Routing Area Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 23, 2017

S. Litkowski
B. Decraene
Orange
C. Filsfils
Cisco Systems
P. Francois
Individual

June 21, 2017

Micro-loop prevention by introducing a local convergence delay
draft-ietf-rtgwg-uloop-delay-05

Abstract

This document describes a mechanism for link-state routing protocols to prevent local transient forwarding loops in case of link failure. This mechanism proposes a two-steps convergence by introducing a delay between the convergence of the node adjacent to the topology change and the network wide convergence.

As this mechanism delays the IGP convergence it may only be used for planned maintenance or when fast reroute protects the traffic between the link failure time and the IGP convergence.

The proposed mechanism will be limited to the link down event in order to keep simplicity.

Simulations using real network topologies have been performed and show that local loops are a significant portion (>50%) of the total forwarding loops.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

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 http://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction

Micro-forwarding loops and some potential solutions are well described in [RFC5715]. This document describes a simple targeted mechanism that solves micro-loops that are local to the failure; based on network analysis, these are a significant portion of the micro-forwarding loops. A simple and easily deployable solution for these local micro-loops is critical because these local loops cause some traffic loss after a fast-reroute alternate has been used (see Section 2.1).

Consider the case in Figure 1 where S does not have an LFA to protect its traffic to D. That means that all non-D neighbors of S on the topology will send to S any traffic destined to D if a neighbor did not, then that neighbor would be loop-free. Regardless of the advanced fast-reroute (FRR) technique used, when S converges to the new topology, it will send its traffic to a neighbor that was not loop-free and thus cause a local micro-loop. The deployment of advanced fast-reroute techniques motivates this simple router-local mechanism to solve this targeted problem. This solution can be work with the various techniques described in [RFC5715].

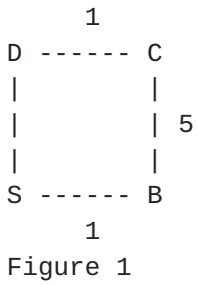


Figure 1

When S-D fails, a transient forwarding loop may appear between S and B if S updates its forwarding entry to D before B.

2. Transient forwarding loops side effects

Even if they are very limited in duration, transient forwarding loops may cause high damages for a network.

2.1. Fast reroute inefficiency

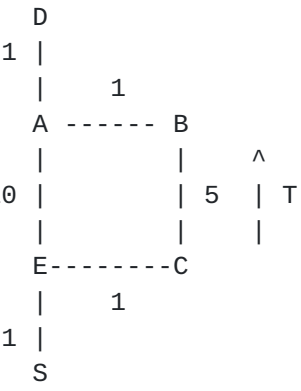


Figure 2 - RSVP-TE FRR case

In the Figure 2, an RSVP-TE tunnel T, provisioned on C and terminating on B, is used to protect against C-B link failure (IGP shortcut is activated on C). The primary path of T is C->B and FRR is activated on T providing an FRR bypass or detour using path C->E->A->B. On the router C, the nexthop to D is the tunnel T thanks to the IGP shortcut. When C-B link fails:

1. C detects the failure, and updates the tunnel path using preprogrammed FRR path, the traffic path from S to D becomes: S->E->C->E->A->B->A->D.

2. In parallel, on router C, both the IGP convergence and the TE tunnel convergence (tunnel path recomputation) are occurring:

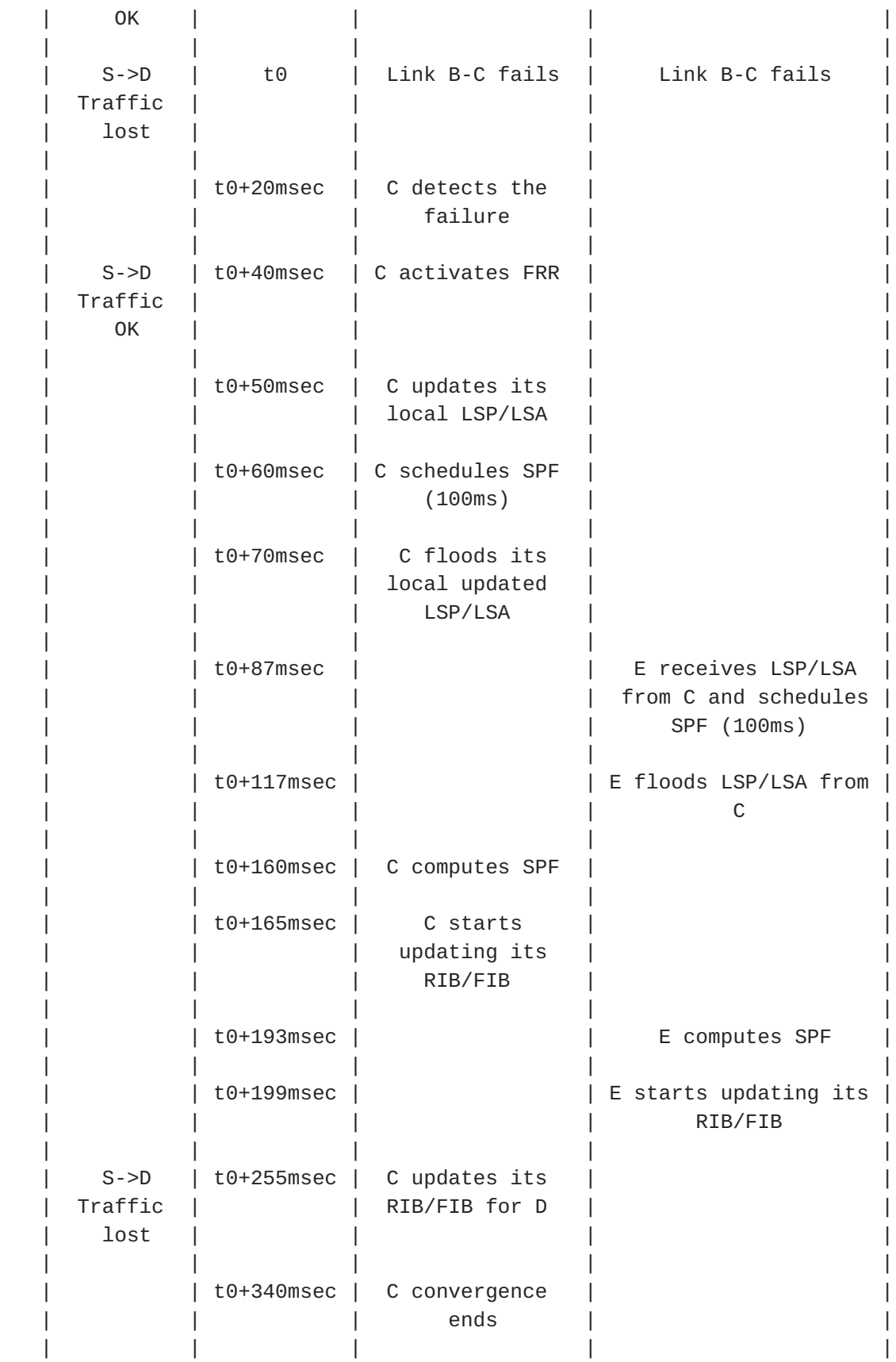
   * The Tunnel T path is recomputed and now uses C->E->A->B.

