

Use Cases and Framework of Service-Oriented MPLS Path Programming (MPP)

draft-li-mpls-path-programming-00

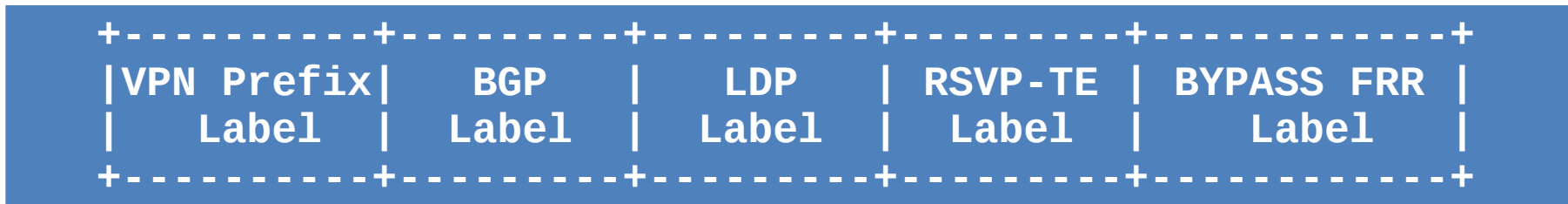
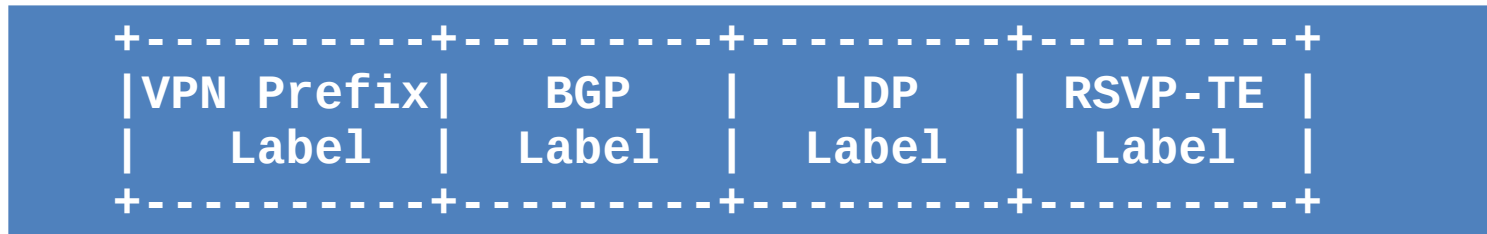
Zhenbin Li, Shunwan Zhuang
Huawei Technologies

Introduction

- SPRING architecture for unicast traffic (Segment Routing) has been proposed to cope with the use cases in traffic engineering, fast re-reroute, service chain, etc. It can leverage existing MPLS data plane without any modification.
- In fact, the label stack capability in MPLS would have been utilized well to implement flexible path programming to satisfy all kinds of requirements of service bearing.
- This document defines the concept of MPLS path programming, then proposes use cases, architecture and protocol extension requirements in the service layer for the SPRING architecture.

History Review (1)

- Hierarchical LSP: e.g. Option C Inter-AS VPN which adopts LDP over TE as the transport tunnel in the ingress node.



- The MPLS label stack in the MPLS path of the example shows limited programming capability. The limitation has two reasons:
 - The limited label usage. MPLS label is always for reachability.
 - The limited path calculation capability in the distributed environment. SPF is always adopted or complex configuration for traffic engineering.

