Service Function Chain Proof of Transit

Hc2vpp Agent

Ver. 0.1

Author	Date	Notes
Srihari Raghavan (srihari@cisco.com)	7 Apr 2017	Initial Version – v0.1

Table of Contents

1. Introduction

Service Function Chain (SFC) Proof of Transit implemented as a module in Hc2vpp is based on the concept of in-band OAM [1]. In-band OAM is an implementation study to record operational information in the packet while the packet traverses a path between two points in the network. More specifically, SFC Proof of Transit via in-band OAM is meant to prove that data or user traffic indeed follows the suggested path or a SFC.

To ensure that the data traffic follows a specified path or a function chain, meta-data is added to all user traffic. The meta-data is based on a 'share of a secret' and provisioned by opendaylight controller over a secure channel to each of the nodes that implement the service function chain. This meta-data is updated at each of the service-hop while a designated node called the verifier checks whether the collected meta-data allows the retrieval of the secret.

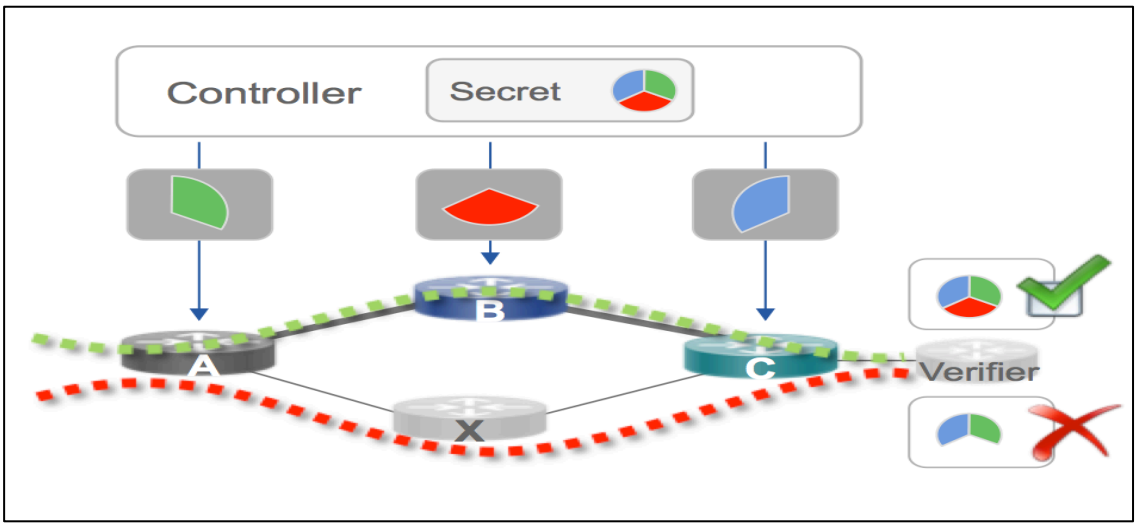

Fig 1: SFC Proof of Transit overview

The proof of transit via a SFC or a path via this solution utilizes Shamir's secret sharing algorithm [2], where each service is given a point on the curve and when the packet travels through each service, it collects these points (meta-data) and a verifier node tries to re-construct the curve using the collected points, thus verifying that the packet traversed through all the service functions along the chain.

Transport options for different protocols includes as a new TLV in SRH header for Segment Routing, Network Service Headers (NSH) Type-2 meta-data, IPv6 extension headers, IPv4 variants and for VXLAN-GPE as well.

2. SFC Proof of Transit and Hc2vpp

Hc2vpp is a java-based agent that runs on the same host as a Vector Packet Processing [3] instance and exposes yang models via NETCONF or RESTCONF to allow the management of that VPP instance from off the box [3]. Hc2vpp is the VPP specific part of the generic Honeycomb project. The following diagram from [3] provides an idea of the interactions involved.

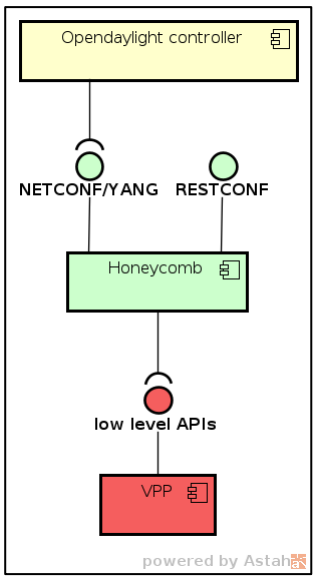

Fig 2: Hc2vpp (Honeycomb) interactions with VPP and ODL

The iOAM agent in Hc2vpp (17.01+ versions) is a Java plugin API to help the agent configure the VPP instance with iOAM feature-set. The feature assumes that the SFC path that is set up in the corresponding VPP nodes is configured to enable the Proof of Transit iOAM feature.

3. Pre-Requisites

- A native Ubuntu X distribution or running as a Virtualbox image with interface
- Java 8 and Maven
- VPP instance running as a Ubuntu X distribution or in a Virtualbox image.
- JVpp (JNI based Java API for VPP) infrastructure as outlined in [4].