   * The IGP path to D is recomputed and now uses C->E->A->D.

3. On C, the tail-end of the TE tunnel (router B) is no more on the shortest-path tree (SPT) to D, so C does not encapsulate anymore the traffic to D using the tunnel T and updates its forwarding entry to D using the nexthop E.

If C updates its forwarding entry to D before router E, there would be a transient forwarding loop between C and E until E has converged.

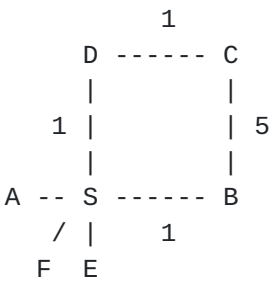
Network condition	Time	Router C events	Router E events
S->D Traffic			


OK
S->D Traffic lost
t0
Link B-C fails
Link B-C fails
t0+20msec
C detects the failure
t0+40msec
C activates FRR
S->D Traffic OK
t0+50msec
C updates its local LSP/LSA
t0+60msec
C schedules SPF (100ms)
t0+70msec
C floods its local updated LSP/LSA
t0+87msec
E receives LSP/LSA from C and schedules SPF (100ms)
t0+117msec
E floods LSP/LSA from C
t0+160msec
C computes SPF
t0+165msec
C starts updating its RIB/FIB
t0+193msec
E computes SPF
t0+199msec
E starts updating its RIB/FIB
S->D Traffic lost
t0+255msec
C updates its RIB/FIB for D
t0+340msec
C convergence ends

Route computation event time scale

The issue described here is completely independent of the fast-reroute mechanism involved (TE FRR, LFA/rLFA, MRT ...). The protection enabled by fast-reroute is working perfectly, but ensures a protection, by definition, only until the PLR has converged. When implementing FRR, a service provider wants to guarantee a very limited loss of connectivity time. The previous example shows that the benefit of FRR may be completely lost due to a transient forwarding loop appearing when PLR has converged. Delaying FIB updates after the IGP convergence may allow to keep the fast-reroute path until the neighbors have converged and preserves the customer traffic.

2.2. Network congestion



In the figure above, as presented in Section 1, when the link S-D fails, a transient forwarding loop may appear between S and B for destination D. The traffic on the S-B link will constantly increase due to the looping traffic to D. Depending on the TTL of the packets, the traffic rate destinated to D and the bandwidth of the link, the S-B link may be congested in few hundreds of milliseconds and will stay overloaded until the loop is solved.

The congestion introduced by transient forwarding loops is problematic as it is impacting traffic that is not directly concerned by the failing network component. In our example, the congestion of the S-B link will impact some customer traffic that is not directly concerned by the failure: e.g. A to B, F to B, E to B. Some class of services may be implemented to mitigate the congestion, but some traffic not directly concerned by the failure would still be dropped

as a router is not able to identify the looping traffic from the normally forwarded traffic.

3. Overview of the solution

This document defines a two-step convergence initiated by the router detecting the failure and advertising the topological changes in the IGP. This introduces a delay between the convergence of the local router and the network wide convergence.

The proposed solution is kept limited to local link down events for simplicity reason.

This ordered convergence, is similar to the ordered FIB proposed defined in [RFC6976], but limited to only a "one hop" distance. As a consequence, it is simpler and becomes a local only feature not requiring interoperability; at the cost of only covering the transient forwarding loops involving this local router. The proposed mechanism also reuses some concept described in [I-D.ietf-rtgwg-microloop-analysis] with some limitations.

4. Specification

4.1. Definitions

This document will refer to the following existing IGP timers:

- LSP_GEN_TIMER: The delay used to batch multiple local events in one single local LSP/LSA update. It is often associated with a damping mechanism to slow down reactions by incrementing the timer when multiple consecutive events are detected.

