Traffic Engineering architecture for services aware MPLS

draft-fuxh-mpls-delay-loss-te-framework-02

Xihua Fu, Malcolm Betts, Qilei Wang ***ZTE***

Vishwas Manral, ***Hewlett-Packard Corp.***

Dave McDysan, Andrew Malis ***Verizon***

Spencer Giacalone ***Thomson Reuters***

John E Drake ***Juniper***

Changes since last version

- Introduced at 79th,80th CCAMP and transferred to MPLS after 80th.
- Refined the structure of document
- Added or refined some texts to keep it consistent with following document
  - *draft-giacalone-ospf-te-express-path-01.txt*
  - *draft-previdi-isis-te-metric-extensions-00.txt*
  - *draft-fuxh-mpls-delay-loss-rsvp-te-ext-00.txt*
- Solved some comments from mailing-list

Open Issues: Latency and jitter of Node

- Advertising node latency may result in oscillation risk because of queue delay.
  - Option 1: We may define it as a fixed or average/approximate latency (without any queuing) and add half of the fixed node latency to each link.
  - Option 2: Assumed that the node latency is a small factor of the total latency in the networks. The node latency is hence ignored for the benefit of simplicity.
- Suggestion: Queuing delay isn't considered. Node latency can be included in the advertised link delay.
- More comments from list?

Open Issues: Anomalous state of link in IGP

- One maximum threshold could be configured to link. If the link performance exceeds the threshold, the IGP should get the anomalous state of this link.
  - Path computation entity may not select this kind of link although end-to-end performance is still met.
  - The solution should support to move one end-to-end path away from any link whose performance exceeds the configured maximum threshold.
- It may result in heavy configuration work.
- More comments from list?

Open Issues: Composite Link Performance Advertisement

- Option 1: Only TLV for Composite Link. The performance may be the range, average or maximum latency/loss of all component links.
- Option 2: Both a TLV for each component link, plus one for the bundle with the average.
- It is related specific implementation. More comments from list?

Clarification

- E2E loss computation:
  - 1-(1-lossrate_L1)∗(1-lossrate_L2)*…*(1-lossrate_Ln)
  - Assume packet loss is 10% for two hops of a link. The measurements will come to 19% total packet loss. Because of 10% loss on the first link only 90% packet reach the second link where another 10% of 90% are lost, which is 9% of total packets.
- There will be a little bit change in next version to clarify the e2e loss computation is multiplication of each link rather than sum.
- FYI: ITU-T Y.1541

Next Step

- Collect feedbacks from the meeting and list
- Adopt it as WG document ?