SPF_DELAY

draft-litkowski-rtgwg-spf-uloop-pb-statement-02
draft-decraene-rtgwg-backoff-algo-01

Bruno Decraene Orange
Stéphane Litkowski Orange

draft-litkowski-rtgwg-spf-uloop-pb-statement

1. Problem statement.
2. Analyze how mixing different link state IGP implementations may increase micro-loops occurrence / duration.
3. Calls for a more standardized behavior of some components.
   - Mono vendor (network) is not an option, sorry ;-)

- In particular a standardized SPF back-off delay algorithm.
  - The most straightforward change
  - The biggest component to micro-loops
    - may add more than 1 or 2 second of micro-loops in typical deployments
    - may add micro-loops every time the back off algo is used.
    - cf IETF 90 (Toronto) meeting / slides

draft-decraene-rtgwg-backoff-algo-01

- Focus on standardizing a SPF back off delay (algorithm)
  - vendors would still be free to have a custom one,
  - IETF would still be free to define another one in the future.

1. Discuss requirements

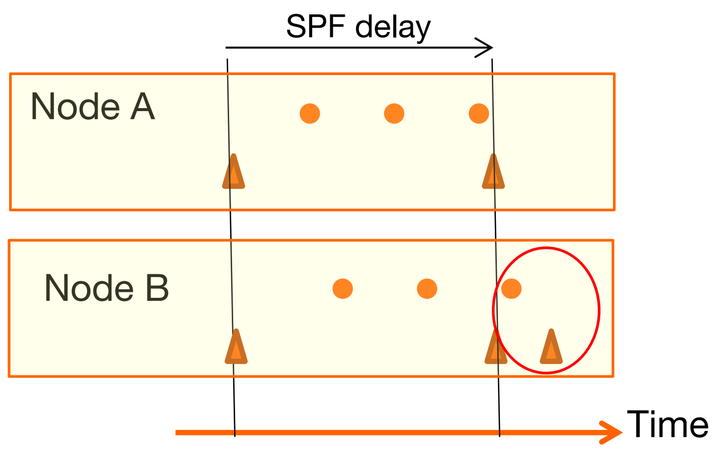
2. Proposes one algorithm

SPF back off delay: requirements (1)

- Very fast convergence for single simple events (link failure).
  - majority of failures
  - a single LSP/LSA is enough (no need to wait for more)
- Fast convergence in general
  - in nominal situations when the IGP stability is considered “under control”.
- A long delay when the IGP stability is considered “out of control”
  - in order to let all related processes calm down.
  - all previous quick SPF did not seemed to solve the issue anyway.

SPF back off delay: requirements (2)

- At any time, try to avoid using different SPF_DELAY values on nodes.
  - Even though all nodes do not receive IGP messages at the same time
    - due to difference in distance from the source
    - due to different flooding implementations on the path from the source.



- Nodes A performs 2 SPFs while node B performs 3 SPFs.
- ➔ if SPF_DELAY increases after each SPF, different nodes will use different SPF_DELAY
  - Even though nodes A & B have the same SPF delay algorithm and see the same LSPs.

Legend:
LSP reception ●
SPF computation ▲

A proposed SPF_delay algorithm (1) Definition

- IGP event:
  - An LSDB change requiring a new RIB computation
    - topology change, prefix change, metric change…
  - No distinction done between the type of computation performed
    - e.g. full SPF, incremental SPF, PRC…
    - The type of computation is a local consideration
      - allowing for liberty and different strategy between vendors.

