note that for some service providers the bidirectional co-routed paths may not be necessary.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Terminology

SID : Segment Identifier

SLA : Service Level Agreement

SR : Segment Routing

CS-SR : Circuit-Style Segment Routing

PCE : Path Computation Element

PCEP : Path Computation Element Communication Protocol

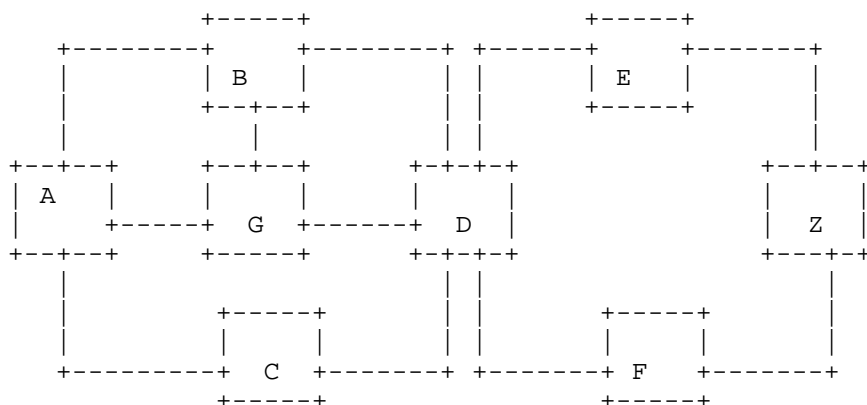
3. Problem statement: Issues with SID list compression

A PCE computes a path for the service according to the network state and available capacity at that time. These paths are referred to as intended paths. It then compresses the intended path into SIDs using a combination of node and adjacency SIDs as defined in SR architecture [RFC8402]. Nodes in the network forward packet to node SID N by using their IGP (or flex-algo) shortest paths to N. This is referred to as path expansion. At the time of installing the compressed SID list, this expansion and the intended path are identical.

However, network changes, particularly link and/or node failures and repairs may cause the intended path and this path expansion to deviate resulting in a service's traffic to use resources on a path that the PCE did not reserve any bandwidth on, causing service degradation for both this service and the other services on that path.

Both the failure and repair cases are illustrated using the example network topology of figure 1. An SR policy from node A to node Z with two diverse traffic engineered candidate paths was computed by PCE and signaled to head end node A resulting in the following intended paths and their respective compressed SID List:

- * Candidate path 1: intended path A-B, B-D, D-E, E-Z links and compressed as SID list B, E, Z
- * Candidate path 2: intended path A-C, C-D, D-F, F-Z links and compressed as SID list C, F, Z



```

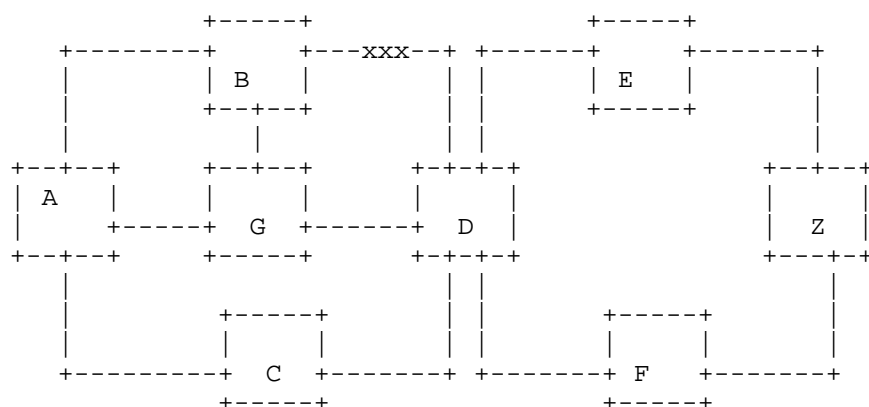
SR Policy A-Z:
  Candidate path1
    SIDList1 [B,E,Z]
  Candidate path2
    SIDList2 [C,F,Z]

```

Figure 1: SR policy with 2 diverse candidate paths

3.1. Deviation due to failures

In Figure 2, link B-D fails. The expected circuit-style behavior is to start using the second candidate path. Though this path may be used initially, once the IGP converges, the candidate path 1 becomes valid as node B regains a shortest path to the next node SID E. Once the headend switches to the candidate path 1, the intended path and the expansion of the SID list which now becomes (A-B, B-G, G-D, D-E, E-Z) deviate. The service starts to use resources on B-G and G-D links where the PCE has not made a bandwidth reservation.