4. SFC Proof of Transit Hc2vpp Architecture

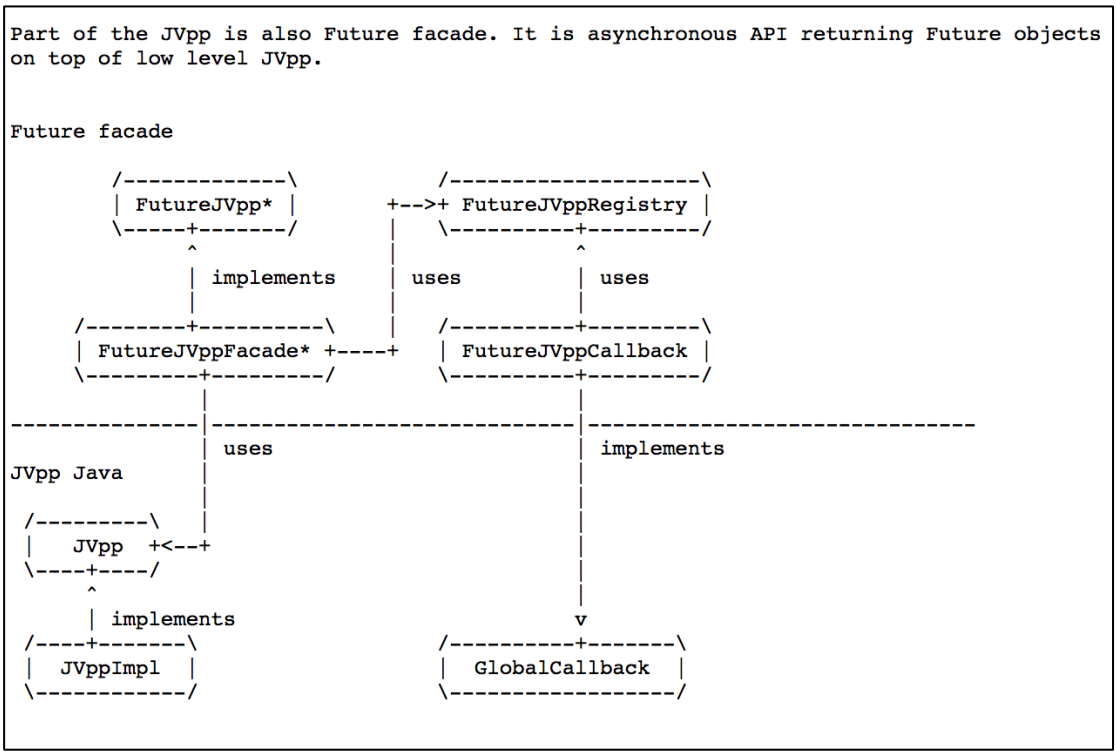

Fig 3: Hc2vpp Jvpp future façade [4]

The general architecture relies on the following aspects:

1. The yang file that describes the Proof of Transit configuration fields are as in: [Link]

```
grouping pot-profile {
 description "A grouping for proof of transit profiles.";
 list pot-profile-list {
  key "index";
  ordered-by user;
  description "A set of pot profiles.";
  leaf index {
   type profile-index-range;
   mandatory true;
   description
    "Proof of transit profile index.";
  }
  leaf prime-number {
   type uint64;
   mandatory true;
   description
    "Prime number used for module math computation";
  }
  leaf secret-share {
   type uint64;
   mandatory true;
 description
    "Share of the secret of polynomial 1 used in computation";
  }
  leaf public-polynomial {
   type uint64;
   mandatory true;
```

2. The following iOAM proof of transit configuration APIs are needed from the VPP side.
    a. https://gerrit.fd.io/r/#/c/3884/
3. The following iOAM proof of transit configuration API support in Hc2vpp are needed at the Honeycomb side.
    a. https://gerrit.fd.io/r/#/c/4268
4. The following table provides the details of the various yang files and classes that comprise the feature and a description of the same.

Index	File Name	Purpose	LoC	Notes
1	Sfc-ioam-sb-pot.yang (SB)	Tracks iOAM PoT parameters that gets sent by the controller to the VPP side via HC agent.	~160	
Index	**Class Name**	**Purpose**	**LoC**	**Notes**
1	VppIoamModule	This binds to the VPP plugin's JVPP, identifies classes to be picked up by the HC framework etc.,	~25	
2	IoamPotWriterCustomizer	This handles the CRUD operations with calls into FutureJVppIoamXXX classes	~150	
3	VppIoamWriterFactory	Factory class for FutureJVppIoamXXX classes	~30	
4	FutureJvppIoamPotCustomizer	Abstract utility to hold the IoamAPI references	~25	
5	JVppIoamPotProvider	Provides Future API (FutureJVppIoampotFacade) for jvpp-ioam plugin	~25	
6	Test Classes			

5. Glossary

SF - Service Function
SFF - Service Function Forwarder
SFC - Service Function Chain
SFP - Service Function Path
RSP - Rendered Service Path
SFC PoT - Service Function Chain Proof of Transit
iOAM – In-band OAM

6. References

1. In-band OAM (iOAM) - https://github.com/ciscodevnet/ioam
2. Shamir's secret sharing - Link
3. Hc2VPP (https://wiki.fd.io/view/Hc2vpp)
4. JVpp (Link)