History Review (2)

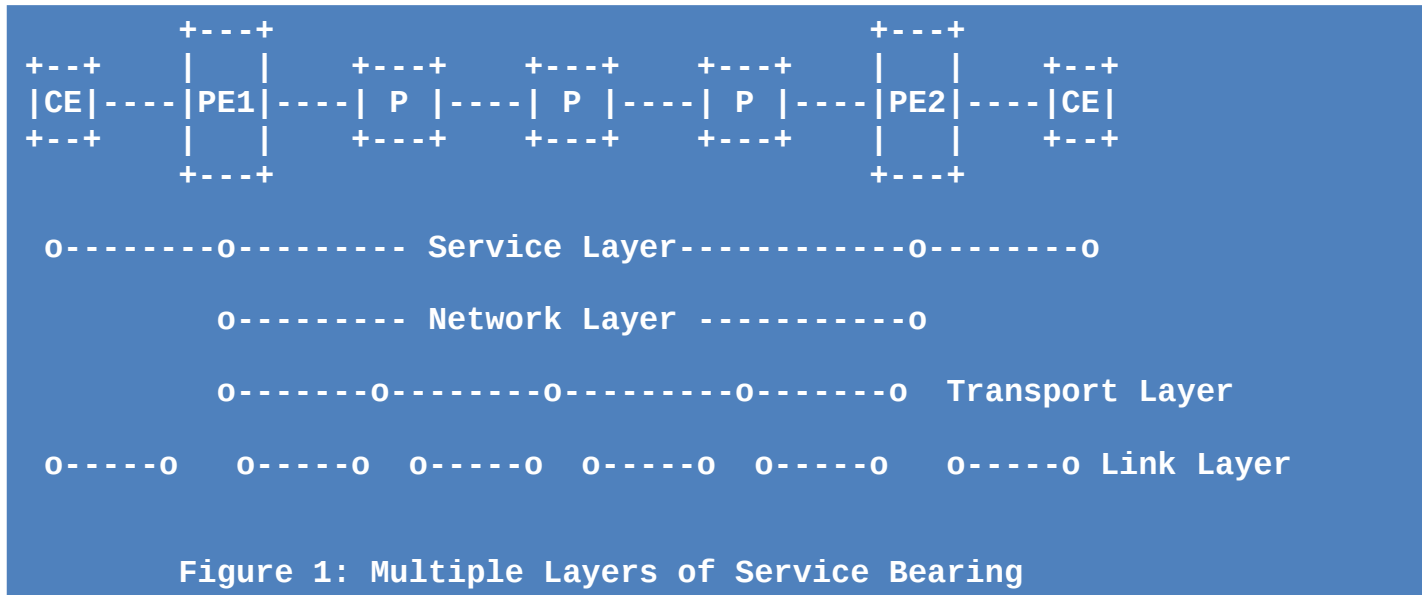
- **MPLS Label beyond reachability:**
 - Entropy Label
 - Source Label
 - Global label use cases defined in [draft-li-mpls-global-label-usecases].
- **Central Control for Enhanced MPLS Path Calculation and label combination.**
 - Stateful PCE
 - PCE for Segment Routing

MPLS Path Programming Capability (1)

- MPLS path is composed by label stacks. Since in the label stack the labels in different layers can represent different meaning and the depth of the label stack can be unlimited in theory, it is possible to make up all kinds of MPLS paths based on the combination of labels.
- If we look on the combination of MPLS labels as programming, it can be seen that the MPLS path has high programming capability.
- As the introducing of central control in the network, the flexible MPLS programming capability becomes possible owing to two factors: **1. It becomes easier to allocate label for more purposes than reachability; 2. It is easy to calculate the MPLS path in a global network view.**

MPLS Path Programming Capability (2)

- There are multiple layers for MPLS path to bear services which is shown in the following figure:



- Two types of MPLS path programming:
 - Transport-Oriented MPLS path programming: Segment Routing, etc.
 - Service-Oriented MPLS Path programming

Use Cases of Service-Oriented MPLS Path Programming

- Traffic Steering in Service/Network Layer : This method is to directly encapsulate the service flow with the service label stack in the ingress PE before it enters into the transport tunnel.
- Use Cases for Unicast Service
 - Basic Reachability
 - VPN Identification
 - ECMP(Equal Cost Multi-Path)
 - Service OAM
 - Traffic Steering
- Use Cases of Multicast Service
 - Basic Reachability
 - MVPN Identification
 - Source Identification
- Use Cases of MPLS Virtual Network

