

Internet Engineering Task Force (IETF)
Request for Comments: 8355
Category: Informational
ISSN: 2070-1721

C. Filsfils, Ed.
S. Previdi, Ed.
Cisco Systems, Inc.
B. Decraene
Orange
R. Shakir
Google
March 2018

Resiliency Use Cases
in Source Packet Routing in Networking (SPRING) Networks

Abstract

This document identifies and describes the requirements for a set of use cases related to Segment Routing network resiliency on Source Packet Routing in Networking (SPRING) networks.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc8355>.

Copyright Notice

Copyright (c) 2018 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Language	4
2. Path Protection	4
3. Management-Free Local Protection	6
3.1. Management-Free Bypass Protection	7
3.2. Management-Free Shortest-Path-Based Protection	8
4. Managed Local Protection	8
4.1. Managed Bypass Protection	9
4.2. Managed Shortest Path Protection	9
5. Loop Avoidance	10
6. Coexistence of Multiple Resilience Techniques in the Same Infrastructure	10
7. Security Considerations	11
8. IANA Considerations	11
9. Manageability Considerations	11
10. References	12
10.1. Normative References	12
10.2. Informative References	12
Acknowledgements	12
Contributors	12
Authors' Addresses	13

1. Introduction

This document reviews various use cases for the protection of services in a SPRING network. The terminology used hereafter is in line with [RFC5286] and [RFC5714].

The resiliency use cases described in this document can be applied not only to traffic that is forwarded according to the SPRING architecture, but also to traffic that originally is forwarded using other paradigms such as LDP signaling or pure IP traffic (IP-routed traffic).

Three key alternatives are described: path protection, local protection without operator management, and local protection with operator management.

Path protection lets the ingress node be in charge of the failure recovery, as discussed in Section 2.

The rest of the document focuses on approaches where protection is performed by the node adjacent to the failed component, commonly referred to as local protection techniques or fast-reroute techniques [RFC5286] [RFC5714].

In Section 3, we discuss two different approaches providing unmanaged local protection, namely link/node bypass protection and shortest-path-based protection.

Section 4 illustrates a case allowing the operator to manage the local protection behavior in order to accommodate specific policies.

In Section 5, we discuss the opportunity for the SPRING architecture to provide loop-avoidance mechanisms such that transient forwarding state inconsistencies during routing convergence do not lead into traffic loss.

The purpose of this document is to illustrate the different use cases and explain how an operator could combine them in the same network (see Section 6). Solutions are not defined in this document.

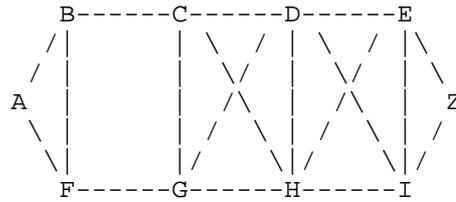


Figure 1: Reference Topology

We use Figure 1 as a reference topology throughout the document. The following link metrics are applied:

- o Links from/to A and Z are configured with a metric of 100.
- o CH, GD, DI, and HE links are configured with a metric of 6.
- o All other links are configured with a metric of 5.

Note: Link metrics are bidirectional; in other words, the same metric value is configured at both sides of each link.

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. Path Protection

As a reminder, one of the major network operator requirements is path disjointness capability. Network operators have deployed infrastructures with topologies that allow paths to be computed in a complete disjoint fashion where two paths wouldn't share any component (link or router), hence allowing an optimal protection strategy.

A first protection strategy consists of excluding any local repair and instead uses end-to-end path protection where each SPRING path is protected by a second disjoint SPRING path. In this case, local protection is not used along the path.

For example, a pseudowire (PW) from A to Z can be "path protected" in the direction A to Z in the following manner: the operator configures two SPRING paths, T1 (primary) and T2 (backup), from A to Z.

The two paths may be used:

- o concurrently, where the ingress router sends the same traffic over the primary and secondary path (this is usually known as 1+1 protection);
- o concurrently, where the ingress router splits the traffic over the primary and secondary path (this is usually known as Equal-Cost Multipath (ECMP) or Unequal-Cost Multipath (UCMP)); or
- o as a primary and backup path, where the secondary path is used only when the primary failed (this is usually known as 1:1 protection).

T1 is established over path {AB, BC, CD, DE, EZ} as the primary path, and T2 is established over path {AF, FG, GH, HI, IZ} as the backup path. The two paths MUST be disjoint in their links, nodes, and Shared Risk Link Groups (SRLGs) to satisfy the requirement of disjointness.

In the case of primary/backup paths, when the primary path T1 is up, the packets of the PW are sent on T1. When T1 fails, the packets of the PW are sent on the backup path T2. When T1 comes back up, the operator either allows for an automated reversion of the traffic onto T1 or selects an operator-driven reversion. Typically, the switchover from path T1 to path T2 is done in a fast-reroute fashion (e.g., sub-50 milliseconds) but, depending on the service that needs to be delivered, other restoration times may be used.

