draft-ietf-rtgwg-segment-routing-ti-lfa-01

S. Litkowski, Orange
A. Bashandy, Individual
C. Filsfils, Cisco
B. Decraene, Orange
P. Francois, INSA Lyon
D. Voyer, Bell Canada
P. Camarillo, Cisco
Reminder

• Fast-reroute mechanism
• Segment Routing based but can protect IP or LDP traffic
• « Evolution » of LFA/RLFA/DLFA concepts
• Added value:
  – Topology independent (always provides 100% coverage)
  – By default it is more optimal (less policy tuning requirements) for most of the cases
Substantial comments received...

- Notion of post-convergence path was unclear
- Benefits of using the «post-convergence» path were also unclear
- Scaling considerations when computing Q-Space
- Some unclear points in the dataplane procedures (deal with PHP...)
- Local SRLG vs global SRLG protection
- Relation of TI-LFA with SR-algorithms
We consider traffic from PE1/PE4 to PE2/PE3 and the protection of X-B link.

FRR paths may use X-H or X-D (expected post-convergence path for X to PE2/3).

X reroutes traffic on X-H/X-D after convergence (same path as FRR)

- X-H/X-D links should be sized accordingly

PE1 may reroute onto X-D-E-F-PE2 path upon convergence => some traffic get out of X-H/X-D when PE1 has converged

- TILFA does not provide an « end-to-end » post-convergence path

If X converges before PE1, it is useful to have X-H/X-D links sized accordingly to prevent any congestion: TILFA uses this assumption to provide a well-sized and optimal path

If an operator does not apply this capacity rule, RFC7916 can still be used to tune the FRR path.
« Post-convergence » path

- TI-LFA precomputes an IGP shortest backup path to protect against a particular failure type (node, SRLG, link).

- If a different type of failure happens than the precomputed one, the computed backup path becomes not optimal (from an SPF point of view).

- We introduce the notion of « expected post-convergence path » to define the optimal path computed by TILFA. « Expected » because depending on the real failure happening, it may not be the post-convergence path.
« Post-convergence » path

-01 updates:

- References to RFC7916 (raises the issue of path optimality)
- We now use the notion of « expected post-convergence path »
- Added a use case with an example dealing with path optimality and capacity planning rules
Scaling considerations of Q-Space computation

• We have a short paragraph that refers to RFC7490 which already introduces the scaling issues with per-destination Q-Space computation

• It is up to the implementation to find the good tradeoff between optimization and computation load
Dataplane procedures

- Used vocabulary has been updated to use only SR related words (PUSH, NEXT, ...) vs a mix of MPLS and SR words

- Normative language has been introduced

- SR-MPLS specific behaviors have been clarified when PHP is involved. For instance, when protecting a link S-F, and considering an incoming packet with a stack [ Adj(S-F), Node(T),... ]:
  - The basic behavior is for S to modify the label stack as follows: [RT(F),node(F),node(T)]
  - However, if the repair-list ends with an Adj-SID terminating on F, and if the Node-SID of F has been signalled with PHP, S should modify the packet as follows [RT(F), node-T]
Local SRLG vs global SRLG protection

• The local SRLG restriction has been removed
TILFA and SR-algorithms

- TILFA was introduced when only algo=0 was available
- Now, we have strict SPF as well as Flexalgo

1. Local policies overriding the SPT of algo 0 Node-SID may break loop-freeness of a TILFA path
   - In fact a local policy would even break loop-freeness of LFA or rLFA today or even the IGP shortest path => it’s the responsibility of the operator
   - Strict SPF node SIDs may be used in a repair-list by an implementation to protect a regular SPF Node-SID (algo 0) or and Adj-SID.

2. TILFA can be applied to Flexalgo as far as the Flexalgo remains CSPF based
   - Node-SIDs used in the repair list must use the same FlexAlgo as the protected Node-SID.
Next steps

• We have tried to address the comments received on the list

• Raise your voice if we’ve missed some or if you have additional comments!

• We hope to trigger discussion on several points such as the TI-LFA and SR-algo relationship

• The main point to agree on is about the required level of detail about computations as each implementation has its own secret sauce to optimize the computation.