A proposed SPF_delay algorithm (2)
To initiate discussion

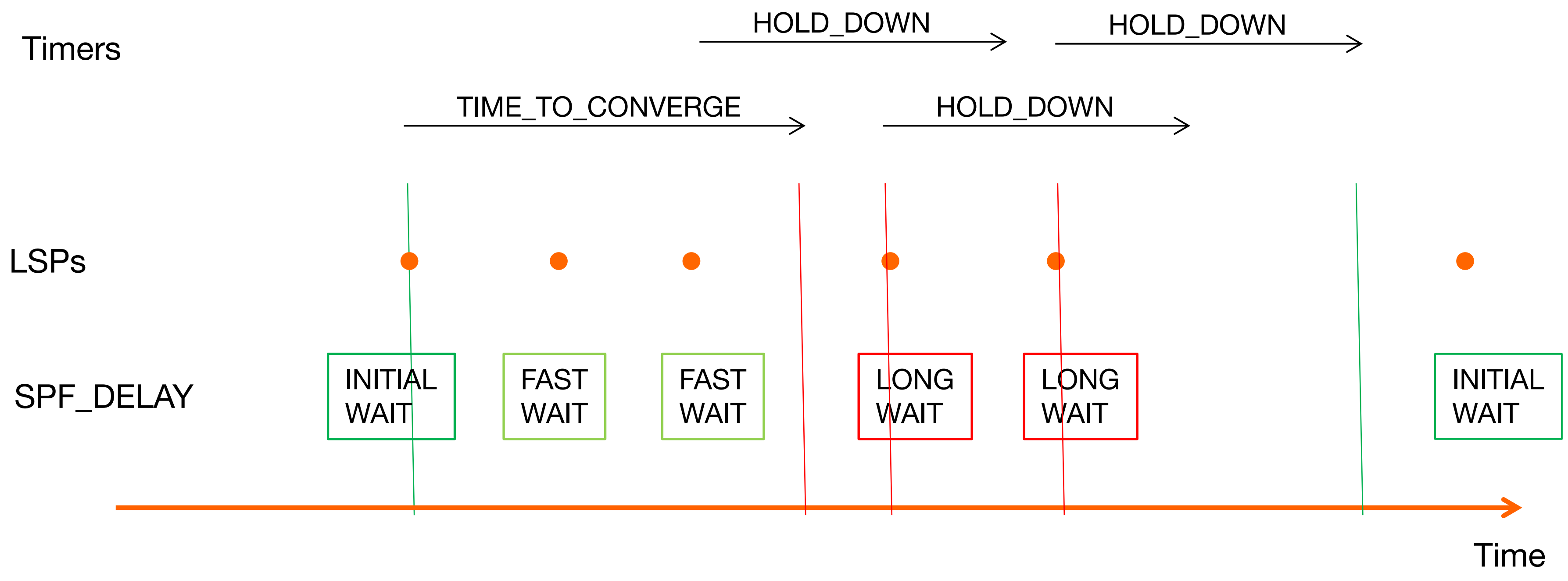
- Only 3 SPF_DELAY values
  - INITIAL_WAIT: a very small delay to quickly handle first event
    - e.g. 0 millisecond
    - Target: quickly react to link failure.
  - FAST_WAIT: a small delay to have a fast convergence. e.g. 50-100 millisecond.
    - we want to be fast, but not too fast: as this failure requires multiple IGP events, being too fast increase the probability to receive additional IGP events just after the RIB computation.
    - Target: node failure, SRLG failures
  - LONG_WAIT: a long delay as IGP is unstable.
    - Let's bring calm in the IGP. e.g. 2 seconds.
    - Target: the unknown / the ugly

A proposed SPF_delay algorithm (3)

- First IGP event:
  - trigger SPF after INITIAL_WAIT
- Subsequent IGP events:
  - trigger SPF after FAST_WAIT
  - until from TIME_TO_CONVERGE since first IGP event
- Subsequent IGP events:
  - trigger SPF after LONG_WAIT
  - until no IGP events during HOLD_DOWN

SPF_delay algorithm Summary

Legend:
LSP reception ●



Discussion

- Proposed algorithm seems to fill all requirements.
- But still, the purpose of this algo is to initiate discussion.
  - not meant to be final.

Next steps

- Both drafts presented in IETF 90 (Toronto).
- Good discussions both during the meeting and on the mailing list.
  - and off list discussions
- Would like to request WG adoption of both drafts.
- Discussion on the mailing list about requirements and algorithms.

Thank you