

MPLS / TE YANG Data Model for Service Provider Networks

draft-openconfig-mpls-consolidated-model-00

IETF 92, 3/27/2015

Joshua George (Google) Luyuan Fang (Microsoft)

Eric Osborne (Level 3) Rob Shakir (BT)

OpenConfig network operator working group www.openconfig.net

OpenConfig: Introduction

- What is OpenConfig?
 - Informal working group of large network operators (including carriers, cable operators, and online service providers).
 - Cross section of a large set of use cases, experiences, and pain points.
- What is the primary goal of OpenConfig?
 - Enable dynamic and programmable network infrastructure for the industry at large.
- What will OpenConfig contribute?
 - Models
 - Documentation
 - Tooling
- Other openconfig efforts: BGP model <https://tools.ietf.org/html/draft-shaikh-idr-bgp-model-00>, policy model <https://tools.ietf.org/html/draft-shaikh-rtgwg-policy-model-00>

Key Ideas

Model Driven Configuration

Declarative, model-driven configuration and management is a Good Thing.

Be Vendor Neutral

Embrace vendor neutrality as much as possible for the data model.

Focus on Use Cases

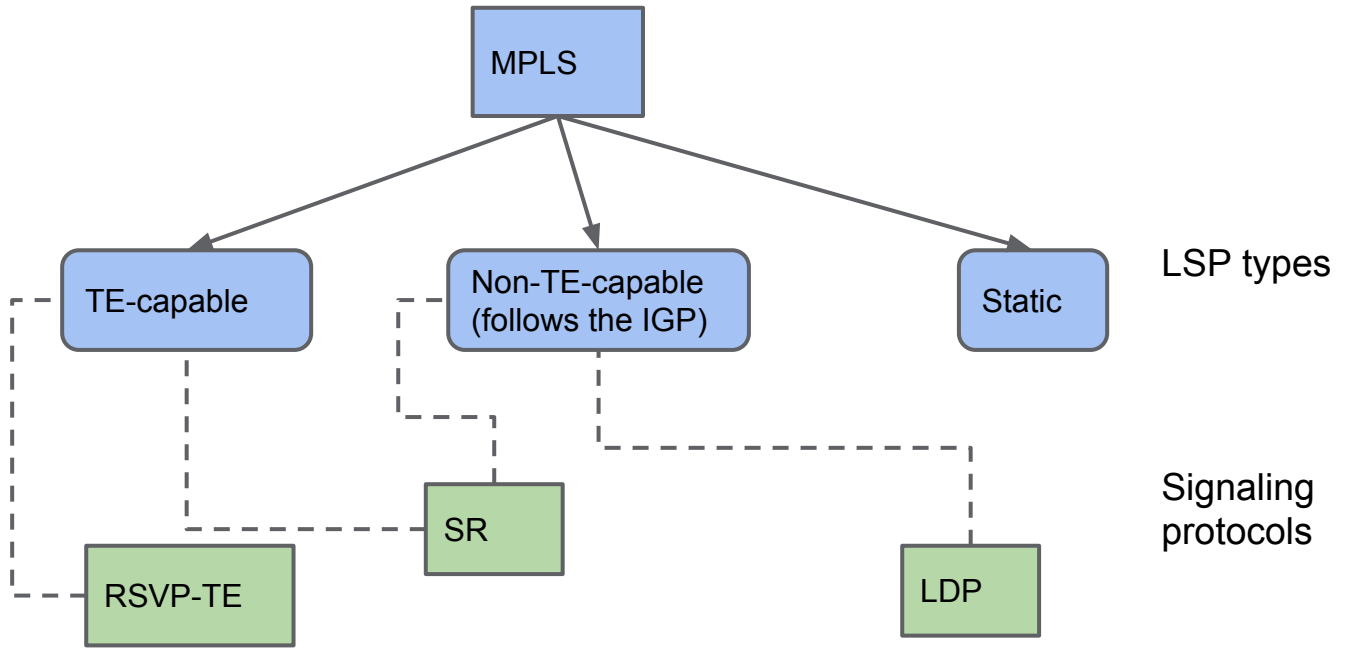
Shape and tune the model through real world use cases - keep it useful, but simple. Deliberately not exhaustive in coverage.