It is essential that any path, primary or backup, benefit from an end-to-end liveness monitoring/verification. The method and mechanisms that provide such a liveness check are outside the scope of this document. An example is given by [RFC5880].

There are multiple options for a liveness check, e.g., path liveness, where the path is monitored at the network level (either by the head-end node or by a network controller/monitoring system). Another possible approach consists of a service-based path monitored by the service instance (verifying reachability of the endpoint). All these options are given here as examples. While this document does express the requirement for a liveness mechanism, it does not mandate, nor define, any specific one.

From a SPRING viewpoint, we would like to highlight the following requirements:

- o SPRING architecture MUST provide a way to compute paths that are not protected by local repair techniques (as illustrated in the example of paths T1 and T2).
- o SPRING architecture MUST provide a way to instantiate pairs of disjoint paths on a topology based on a protection strategy (link, node, or SRLG protection) and allow the validation or recomputation of these paths upon network events.
- o The SPRING architecture MUST provide an end-to-end liveness check of SPRING-based paths.

3. Management-Free Local Protection

This section describes two alternatives that provide local protection without requiring operator management, namely bypass protection and shortest-path-based protection.

For example, traffic from A to Z, transported over the shortest paths provided by the SPRING architecture, benefits from management-free local protection by having each node along the path automatically precompute and preinstall a backup path for the destination Z. Upon local detection of the failure, the traffic is repaired over the backup path in sub-50 milliseconds. When the primary path comes back up, the operator either allows for an automated reversion of the traffic onto it or selects an operator-driven reversion.

The backup path computation SHOULD support the following requirements:

- o 100% link, node, and SRLG protection in any topology;
- o automated computation by the IGP; and
- o selection of the backup path such as to minimize the chance for transient congestion and/or delay during the protection period, as reflected by the IGP metric configuration in the network.

3.1. Management-Free Bypass Protection

One way to provide local repair is to enforce a failover along the shortest path around the failed component.

In case of link protection, the point of local repair will create a repair path avoiding the protected link and merging back to the primary path at the next hop.

In case of node protection, the repair path will avoid the protected node and merge back to the primary path at the next-next hop.

In case of SRLG protection, the repair path will avoid members of the same group and merge back to the primary path just after.

In our example, C protects destination Z against a failure of the CD link by enforcing the traffic over the bypass {CH, HD}. The resulting end-to-end path between A and Z, upon recovery from the failure of CD, is depicted in Figure 2.

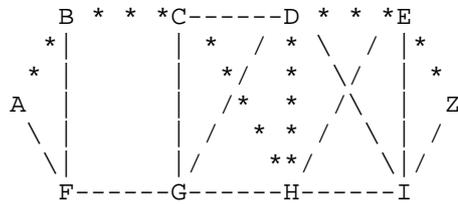


Figure 2: Bypass Protection around Link CD

When the primary path comes back up, the operator either allows for an automated reversion of the traffic onto the primary path or selects an operator-driven reversion.

3.2. Management-Free Shortest-Path-Based Protection

An alternative protection strategy consists in management-free local protection that is aimed at providing a repair for the destination based on the shortest path to the destination.

In our example, C protects Z (which the traffic initially reaches via CD) by enforcing the traffic over its shortest path to Z and considering the failure of the protected component. The resulting end-to-end path between A and Z, upon recovery from the failure of CD, is depicted in Figure 3.

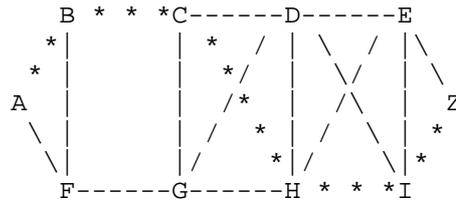


Figure 3: Shortest Path Protection around Link CD

When the primary path comes back up, the operator either allows for an automated reversion of the traffic onto the primary path or selects an operator-driven reversion.

4. Managed Local Protection

There may be cases where a management-free repair does not fit the policy of the operator. For example, in our illustration, the operator may not want to have CD and CH used to protect each other due to the bandwidth (BW) availability in each link that could not suffice to absorb the other link traffic.

In this context, the protection mechanism MUST support the explicit configuration of the backup path either under the form of high-level constraints (end at the next hop, end at the next-next hop, minimize this metric, avoid this SRLG, etc.) or under the form of an explicit path. Upon local detection of the failure, the traffic is repaired over the backup path in sub-50 milliseconds. When the primary path comes back up, the operator either allows for an automated reversion of the traffic onto it or selects an operator-driven reversion.

We discuss such aspects for both bypass and shortest-path-based protection schemes.

4.1. Managed Bypass Protection

Let us illustrate the case using our reference example. For the demand from A to Z, the operator does not want to use the shortest failover path to the next hop, {CH, HD}, but rather the path {CG, GH, HD}, as illustrated in Figure 4.