- SPF_DELAY: The delay between the first IGP event triggering a new routing table computation and the start of that routing table computation. It is often associated with a damping mechanism to slow down reactions by incrementing the timer when the IGP becomes unstable. As an example, [I-D.ietf-rtgwg-backoff-algo] defines a standard SPF delay algorithm.

This document introduces the following new timer:

- ULOOP_DELAY_DOWN_TIMER: used to slow down the local node convergence in case of link down events.

4.2. Current IGP reactions

Upon a change of the status of an adjacency/link, the existing behavior of the router advertising the event is the following:

1. The Up/Down event is notified to the IGP.

2. The IGP processes the notification and postpones the reaction in LSP_GEN_TIMER msec.

3. Upon LSP_GEN_TIMER expiration, the IGP updates its LSP/LSA and floods it.

4. The SPF computation is scheduled in SPF_DELAY msec.

5. Upon SPF_DELAY expiration, the SPF is computed, then the RIB and FIB are updated.

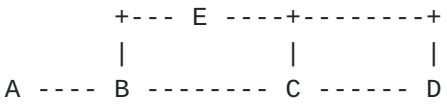
4.3. Local events

The mechanism described in this document assumes that there has been a single link failure as seen by the IGP area/level. If this assumption is violated (e.g. multiple links or nodes failed), then standard IP convergence MUST be applied (as described in Section 4.2).

To determine if the mechanism can be applicable or not, an implementation SHOULD implement a logic to correlate the protocol messages (LSP/LSA) received during the SPF scheduling period in order to determine the topology changes that occured. This is necessary as multiple protocol messages may describe the same topology change and a single protocol message may describe multiple topology changes. As a consequence, determining a particular topology change MUST be independent of the order of reception of those protocol messages. How the logic works is let to implementation details.

Using this logic, if an implementation determines that the associated topology change is a single local link failure, then the router MAY use the mechanism described in this document, otherwise the standard IP convergence MUST be used.

Example:



Let router B be the computing router when the link B-C fails. B updates its local LSP/LSA describing the link B->C as down, C does the same, and both start flooding their updated LSP/LSAs. During the SPF_DELAY period, B and C learn all the LSPs/LSAs to consider. B sees that C is flooding as down a link where B is the other end and that B and C are describing the same single event. Since B receives no other changes, B can determine that this is a local link failure and may decide to activate the mechanism described in this document.

4.4. Local delay for link down

Upon an adjacency/link down event, this document introduces a change in step 5 (Section 4.2) in order to delay the local convergence compared to the network wide convergence: the node SHOULD delay the forwarding entry updates by ULOOP_DELAY_DOWN_TIMER. Such delay SHOULD only be introduced if all the LSDB modifications processed are only reporting a single local link down event (Section 4.3). If a subsequent LSP/LSA is received/updated and a new SPF computation is triggered before the expiration of ULOOP_DELAY_DOWN_TIMER, then the same evaluation SHOULD be performed.

As a result of this addition, routers local to the failure will converge slower than remote routers. Hence it SHOULD only be done for a non-urgent convergence, such as for administrative de-activation (maintenance) or when the traffic is protected by fast-reroute.

5. Applicability

As previously stated, the mechanism only avoids the forwarding loops on the links between the node local to the failure and its neighbor. Forwarding loops may still occur on other links.

5.1. Applicable case: local loops

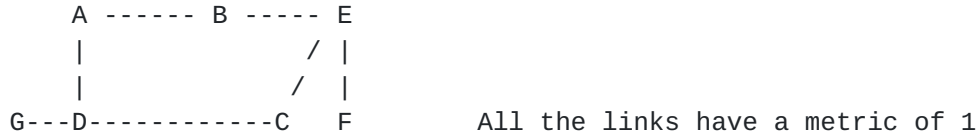


Figure 2

Let us consider the traffic from G to F. The primary path is G->D->C->E->F. When link C-E fails, if C updates its forwarding entry for F before D, a transient loop occurs. This is sub-optimal as C has FRR enabled and it breaks the FRR forwarding while all upstream routers are still forwarding the traffic to itself.

By implementing the mechanism defined in this document on C, when the C-E link fails, C delays the update of its forwarding entry to F, in order to let some time for D to converge. FRR keeps protecting the traffic during this period. When the timer expires on C, its forwarding entry to F is updated. There is no transient forwarding loop on the link C-D.

5.2. Non applicable case: remote loops

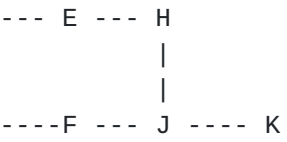


Figure 3

Let us consider the traffic from G to K. The primary path is G->D->C->F->J->K. When the C-F link fails, if C updates its forwarding entry to K before D, a transient loop occurs between C and D.

By implementing the mechanism defined in this document on C, when the link C-F fails, C delays the update of its forwarding entry to K, letting time for D to converge. When the timer expires on C, its forwarding entry to F is updated. There is no transient forwarding loop between C and D. However, a transient forwarding loop may still occur between D and A. In this scenario, this mechanism is not enough to address all the possible forwarding loops. However, it does not create additional traffic loss. Besides, in some cases -such as when the nodes update their FIB in the following order C, A, D, for example because the router A is quicker than D to converge- the mechanism may still avoid the forwarding loop that was occurring.

6. Simulations

Simulations have been run on multiple service provider topologies.