Service-Oriented MPLS Path Programming

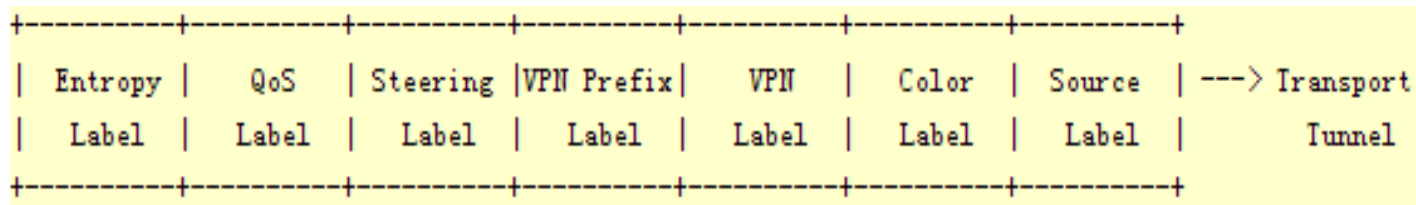
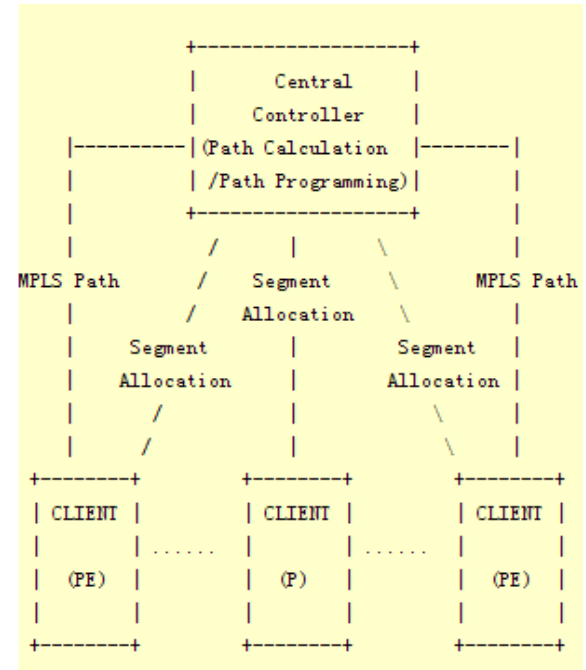
Service Label : MPLS label beyond reachability.

MPLS label becomes the instruction to indicate specific service process.