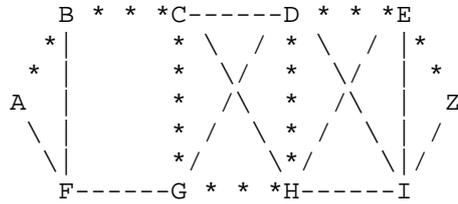


Figure 4: Managed Bypass Protection

The computation of the repair path SHOULD be possible in an automated fashion as well as statically expressed in the point of local repair.

4.2. Managed Shortest Path Protection

In the case of shortest path protection, the operator does not want to use the shortest failover via link CH, but rather the traffic should reach H via {CG, GH} due to constraints such as delay, BW, or SRLG.

The resulting end-to-end path upon activation of the protection is illustrated in Figure 5.

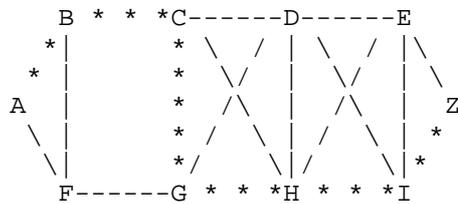


Figure 5: Managed Shortest Path Protection

The computation of the repair path SHOULD be possible in an automated fashion as well as statically expressed in the point of local repair.

The computation of the repair path based on a specific constraint SHOULD be possible on a per-destination prefix base.

5. Loop Avoidance

It is part of routing protocols' behavior to have what are called "transient routing inconsistencies". This is due to the routing convergence that happens in each node at different times and during a different lapse of time.

These inconsistencies may cause routing loops that last the time that it takes for the node impacted by a network event to converge. These loops are called "micro-loops".

Usually, in normal routing protocol operations, micro-loops do not last long and are only noticed during the time it takes the network to converge. However, with the emergence of fast-convergence and fast-reroute technologies, micro-loops can be an issue in networks where sub-50 millisecond convergence/reroute is required. Therefore, the micro-loop problem needs to be addressed.

Networks may be affected by micro-loops during convergence depending of their topologies. Detecting micro-loops can be done during topology computation (e.g., Shortest Path First (SPF) computation), and therefore techniques to avoid micro-loops may be applied. An example of such technique is to compute a path free of micro-loops that would be used during network convergence.

The SPRING architecture SHOULD provide solutions to prevent the occurrence of micro-loops during convergence following a change in the network state. Traditionally, the lack of packet steering capability made it difficult to apply efficient solutions to micro-loops. A SPRING-enabled router could take advantage of the increased packet steering capabilities offered by SPRING in order to steer packets in a way that packets do not enter such loops.

6. Coexistence of Multiple Resilience Techniques in the Same Infrastructure

The operator may want to support several very different services on the same packet-switching infrastructure. As a result, the SPRING architecture SHOULD allow for the coexistence of the different use cases listed in this document, in the same network.

Let us illustrate this with the following example:

- o Flow F1 is supported over path {C, CD, E}
- o Flow F2 is supported over path {C, CD, I}
- o Flow F3 is supported over path {C, CD, Z}
- o Flow F4 is supported over path {C, CD, Z}

It should be possible for the operator to configure the network to achieve path protection for F1, management-free shortest path local protection for F2, managed protection over path {CG, GH, Z} for F3, and management-free bypass protection for F4.

7. Security Considerations

This document describes requirements for the SPRING architecture to provide resiliency in SPRING networks. As such, it does not introduce any new security considerations beyond those discussed in [RFC7855].

8. IANA Considerations

This document has no IANA actions.

9. Manageability Considerations

This document provides use cases. Solutions aimed at supporting these use cases should provide the necessary mechanisms in order to allow for manageability as described in [RFC7855].

Manageability concerns the computation, installation, and troubleshooting of the repair path. Also, necessary mechanisms SHOULD be provided in order for the operator to control when a repair path is computed, how it has been computed, and if it's installed and used.

10. References

10.1. Normative References

- [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/info/rfc2119>>.
- [RFC7855] Previdi, S., Ed., Filsfils, C., Ed., Decraene, B., Litkowski, S., Horneffer, M., and R. Shakir, "Source Packet Routing in Networking (SPRING) Problem Statement and Requirements", RFC 7855, DOI 10.17487/RFC7855, May 2016, <<https://www.rfc-editor.org/info/rfc7855>>.
- [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/info/rfc8174>>.

10.2. Informative References

- [RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <<https://www.rfc-editor.org/info/rfc5286>>.
- [RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714, January 2010, <<https://www.rfc-editor.org/info/rfc5714>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.

Acknowledgements

The authors would like to thank Stephane Litkowski and Alexander Vainshtein for the comments and review of this document.

Contributors

Pierre Francois contributed to the writing of the first draft version of this document.

Authors' Addresses

Clarence Filsfils (editor)
Cisco Systems, Inc.
Brussels
Belgium

Email: cfilsfil@cisco.com

Stefano Previdi (editor)
Cisco Systems, Inc.
Via Del Serafico, 200
Rome 00142
Italy

Email: stefano@previdi.net

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Rob Shakir
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
United States of America

Email: robjs@google.com