Topology	Gain
T1	71%
T2	81%
T3	62%
T4	50%
T5	70%
T6	70%
T7	59%
T8	77%

Table 1: Number of Repair/Dst that may loop

We evaluated the efficiency of the mechanism on eight different service provider topologies (different network size, design). The benefit is displayed in the table above. The benefit is evaluated as follows:

- We consider a tuple (link A-B, destination D, PLR S, backup nexthop N) as a loop if upon link A-B failure, the flow from a router S upstream from A (A could be considered as PLR also) to D may loop due to convergence time difference between S and one of his neighbor N.

- We evaluate the number of potential loop tuples in normal conditions.

- We evaluate the number of potential loop tuples using the same topological input but taking into account that S converges after N.

- The gain is how much loops (remote and local) we succeed to suppress.

On topology 1, 71% of the transient forwarding loops created by the failure of any link are prevented by implementing the local delay. The analysis shows that all local loops are obviously solved and only remote loops are remaining.

7. Deployment considerations

Transient forwarding loops have the following drawbacks:

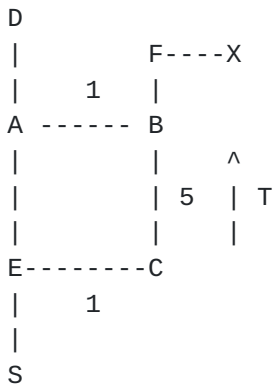
- They limit FRR efficiency: even if FRR is activated in 50msec, as soon as PLR has converged, the traffic may be affected by a transient loop.

o They may impact traffic not directly concerned by the failure (due to link congestion).

This local delay proposal is a transient forwarding loop avoidance mechanism (like OFIB). Even if it only addresses local transient loops, the efficiency versus complexity comparison of the mechanism makes it a good solution. It is also incrementally deployable with incremental benefits, which makes it an attractive option for both vendors to implement and Service Providers to deploy. Delaying the convergence time is not an issue if we consider that the traffic is protected during the convergence.

8. Examples

We will consider the following figure for the associated examples :

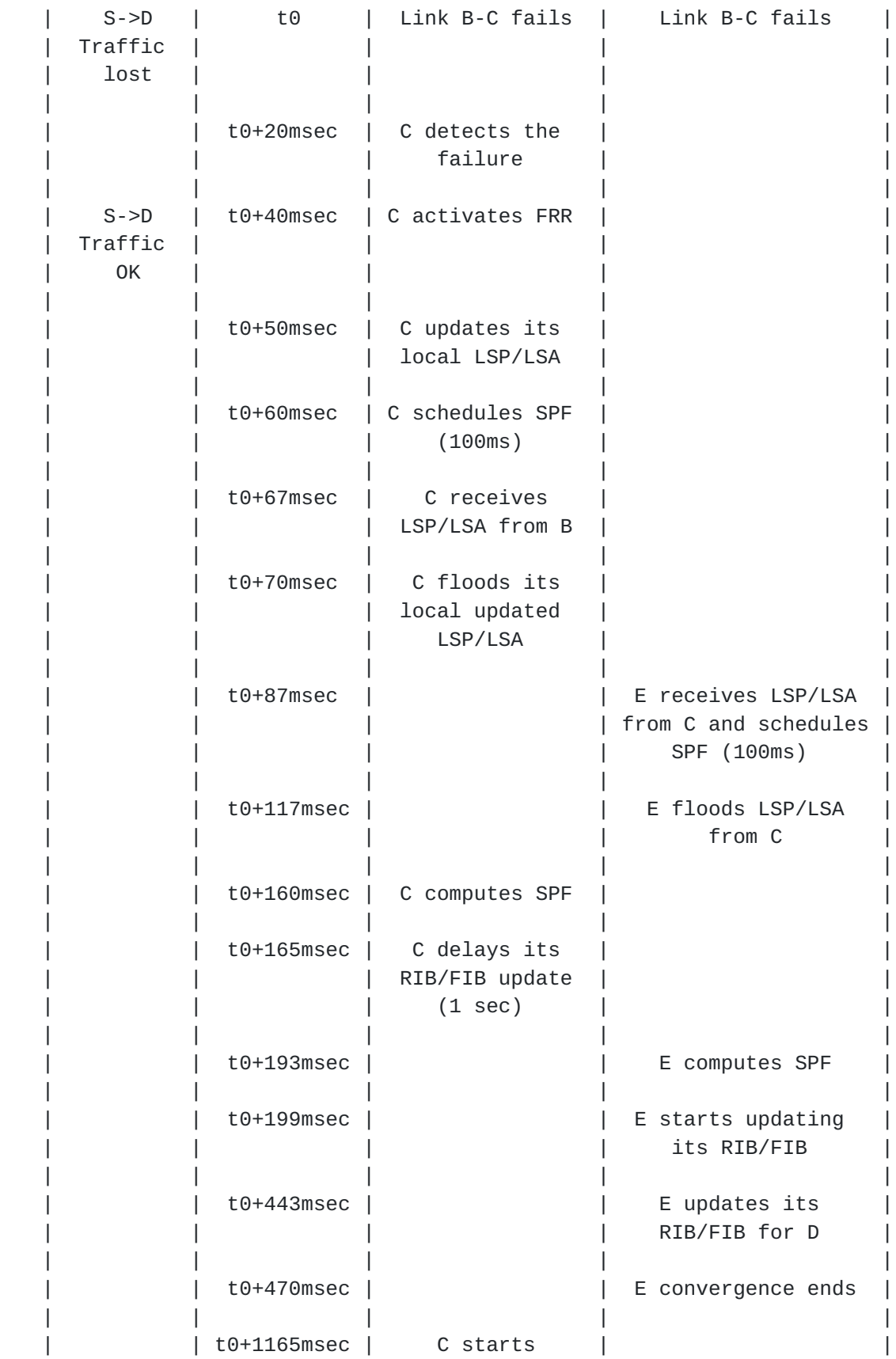


The network above is considered to have a convergence time about 1 second, so ULOOP_DELAY_DOWN_TIMER will be adjusted to this value. We also consider that FRR is running on each node.

