SFC Trace Issue Analysis and Solutions

IETF 94 Yokohama
draft-yang-sfc-trace-issue-analysis-00
Xu. Yang
L. Zhu
G. Karagiannis
Requirements on SFC trace

• Contained in sfc-oam-framework
  – Ability to trigger action from every transit device on the tested layer towards an SF or through an SFC, using TTL (Time To Live) or other means.
  – Ability to trigger every transit device to generate response with OAM code(s) on the tested layer towards an SF or through an SFC, using TTL or other means.
  – Ability to discover and traverse ECMP paths within an SFC.
  – Ability to skip un-supported SF's while tracing SF's in an SFC.
Issues of current solution

- Un-supported SFs
- Reporting SFF information
- ECMP support
- How to send report message to OAM controller
- More command parameters
- TTL-agnostic solutions
Trace header

- **LSI**: last service index, used to record the service index of the last service function processed the packet.
- **Number Index**: number of hops the packet has traversed, default value is 0.
- **Reserved flags**: can be used to indicate the function blocks that need to send reports, whether uses ECMP, etc.
- **Next Hop Len & Info**: records information of the next hop.
Un-supported SFs

• move all the trace logic to SFF, make no assumption that the SF supports trace

• the SFF will provide information of the SF, if the SF supports trace, it can modify/replace the SFF’s provided information
Reporting SFF information

• Report information of SFFs to form a complete view of the SFP, similar to the process illustrated in the un-supported SFs section.

• Following the traditional traceroute (TTL-like) design, only SFFs between the last and second-to-last hops send reports:
  – comparing LSI/Sl with SIL (1 greater than SIL)
  – the reports will be ordered by number index on OAM controller

• Report the identification of the SFF, e.g., name, IP address, etc.
Step 1: receive trace packet

Step 2: SF reduce SI

Step 3: next SF is the last hop, add Next Hop Info field

Step 4: SFF modify the LSI, forward the packet

Step 5: generate SFF report

Step 6: forward to last hop, add SF's information to the Next Hop Info field

Step 7: SF reduce SI, if support trace, it can update the information

Step 8: drop the trace packet, generate SF report
ECMP support

- SFF broadcasts the trace packet on all the possible (equal-cost) paths
- the trace packet needs to identify the exact path the packet traversed
ECMP support

Step 1: receive trace packet
Step 2: broadcast the packet to all possible (equal-cost) paths
Send report to OAM controller

• the OAM control plane can be centralized or distributed

• centralized case:
  – the trace report packet can be forwarded to the control plane directly

• distributed case:
  – the OAM control entity may not be directly connected with the SFF
  – a dedicated control path or a reverse path is needed
More command parameters

• The following parameters are worth to be taken into consideration:
  - report object: sending report of SF, SFF or both
  - ECMP support
  - number of queries to send per hop
  - time to wait for a response/report
  - number of queries that can be sent out simultaneously
  - time interval between sending queries
TTL-agnostic solution

• It’s not necessarily to use a TTL-like way to conduct SFC trace since there’s no TTL field in the current NSH
• TTL-agnostic way: send only one trace packet, which will traverse the service path and trigger report on every SFF/SF it passes.
• The controller will reorder the received reports and show the status of the SFP.
Thanks