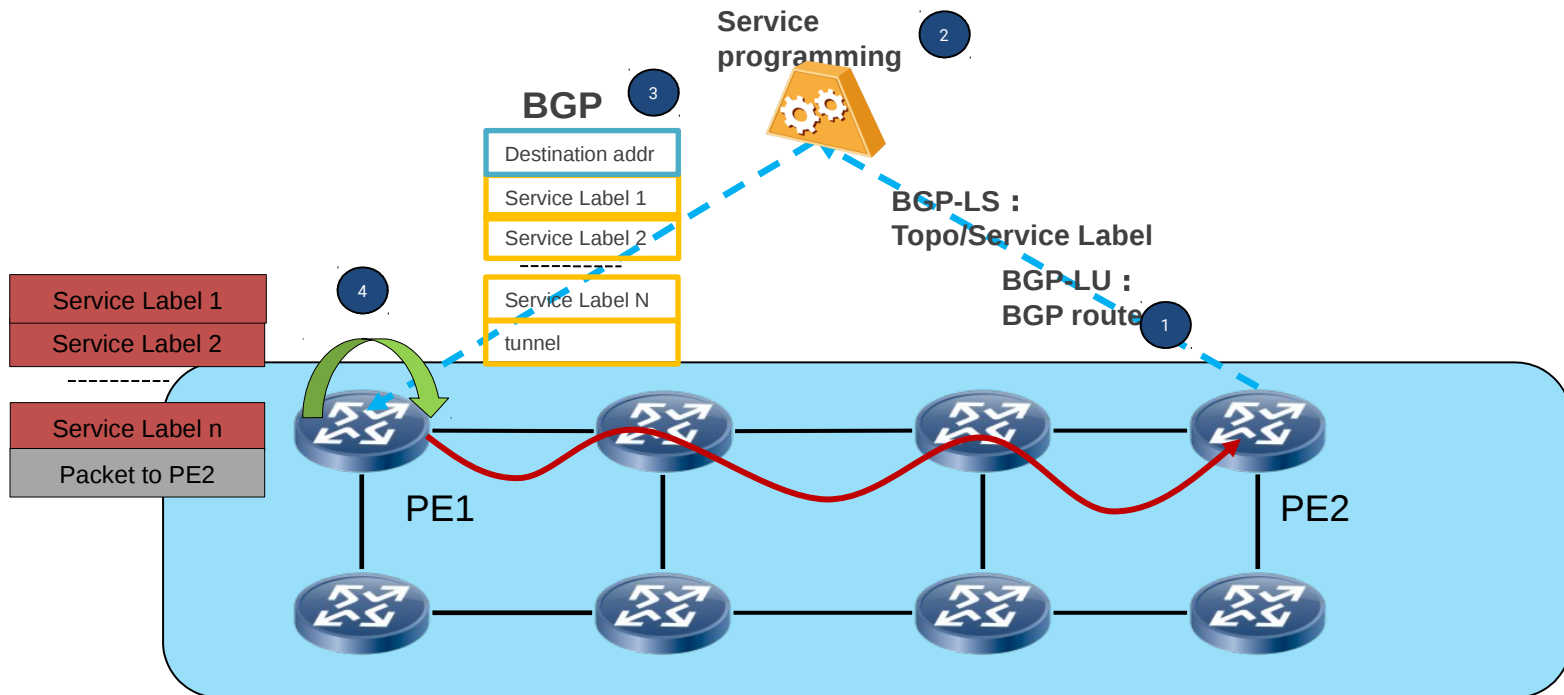
- Entropy Label/Flow Label
- Synonymous Flow Label
- Steering Label
- BW-Guaranteed Label
- QoS Label
- VPN Label
- etc.

Service-Oriented MPLS Path

- Programmed by Central Controller based on service requirement.
- BGP-based download for specific flow