8.1. Local link down

The table below describes the events and associating timing that happens on router C and E when link B-C goes down. As C detects a single local event corresponding to a link down (its LSP + LSP from B received), it decides to apply the local delay down behavior and no microloop is formed.

Network condition	Time	Router C events	Router E events
S->D Traffic OK			


S->D Traffic lost
t0
Link B-C fails
Link B-C fails
t0+20msec
C detects the failure
S->D Traffic OK
t0+40msec
C activates FRR
t0+50msec
C updates its local LSP/LSA
t0+60msec
C schedules SPF (100ms)
t0+67msec
C receives LSP/LSA from B
t0+70msec
C floods its local updated LSP/LSA
t0+87msec
E receives LSP/LSA from C and schedules SPF (100ms)
t0+117msec
E floods LSP/LSA from C
t0+160msec
C computes SPF
t0+165msec
C delays its RIB/FIB update (1 sec)
t0+193msec
E computes SPF
t0+199msec
E starts updating its RIB/FIB
t0+443msec
E updates its RIB/FIB for D
t0+470msec
E convergence ends
t0+1165msec
C starts

		updating its RIB/FIB	
	t0+1255msec	C updates its RIB/FIB for D	
	t0+1340msec	C convergence ends	

Route computation event time scale

Similarly, upon B-C link down event, if LSP/LSA from B is received before C detects the link failure, C will apply the route update delay if the local detection is part of the same SPF run.

Network condition	Time	Router C events	Router E events
S->D Traffic OK			
S->D Traffic lost	t0	Link B-C fails	Link B-C fails
	t0+32msec	C receives LSP/LSA from B	
	t0+33msec	C schedules SPF (100ms)	
	t0+50msec	C detects the failure	
S->D Traffic OK	t0+55msec	C activates FRR	
	t0+55msec	C updates its local LSP/LSA	
	t0+70msec	C floods its local updated LSP/LSA	

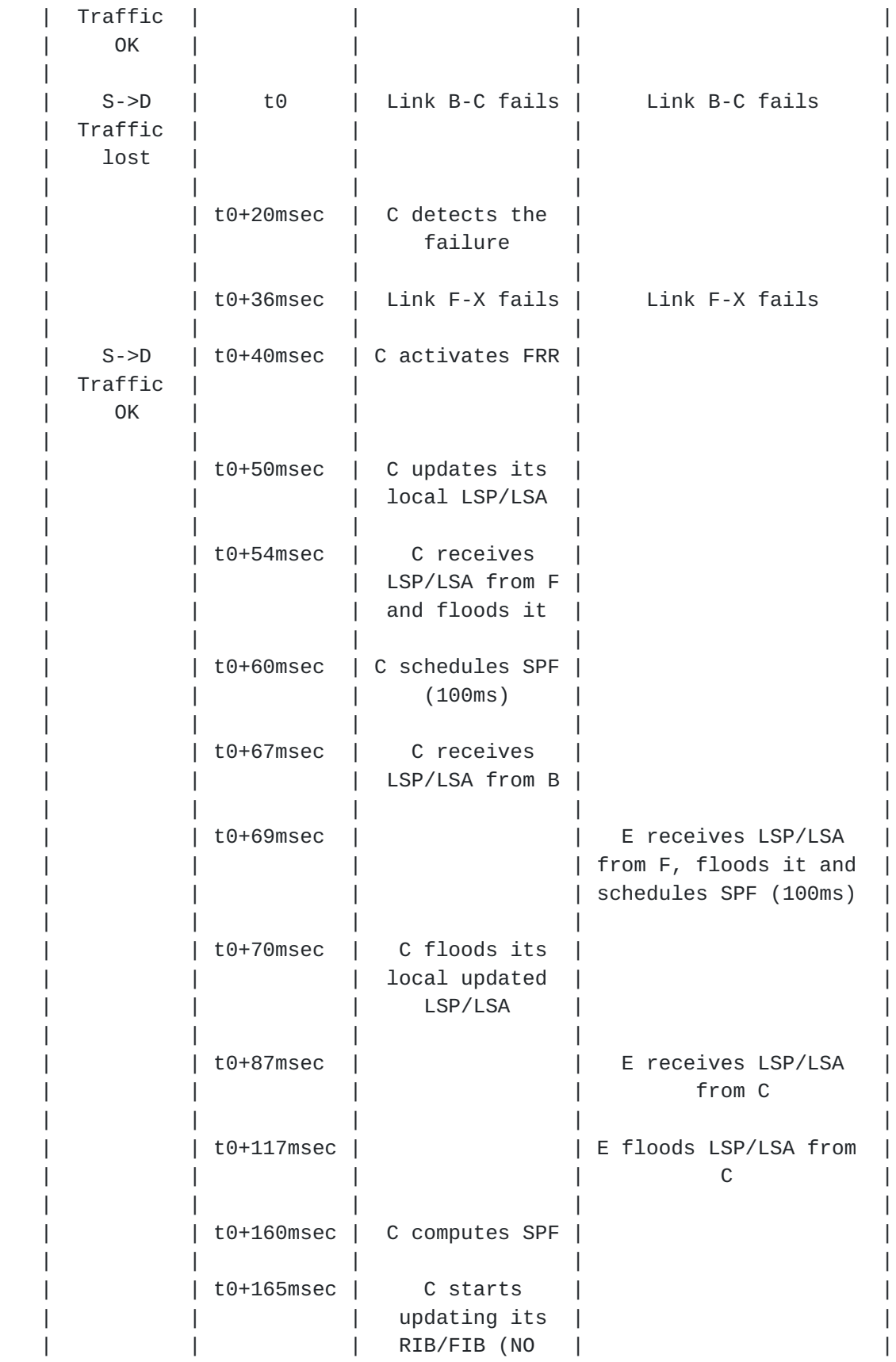
	t0+87msec		E receives LSP/LSA from C and schedules SPF (100ms)
	t0+117msec		E floods LSP/LSA from C
	t0+160msec	C computes SPF	
	t0+165msec	C delays its RIB/FIB update (1 sec)	
	t0+193msec		E computes SPF
	t0+199msec		E starts updating its RIB/FIB
	t0+443msec		E updates its RIB/FIB for D
	t0+470msec		E convergence ends
	t0+1165msec	C starts updating its RIB/FIB	
	t0+1255msec	C updates its RIB/FIB for D	
	t0+1340msec	C convergence ends	