SR Policy A-Z:

Candidate path1

SIDList1 [B,E,Z] --> deviation from intended path due to failure

Candidate path2

SIDList2 [C,F,Z]

Figure 2: SR policy CP1 deviation after link failure and IGP convergence

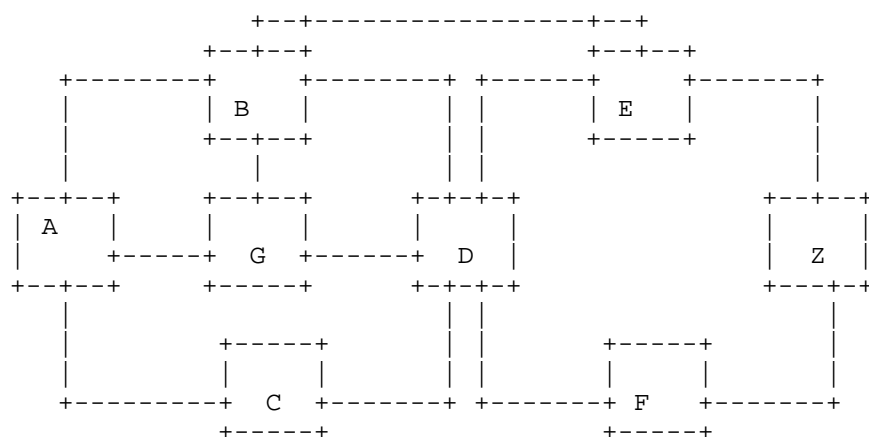
A possible solution to this is for PCE to monitor these deviations and correct the compressed SID lists. However, the PCE is not as real-time as the IGP (e.g. many BGP-LS implementations use periodic injection of IGP events into BGP) and PCE is burdened by many more services going over this link not just by the services originating at node A. As a result, relying on PCE to correct this behavior is not desired.

This document uses the proposed extension to active path selection algorithm introduced in draft

[I-D.karboubi-spring-sr-policy-eligibility] which allows a system to render the candidate path 1 ineligible for selection at the head-end node. Making a path eligible again would be the responsibility of the PCE. This is elaborated in Section 4.

3.2. Deviation due to repairs

Figure 3 shows an example where a link B-E that was down at the time PCE computed the above two candidate paths is now repaired. When the link B-E repairs, the compressed SID list expands now to (A-B, B-E, E-Z) which is a deviation from the intended path. Though this path looks attractive, it may not have the bandwidth the service needs.



SR Policy A-Z:

Candidate path1

SIDList1 [B,E,Z] --> deviation from intended path due to repair

Candidate path2

SIDList2 [C,F,Z]

Figure 3: SR policy CP1 deviation after link repair and IGP convergence

This document presents a SID compression algorithm that is resilient to such repairs. This is elaborated in Section 5.

4. Dealing with deviation due to failures

We use the eligibility concept introduced in [I-D.karboubi-spring-sr-policy-eligibility]. the head-end node is responsible for detecting failures , setting the eligibility to false and switching to the next candidate path within 50 milliseconds.

The eligibility of a path is also controlled by the PCE. The PCE may set it to true or false depending on whether the expanded SID list matches the intended path. When the link B-D in Figure 2 repairs, it is the responsibility of the PCE to set the eligibility of the candidate path 1 to true. This allows eligibility mechanism to work across IGP areas and BGP autonomous systems.