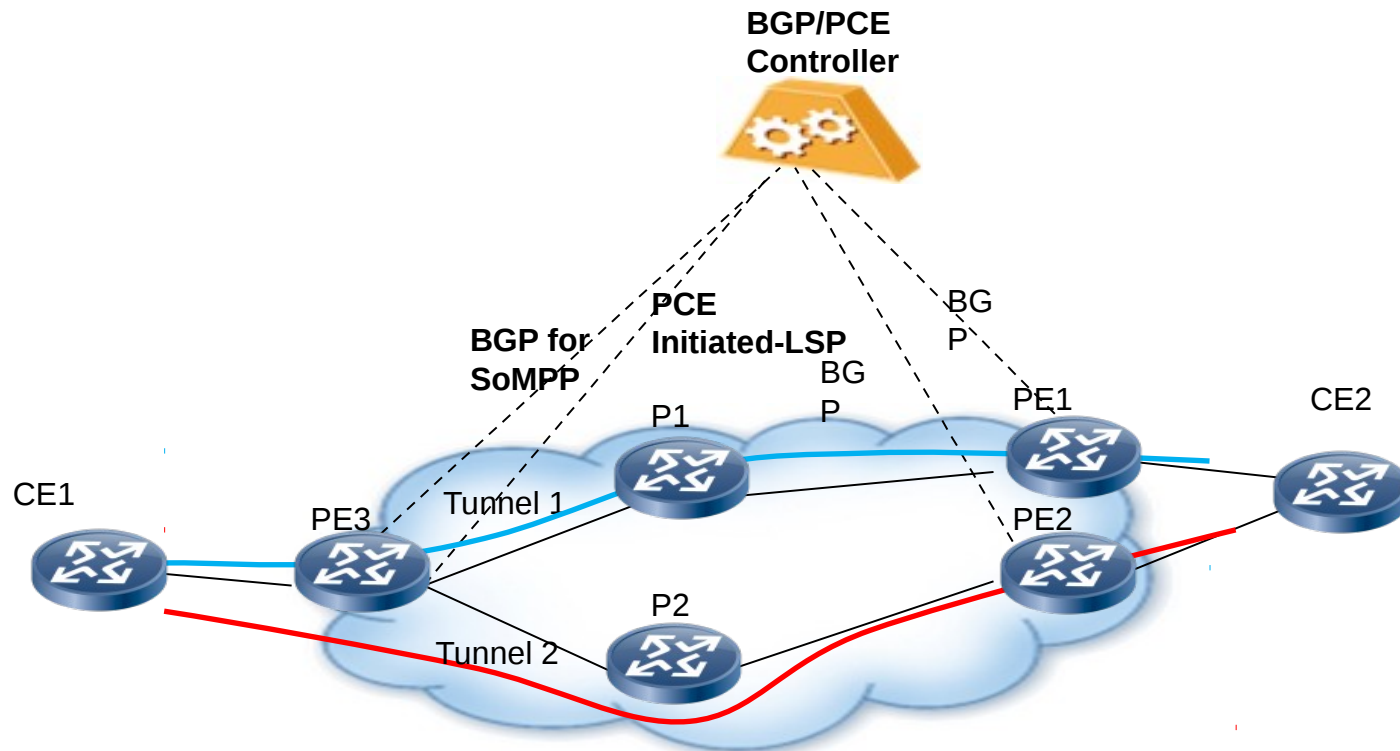
Service-Oriented MPLS Path Programming



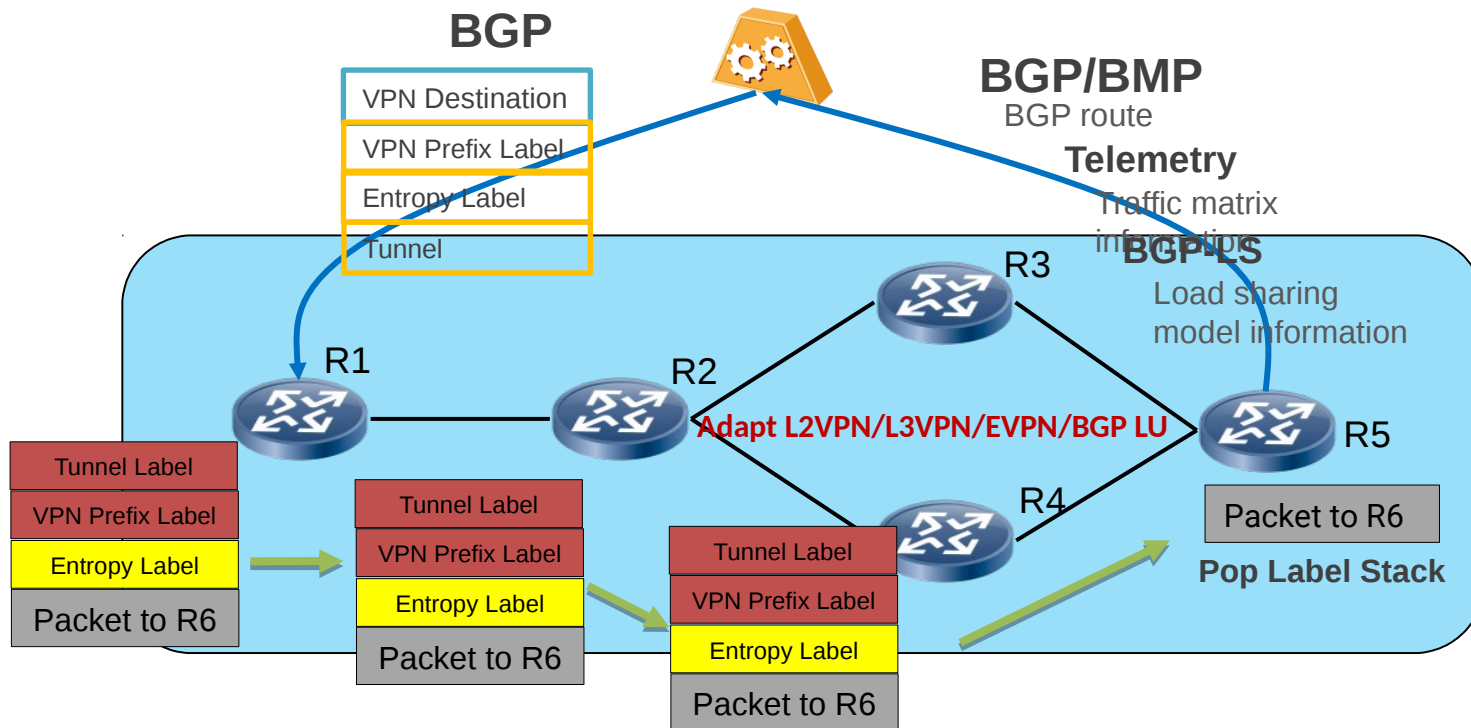
- Network information collection: the controller can collect topology information and Service Label information through BGP-LS and collect routing information through BGP.
- Service Path Programming: the controller programs the path based on the service requirement for a specific flow identified by the BGP prefix.
- Downloading of Service Label Stack: the controller downloads the BGP route with Service label stack information to the specific head-ends.
- Traffic Steering: The head-end created the forwarding entry which can street the traffic and encapsulates the Service label stack to indicate the service process along the forwarding path.

Scenario of SoMPP: Intra-domain Traffic Optimization

- Intra-domain Traffic Optimization:
 - PCE will initiate a new TE tunnel based on the network traffic matrix.
 - The controller downloads the BGP route with the new TE tunnel information to steer the traffic to implement traffic optimization.



Scenario of SoMPP: Determined ECMP with Entropy Label



Benefit:

- The controller can use the traffic matrix information and the load sharing model of each device to calculate the Entropy Label and send it to the head node rather than the head node calculate it on its own. This ensures better ECMP.