Route computation event time scale

8.2. Local and remote event

The table below describes the events and associating timing that happens on router C and E when link B-C goes down, in addition F-X link will fail in the same time window. C will not apply the local delay because a non local topology change is also received.

Network condition	Time	Router C events	Router E events
S->D			


Traffic OK
S->D Traffic lost
t0
Link B-C fails
Link B-C fails
t0+20msec
C detects the failure
t0+36msec
Link F-X fails
Link F-X fails
S->D Traffic OK
t0+40msec
C activates FRR
t0+50msec
C updates its local LSP/LSA
t0+54msec
C receives LSP/LSA from F and floods it
t0+60msec
C schedules SPF (100ms)
t0+67msec
C receives LSP/LSA from B
t0+69msec
E receives LSP/LSA from F, floods it and schedules SPF (100ms)
t0+70msec
C floods its local updated LSP/LSA
t0+87msec
E receives LSP/LSA from C
t0+117msec
E floods LSP/LSA from C
t0+160msec
C computes SPF
t0+165msec
C starts updating its RIB/FIB (NO

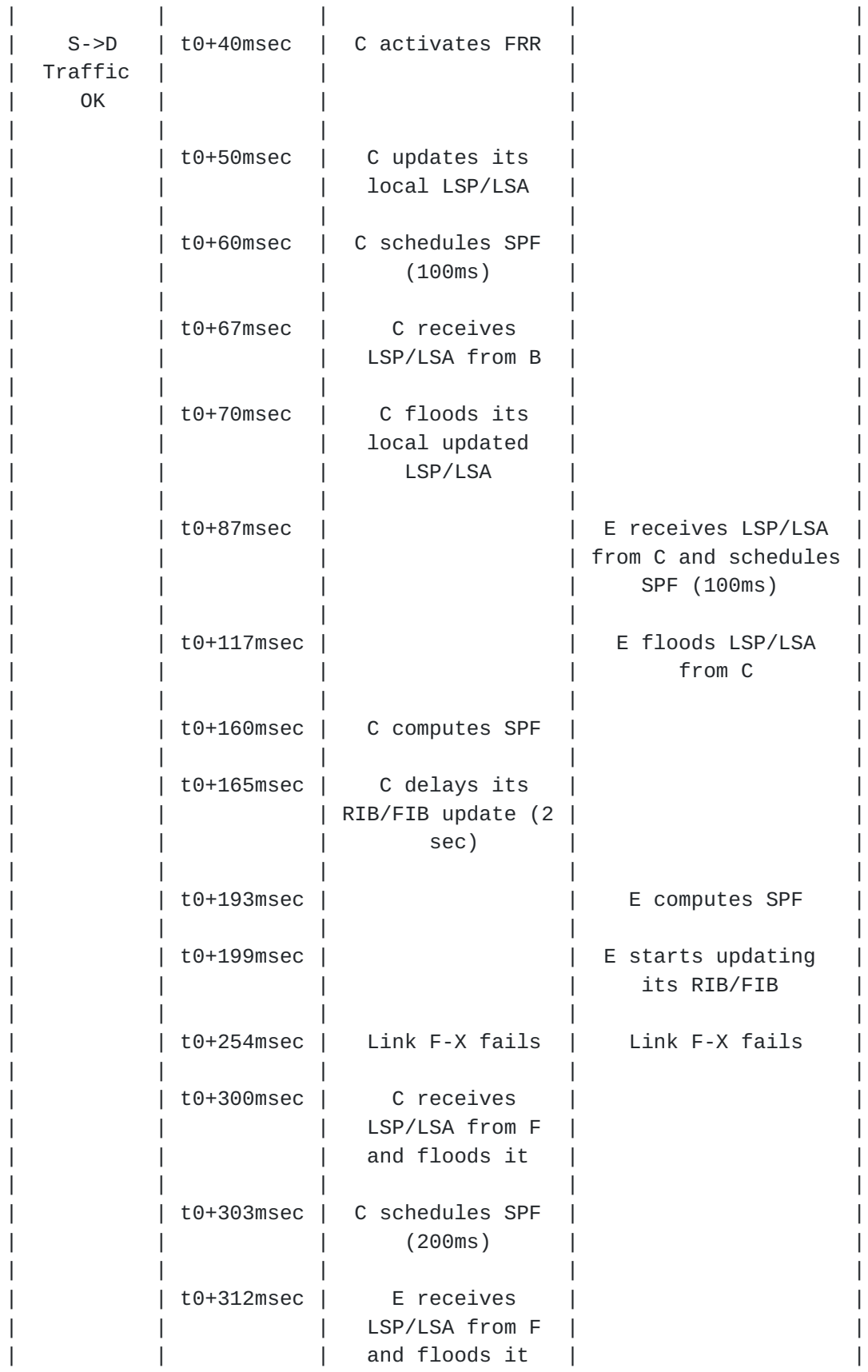
		DELAY)	
	t0+170msec		E computes SPF
	t0+173msec		E starts updating its RIB/FIB
S->D Traffic lost	t0+365msec	C updates its RIB/FIB for D	
S->D Traffic OK	t0+443msec		E updates its RIB/FIB for D
	t0+450msec	C convergence ends	
	t0+470msec		E convergence ends