We introduce a second attribute that controls this new automated behavior of setting eligibility. An operator who plans to implement circuit style policies would enable this behavior, see Section 4.3

4.1. Head end node procedure

The head-end node shall run a connectivity verification protocol as specified in section 7.1 of [I-D.ietf-spring-cs-sr-policy] to determine path failure. When the head end detects a failure of a candidate path, the eligibility attribute is set immediately to false. As per [I-D.karboubi-spring-sr-policy-eligibility], Head end node will no longer consider this candidate path in its active path selection logic no matter what other link/node failures and repairs and IP convergence may happen in the network. If another candidate path exists, the head end will switch to the next eligible candidate path per the active candidate path selection algorithm. The recovery scheme for such policies is same as described in CS-SR draft , where such policies can be unprotected, use 1:N protection or protection combined with restoration.

Note that this implies that head end node needs to detect end-to-end failures before any local repair (TI-LFA) or IP convergence occurs. There are various implementation ways to achieve this:

- * Configure the CCV protocol (e.g. S-BFD or STAMP) for these SR Policies at a lower interval than the IP link BFD. This will not impact non-CS SR policies which will continue to benefit from TI-LFA local repairs with same detection/repair time as before. Note that CCV is mandatory for CS SR policies, so the only new addition imposed here is regarding its detection timer (i.e. inverted hierarchical fault detection where e2e fault is detected before 1-hop fault).
- * Another implementation solution to circumvent the TI-LFA, is to disable TI-LFA for CS-SR based traffic. This can be achieved by using only flexAlgo Node SIDs that have TILFA disabled, so when computing SID List for a CS-SR Policies, only Nodes SIDs from flexAlgo with disabled TI LFA would be used. This will not require separate loopback for nodes, but simply defining a flexAlgo with TI-LFA disabled on all Nodes pertaining to CS-SR domain. So, in the case a link fails, (before the e2e failure could be detected) the PLR will perform the usual TI LFA post convergence path for standard SID and will not initiate TI-LFA for traffic destined to CS-SR SIDs. With this approach we only need to ensure that e2e detection is lower than IGP convergence time only.

4.2. Controller/PCE component procedure

The PCE also maintains an accurate view of the network topology in all IGP areas and BGP autonomous systems in the network. After the failures have been repaired, the candidate paths that have been set as not eligible by head-end nodes may now be eligible again. In this case, PCE will set the eligibility attribute of these candidate paths to true.

It is up to the SR policy head-end node to reselect the active candidate path after PCE changes eligibility of the candidate paths. The head end may either implement a standard revertive behavior whereby it can revert immediately once a better preferred path becomes available for selection, or wait for a period of time or implement a non-revertive behavior whereby traffic is not switched back automatically until there is a failure on the currently active candidate path. This reversion behavior may be controlled by a SP provider policy and is outside the scope of this document.

A PCE may also set a candidate path as ineligible if it detects that the SID list when expanded is different from the intended path. This step is not mandatory when head-end is able to monitor all candidate paths for failures. But, this step is necessary for implementations that do not monitor the inactive candidate paths. This is an implementation detail. We allow PCE to set eligibility attribute to true or false. The node is only allowed to set it to false for our use case.

4.3. Eligibility control attribute

The second configuration is at the SR policy level and is used by head end node to determine whether the behavior described in Section 4.1 is desirable or not (notably the head end disabling the eligibility of a path). This attribute is called `eligibilityControl` and when set to false (default) the head end node will not alter the eligibility setting of a CP.

5. Dealing with deviation due to repairs/changes

Network improvements and node and/or link repairs can also result in segment list expansions and intended paths to deviate. Network improvement may include addition of brand new links or changes of link attributes such as metric, SRLG values, affinity values, etc.

Most of these changes, with the exception of restoration of down links, are typically done in maintenance-windows and under the supervision of an SDN controller. By performing these operations under the supervision of a controller, operator can work around their

impacts on paths before making them. Such coordination would be necessary for existing MPLS-TP based solutions or CS-SR solution, as changes to these properties e.g. affinity or delay may cause an intent violation with original path, which needs to be reassessed. Such controller role providing automation and coordination between different layers and workflows is not uncommon and is beneficial for self-optimizing networks. Hence these kinds of changes are not the focus of this solution. Our focus is on node and/or link repairs (not necessarily limited to the links used by the candidate paths)

For repairs, we propose a segment compaction algorithm whose compaction is resilient to nodes and/or links repairs; that is the segment list expands to the same path before or after any of these down links in any combination repairs. Any algorithm that is resilient to repairs would work. We highlight one such algorithm in the next paragraph.