Make Telemetry Possible

Include operational state into the model.

Model coverage

- Covers both label switched path configuration and protocol configuration
- LSP configuration is driven by the type of LSP rather than by the signaling protocol
 - Similar approach to the proposed TE model and its companion the RSVP model
 - Type of LSP is one of: “traffic-engineering capable”, “non-traffic-engineering capable” and “static” (called constrained-path, unconstrained-path and static in the model)
 - Signaling protocols may be RSVP-TE, LDP/mLDP, SR
- Lays out a generic framework for LSPs by type
 - Protocol specific configuration has its place in this framework



Why a data model for MPLS / TE?

- Data model defines the API to the devices
 - Common way to program LSPs of different types on different platforms
 - Enables intent-driven configuration and operation
- Why center the model around the tunnels and not the protocols?
 - Focus on the core function provided, rather than on the mechanics used to achieve it
 - Focus on the items that are most often subject to add/modify/delete provisioning operations - the tunnels themselves
 - Leverage commonalities based on the LSP type

MPLS model overall structure

```
+--rw mpls!  
  +--rw global  
  |   ...  
  +--rw signaling-protocols  
  |   +--rw rsvp  
  |   |   ...  
  |   +--rw segment-routing  
  |   +--rw ldp  
  |   ...  
  +--rw lsps  
    +--rw constrained-path  
    |   ...  
    +--rw unconstrained-path  
    |   ...  
    +--rw static-lsps  
        ...
```

- Three main sections:
 - Global (e.g. specific to the forwarding plane)
 - Signaling-protocol specific
 - Tunnel-specific
- Types of tunnels
 - Traffic-engineering capable (constrained-path)
 - Non-traffic engineering capable (unconstrained-path, follow IGP)
 - Static

Traffic-engineering capable

```
+-rw mpls!  
  +-rw lsps  
    +-rw constrained-path  
      +-rw path-information  
        | +-rw path* [path-name]  
        |     ...  
      +-rw label-switched-path* []  
        +-rw signaled-name  
        +-rw lsp-description?  
        +-rw path-computation-method  
        |     ...  
        +-rw path-attributes  
        |     ...  
        +-rw path-setup  
            ...
```

- TE-capable LSPs
 - Constrained-path - *capable* of instantiating a constrained path
 - Always associated with a path computation method, and with path attributes applicable to TE LSPs
- Path-setup
 - Contains protocol-specific information

Traffic-engineering capable, RSVP-signaled LSPs

RSVP-signaled LSPs

```
+--rw mpls!  
  +--rw lsps  
    +--rw constrained-path  
      +--rw label-switched-path* []  
        +--rw path-setup  
          +--rw rsvp!  
            +--rw path-specification  
            +--rw setup-priority?  
            +--rw hold-priority?  
            +--rw retry-timer?  
            +--rw tunnel  
              +--rw tunnel-type?  
                +--rw p2p-lsp  
                | ...  
                +--rw p2mp-lsp  
                ...
```

- RSVP-specific attributes
 - setup/hold priority
- Tunnel-type-specific configuration (p2p/p2mp)

Summary

- The model provides a structure for protocol configuration separate from tunnel configuration
 - this approach is not alien to how some of the protocols are implemented today
- First cut of the model
- Operational state and RPC will be included in the next version

Next steps

- Add operational state and RPC
- Continue building out the model - in particular LDP and SR
- Publish a new version in the public YangModels repository
<https://github.com/YangModels/yang/tree/master/experimental/openconfig>
- Seek out vendor feedback on implementation readiness