Overview of Progress of Related Work of So MPP

- draft-li-spring-mpls-path-programming-00/draft-li-idr-mpls-path-programming are proposed three years ago.
- draft-li-mpls-path-programming-00 is proposed in MPLS WG which is believed the right home of the solution.
- **MPLS Service Label:**
 - Entropy Label/Flow Label
 - SFL: RFC 8372
 - Path Segment: draft-cheng-spring-mpls-path-segment
 - VPN Label: draft-zzhang-bess-mvpn-evpn-aggregation-label-00
 - Requirements & Challenges: Competence of the bottom label
- **MPLS Extension Header: draft-song-mpls-extension-header-00**
 - Requirements & Challenges: Carry the metadata info including iOAM, SFC, etc.
- **BGP Extensions:**
 - BGP for MPLS Path Programming: draft-li-idr-mpls-path-programming
 - BGP Tunnel Encapsulation Attribute: draft-ietf-idr-tunnel-encaps
 - Redirect to Tunnel: draft-ietf-idr-flowspec-path-redirect-06
 - SR Policy: draft-ietf-spring-segment-routing-policy
 - Requirements & Challenges: Carrier combined ID information besides MPLS label stack

Next Step

- Revise the draft to consolidate the solution.
- Revise the draft draft-li-idr-mpls-path-programming to consolidate the BGP extensions.
- Solicit comments and feedbacks
- Seek more cooperation