While the PCE computes the intended path on current state of the network, the proposed segment compaction algorithm uses a network view where all down links are restored to produce the SID list for the intended path. This compaction may not be as short as the compaction with the restored links as down but has the property that it is resilient to repairs. That is, the SID list will always expand to the intended path. This property is independent of the order at which the links are repaired.

Note that in absence of such algorithm (SID List being resilient to repairs), the paths could still be corrected by controller where upon link repair it would assess CS-SR policies and check if the newly repair link have caused any deviation from their intended paths and when such deviation is detected, a new SID List, that is expressing the intended path, is updated on head end. The drawback though is that deviation will be momentarily observed and traffic may be going on the repaired link until controller corrects the SID List.

6. Protocol and model changes

6.1. Active candidate path selection

This is already covered in
[I-D.karboubi-spring-sr-policy-eligibility].

6.2. PCEP extensions

The extensions defined in
[I-D.sidor-pce-circuit-style-pcep-extensions] regarding the strict path enforcement (using strict adjacency SIDs) becomes optional.

PCEP shall be extended to signal the new eligibility control and eligibility attributes defined above.

6.3. SR policy Yang changes

The eligibility control and eligibility attributes shall be added for the SR policy and candidate path YANG models respectively.

NetConf RPC calls can be used to set eligibility of candidate paths to true or false.

7. IANA considerations

This document includes no request to IANA.

8. Security considerations

TO BE ADDED

9. References

9.1. Normative References

- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/rfc/rfc8402>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

9.2. Informative References

- [I-D.ietf-spring-cs-sr-policy] Schmutzer, C., Ali, Z., Maheshwari, P., Rokui, R., and A. Stone, "Circuit Style Segment Routing Policies, Work in Progress, Internet-Draft, draft-ietf-spring-cs-sr-policy-01", 23 October 2023, <<https://datatracker.ietf.org/doc/draft-ietf-spring-cs-sr-policy>>.

[I-D.sidor-pce-circuit-style-pcep-extensions]

Sidor, S., Ali, Z., Maheshwari, P., Rokui, R., Stone, A., Jalil, L., Peng, S., Saad, T., and D. Voyer, "PCEP extensions for Circuit Style Policies", Work in Progress, Internet-Draft, draft-sidor-pce-circuit-style-pcep-extensions-06", 15 December 2023, <<https://datatracker.ietf.org/doc/html/draft-sidor-pce-circuit-style-pcep-extensions-06>>.

[I-D.karboubi-spring-sr-policy-eligibility]

Karboubi, A., Shah, H., Sivalaban, S., Stone, A., and C. Schmutz, "Eligibility Concept in Segment Routing Policies, Work in Progress, Internet-Draft, draft-karboubi-spring-sr-policy-eligibility", 21 February 2025, <<https://datatracker.ietf.org/doc/draft-karboubi-spring-sr-policy-eligibility>>.

[RFC9256] Filsfils, C., Talaulikar, K., Ed., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", RFC 9256, DOI 10.17487/RFC9256, July 2022, <<https://www.rfc-editor.org/rfc/rfc9256>>.

Acknowledgements

The authors would like to thank Abhijeet Bidawe for his input and support.

Authors' Addresses

Amal Karboubi (editor)
Ciena
Email: akarboub@ciena.com

Cengiz Alaettinoglu
Ciena
Email: cengiz@ciena.com

Himanshu Shah (editor)
Ciena
Email: hshah@ciena.com

Siva Sivabalan
Ciena
Email: ssivabal@ciena.com

Todd Defillipi
Ciena
Email: todd@ciena.com

Praveen Maheshwari
Airtel India
Email: Praveen.Maheshwari@airtel.com