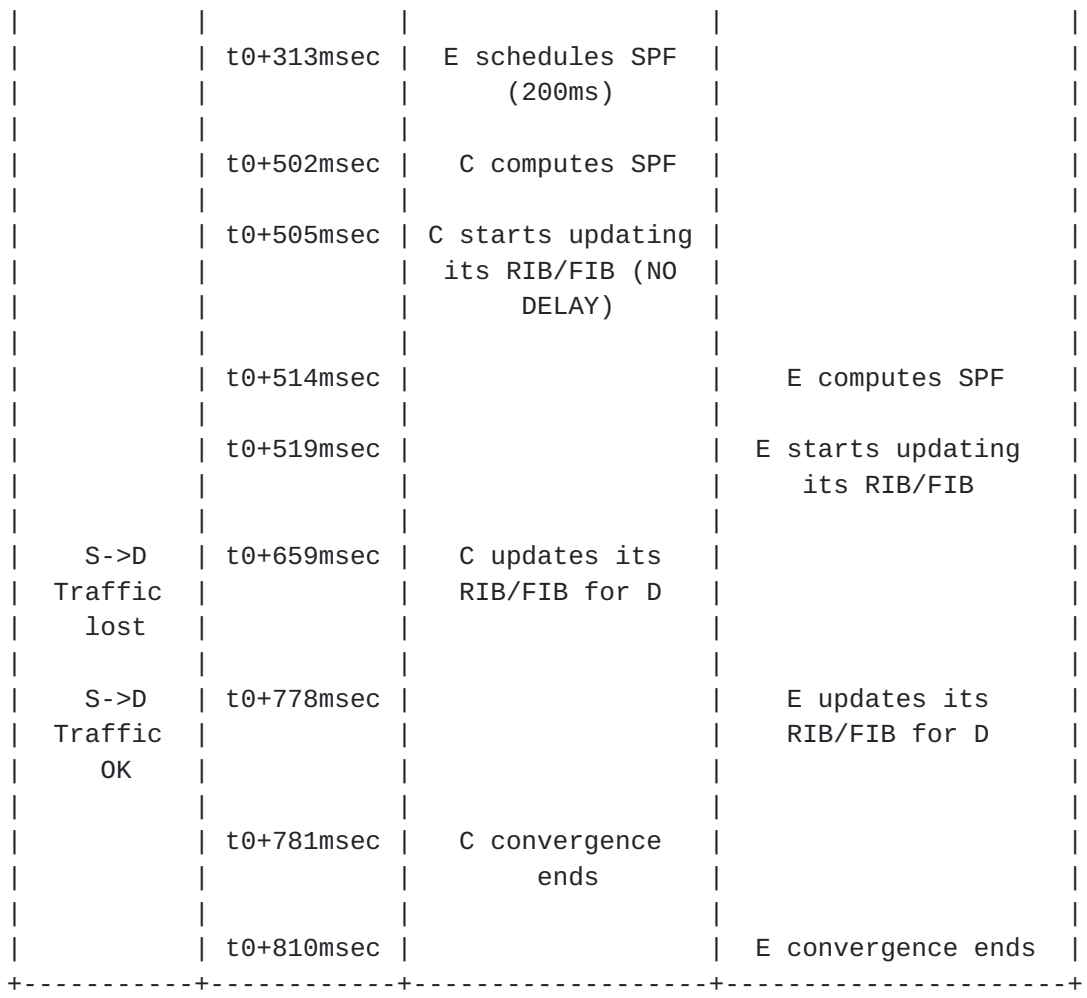
Route computation event time scale

8.3. Aborting local delay

The table below describes the events and associating timing that happens on router C and E when link B-C goes down, in addition F-X link will fail during local delay run. C will first apply local delay, but when the new event happens, it will fall back to the standard convergence mechanism without delaying route insertion anymore. In this example, we consider a ULOOP_DELAY_DOWN_TIMER configured to 2 seconds.

Network condition	Time	Router C events	Router E events
S->D Traffic OK			
S->D Traffic lost	t0	Link B-C fails	Link B-C fails
	t0+20msec	C detects the failure	


S->D Traffic OK
t0+40msec
C activates FRR
t0+50msec
C updates its local LSP/LSA
t0+60msec
C schedules SPF (100ms)
t0+67msec
C receives LSP/LSA from B
t0+70msec
C floods its local updated LSP/LSA
t0+87msec
E receives LSP/LSA from C and schedules SPF (100ms)
t0+117msec
E floods LSP/LSA from C
t0+160msec
C computes SPF
t0+165msec
C delays its RIB/FIB update (2 sec)
t0+193msec
E computes SPF
t0+199msec
E starts updating its RIB/FIB
t0+254msec
Link F-X fails
Link F-X fails
t0+300msec
C receives LSP/LSA from F and floods it
t0+303msec
C schedules SPF (200ms)
t0+312msec
E receives LSP/LSA from F and floods it



Route computation event time scale

9. Comparison with other solutions

As stated in Section 3, our solution reuses some concepts already introduced by other IETF proposals but tries to find a tradeoff between efficiency and simplicity. This section tries to compare behaviors of the solutions.

