Synonymous Flow Labels

S. Bryant: Independent
Swallow, S. Sivabalan & C Pignatoro: Cisco
G. Mirsky : Ericsson
M. Chen & Z. Li :Huawei

The Drafts

- draft-ietf-mpls-flow-ident
- draft-bryant-mpls-sfl-framework
- draft-bryant-mpls-rfc6374-sfl
- draft-bryant-mpls-sfl-control
- draft-bryant-mpls-rfc6374-over-udp

The Purpose Today

- To restart this project after a slight pause.
- To ask the WG a number of questions about a number of design decisions that we need to make.

Revision: Synonymous Flow Label

A synonymous flow label (SFL) is a label that causes the Egress LSR to perform a previously agreed action **in addition** to processing and delivering the packet in exactly the same way as the label that it is synonymous with (except if the action says otherwise).

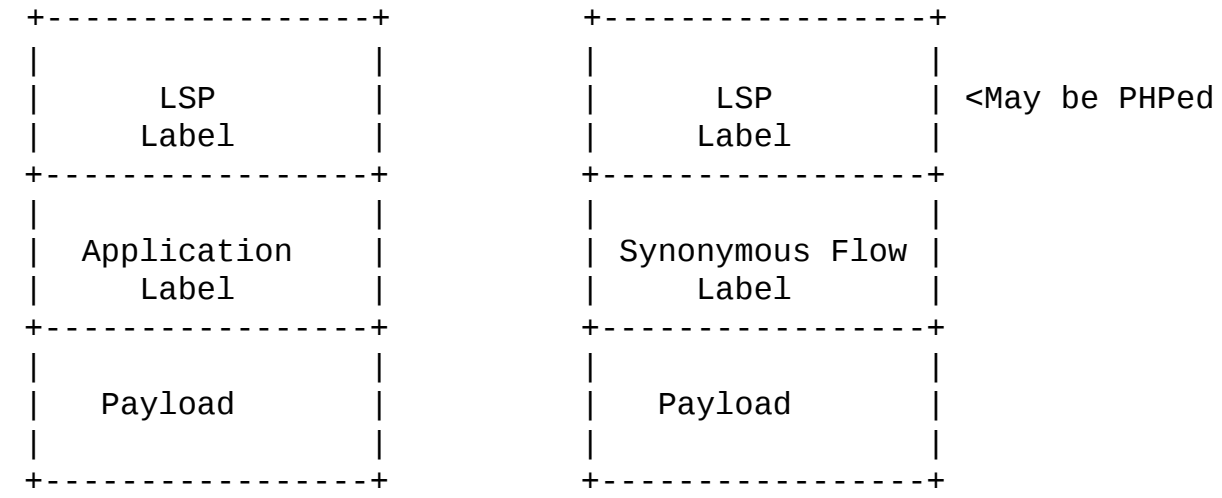
The action may be increment a counter, log a packet, or anything else that is agreed between the MPLS peers.

The additional action that RF6374 needs is the incrementing of a flow specific counter, something that many LSRs can already do!

Revision: Other Possible Synonymous Actions

- Record this packet
- Get IPFIX to look at this packet
- DPI this packet
- Send this packet for DOS washing
- …..

Revision: Synonymous with Application Label



"Normal" Label Stack

Label Stack with SFL

draft-ietf-mpls-flow-ident

- This is the requirements draft.
- When this was written we really had RFC6374 loss and delay in mind.
- After it was written we thought of other things we might do with SLs. Possibly others have thought of further applications since then?
- When this was written we were thinking of garden variety LSPs. What about:
  - Segment Routed LSPs?
  - LSPs under repair via FRR, or fast repair?
  - Do we need to consider them, of can we mark them FFS?
- Do we need to add other applications, and/or other requirements?
- If we have nothing to add, this is pretty much complete.

draft-bryant-mpls-sfl-framework

- Our judgement on the completeness of this draft needs to be based on our discussion of the requirements draft.
- Do we add SR, and/or FRR, and/or Prot Sw?
- If not this is pretty much complete, other than adding some text about control plane, and management.

draft-bryant-mpls-rfc6374-sfl-00

- Initial draft describing how we run RFC6374 in conjunction with SFLs.
- Needs two major additions
  - Delay
  - More thought on multiplexing the RFC6374 message
- Assumption is that we would mux using GAL, but we could use another SL or perhaps send it over UDP, or send it some other way.
- Is everyone happy with using GAL, or do we need to look at another method?

Control Protocol

- We need a method of exchanging SLs.
- What type of control protocol do we need?
  - Extension to the existing control planes (LDP + RSVP + SDN + Operator Config etc)?
  - A purpose designed control plane that complements the existing control protocols?
  - (Many) Application extensions (let them manage their own labels)

A Purpose Designed CP

- Has the advantage that we design it once and it applies to all MPLS applications.
- Means that we do not need to touch the existing CPs (assuming we can make it work in all cases!)
- Has the advantage and possible disadvantage that the existing CPs don’t know SL is happening.
- Has the disadvantage that operators will need to understand, configure and manage a new protocol.

draft-bryant-mpls-sfl-control-00

- A request/response/refresh/die-of-old-age protocol that runs over an ACH.
- Very much from the MPLS-TP stable rather than the LDP/RSVP/BGP stable.
- In the style that you would expect of an OAM control protocol.
- Should we continue to develop this, or should we use a different approach, or should we adapt the exist control protocols?

draft-bryant-mpls-rfc6374-over-udp

- This was just a thought piece considering the case where we could not use GAL as the mux.
- Is this a scenario that we need to consider, or can we regard GAL support as an invariant?
- We sketched out UDP, do we need to consider any other approach for example using SLs as the mux (one for the data and a further one for the application/OAM)
  - i.e. normal == data == application (e.g. RFC6374)