9.1. PLSN

PLSN ([I-D.ietf-rtgwg-microloop-analysis]) describes a mechanism where each node in the network tries to avoid transient forwarding loops upon a topology change by always keeping traffic on a loop-free path for a defined duration (locked path to a safe neighbor). The locked path may be the new primary nexthop, another neighbor, or the old primary nexthop depending how the safety condition is satisfied.

PLSN does not solve all transient forwarding loops (see [I-D.ietf-rtgwg-microloop-analysis] Section 4 for more details).

Our solution reuses some concept of PLSN but in a more simple fashion:

- PLSN has three different behaviors: keep using old nexthop, use new primary nexthop if it is safe, or use another safe nexthop, while our solution only have one: keep using the current nexthop (old primary, or already activated FRR path).

- PLSN may cause some damage while using a safe nexthop which is not the new primary nexthop in case the new safe nexthop does not enough provide enough bandwidth (see [RFC7916]). Our solution may not experience this issue as the service provider may have control on the FRR path being used preventing network congestion.

- PLSN applies to all nodes in a network (remote or local changes), while our mechanism applies only on the nodes connected to the topology change.

9.2. OFIB

OFIB ([RFC6976]) describes a mechanism where the convergence of the network upon a topology change is made ordered to prevent transient forwarding loops. Each router in the network must deduce the failure type from the LSA/LSP received and computes/applies a specific FIB update timer based on the failure type and its rank in the network considering the failure point as root.

This mechanism allows to solve all the transient forwarding loop in a network at the price of introducing complexity in the convergence process that may require a strong monitoring by the service provider.

Our solution reuses the OFIB concept but limits it to the first hop that experiences the topology change. As demonstrated, our proposal allows to solve all the local transient forwarding loops that represents an high percentage of all the loops. Moreover limiting the mechanism to one hop allows to keep the network-wide convergence behavior.

10. Existing implementations

At this time, there are three different implementations of this mechanism: CISCO IOS-XR, CISCO IOS-XE and Juniper JUNOS. The three implementations have been tested in labs and demonstrated a good behavior in term of local micro-loop avoidance. The feature has also

been deployed in some live networks. No side effects have been found.

11. Security Considerations

This document does not introduce any change in term of IGP security. The operation is internal to the router. The local delay does not increase the attack vector as an attacker could only trigger this mechanism if he already has be ability to disable or enable an IGP link. The local delay does not increase the negative consequences as if an attacker has the ability to disable or enable an IGP link, it can already harm the network by creating instability and harm the traffic by creating forwarding packet loss and forwarding loss for the traffic crossing that link.

12. Acknowledgements

We would like to thanks the authors of [RFC6976] for introducing the concept of ordered convergence: Mike Shand, Stewart Bryant, Stefano Previdi, and Olivier Bonaventure.

13. IANA Considerations

This document has no actions for IANA.

14. References

14.1. 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, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5715] Shand, M. and S. Bryant, "A Framework for Loop-Free Convergence", RFC 5715, DOI 10.17487/RFC5715, January 2010, <http://www.rfc-editor.org/info/rfc5715>.

14.2. Informative References

[I-D.ietf-rtgwg-backoff-algo] Decraene, B., Litkowski, S., Gredler, H., Lindem, A., Francois, P., and C. Bowers, "SPF Back-off algorithm for link state IGPs", draft-ietf-rtgwg-backoff-algo-05 (work in progress), May 2017.

[I-D.ietf-rtgwg-microloop-analysis]
Zinin, A., "Analysis and Minimization of Microloops in Link-state Routing Protocols", draft-ietf-rtgwg-microloop-analysis-01 (work in progress), October 2005.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <http://www.rfc-editor.org/info/rfc3630>.

[RFC6571] Filsfils, C., Ed., Francois, P., Ed., Shand, M., Decraene, B., Uttaro, J., Leymann, N., and M. Horneffer, "Loop-Free Alternate (LFA) Applicability in Service Provider (SP) Networks", RFC 6571, DOI 10.17487/RFC6571, June 2012, <http://www.rfc-editor.org/info/rfc6571>.

[RFC6976] Shand, M., Bryant, S., Previdi, S., Filsfils, C., Francois, P., and O. Bonaventure, "Framework for Loop-Free Convergence Using the Ordered Forwarding Information Base (oFIB) Approach", RFC 6976, DOI 10.17487/RFC6976, July 2013, <http://www.rfc-editor.org/info/rfc6976>.

[RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <http://www.rfc-editor.org/info/rfc7490>.

[RFC7916] Litkowski, S., Ed., Decraene, B., Filsfils, C., Raza, K., Horneffer, M., and P. Sarkar, "Operational Management of Loop-Free Alternates", RFC 7916, DOI 10.17487/RFC7916, July 2016, <http://www.rfc-editor.org/info/rfc7916>.

Authors' Addresses

Stephane Litkowski
Orange

Email: stephane.litkowski@orange.com

Bruno Decraene
Orange

Email: bruno.decraene@orange.com

Clarence Filsfils
Cisco Systems

Email: cfilsfil@cisco.com

Pierre Francois
Individual

Email: pfrpfr@gmail.com