Getting Started with OpenConfig

Santiago Alvarez
OpenConfig introduction

- Operator group pursuing more dynamic and programmable networks
- Specifications designed by operators for operators
- Initial specifications made public mid-2015
- Initial implementations available from several device vendors

Google
at&t
Microsoft
British Telecom
Facebook
Comcast
Verizon
Level 3
Cox Communications

Yahoo!
Apple
Jive Communications
Deutsche Telekom
Bell Canada
SK Telecom
Bloomberg
Netflix
Cloudflare
OpenConfig Major Components

Data

- Config / oper models
  - YANG

Management Protocol

- gRPC Network Management Interface (gNMI)
  - protobuf

Operational RPCs

- gRPC Network Operations Interface (gNOI)
  - protobuf
OpenConfig Data Model Principles

- Modular model definition
- Model structure combines
  - Configuration (intended)
  - Operational data (applied config and derived state)
- Each module subtree declares config and state containers
- Model backward compatibility
  - Driven by use of semantic versioning (xx.yy.zz)
  - Diverges from IETF YANG guidelines (full compatibility)
- String patterns (regex) follow POSIX notation (instead of W3C as defined at IETF)
## OpenConfig Style Conventions

<table>
<thead>
<tr>
<th></th>
<th><strong>Naming</strong></th>
<th><strong>Example</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Module (File) Name</strong></td>
<td>openconfig-&lt;function&gt;(.yang)</td>
<td>openconfig-bgp(.yang)</td>
</tr>
<tr>
<td></td>
<td>openconfig-&lt;function&gt;-&lt;subfunction&gt;(.yang)</td>
<td>openconfig-bgp-policy(.yang)</td>
</tr>
<tr>
<td><strong>Module Prefix Name</strong></td>
<td>oc-&lt;abbreviation&gt;</td>
<td>oc-bgp</td>
</tr>
<tr>
<td></td>
<td>oc-&lt;abbreviation&gt;-&lt;abbreviation&gt;</td>
<td>oc-bgp-pol</td>
</tr>
<tr>
<td><strong>Module Namespace</strong></td>
<td><a href="http://openconfig.net/yang/">http://openconfig.net/yang/</a>&lt;function&gt;</td>
<td><a href="http://openconfig.net/yang/bgp">http://openconfig.net/yang/bgp</a></td>
</tr>
<tr>
<td></td>
<td><a href="http://openconfig.net/yang/">http://openconfig.net/yang/</a>&lt;function&gt;-&lt;subfunction&gt;</td>
<td><a href="http://openconfig.net/yang/bgp-policy">http://openconfig.net/yang/bgp-policy</a></td>
</tr>
<tr>
<td><strong>Enumeration and Identity Identifiers</strong></td>
<td>UPPERCASE_WITH_UNDERSCORES</td>
<td>NOT_PRESENT</td>
</tr>
<tr>
<td><strong>Other Node Identifiers</strong></td>
<td>lowercase-with-dashes</td>
<td>oper-status</td>
</tr>
</tbody>
</table>
## OpenConfig Models

<table>
<thead>
<tr>
<th>Post version 1.0.0</th>
<th>Pre version 1.0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>openconfig-interfaces</td>
<td>openconfig-system</td>
</tr>
<tr>
<td>openconfig-vlan</td>
<td>openconfig-ospfv2</td>
</tr>
<tr>
<td>openconfig-acl</td>
<td>openconfig-network-instance</td>
</tr>
<tr>
<td>openconfig-routing-policy</td>
<td>openconfig-rib-bgp</td>
</tr>
<tr>
<td>openconfig-bgp</td>
<td>openconfig-segment-routing</td>
</tr>
<tr>
<td>openconfig-local-routing</td>
<td>openconfig-openflow</td>
</tr>
<tr>
<td>openconfig-mpls</td>
<td>openconfig-channel-monitor</td>
</tr>
<tr>
<td>openconfig-terminal-device</td>
<td>openconfig-optical-amplifier</td>
</tr>
<tr>
<td>openconfig-transport-line-protection</td>
<td>openconfig-transport-line-protection</td>
</tr>
<tr>
<td>openconfig-wavelength-router</td>
<td>openconfig-wavelength-router</td>
</tr>
</tbody>
</table>

openconfig-isis

openconfig-telemetry

openconfig-lldp

openconfig-policy-forwarding

openconfig-spanning-tree

openconfig-platform

openconfig-module-catalog
OpenConfig BGP: Overview (openconfig-bgp)

• Model for BGP configuration and operational data
• Three top-level containers:
  • Global
  • Neighbors
  • Peer groups
• Multi-protocol support

module: openconfig-bgp
  +--rw bgp!
  |    +--rw global
  |        |    ...
  |    +--rw neighbors
  |        |    ...
  +--rw peer-groups
     ...
OpenConfig BGP: Neighbors and Peer Groups

module: openconfig-bgp
tree-path /bgp/neighbors
  +--rw bgp!
    +--rw neighbors
      +--rw neighbor* [neighbor-address]
        +--rw neighbor-address
        +--rw config
          |  ...
        +--ro state
          |  ...
        +--rw timers
          |  ...
        +--rw transport
          |  ...
        +--rw error-handling
          |  ...
        +--rw logging-options
          |  ...

module: openconfig-bgp
tree-path /bgp/peer-groups
  +--rw bgp!
    +--rw peer-groups
      +--rw peer-group* [peer-group-name]
        +--rw peer-group-name
        +--rw config
          |  ...
        +--ro state
          |  ...
        +--rw timers
          |  ...
        +--rw transport
          |  ...
        +--rw error-handling
          |  ...
        +--rw logging-options
          |  ...

OpenConfig BGP: Neighbors and Peer Groups
OpenConfig Routing Policy: Overview (openconfig-routing-policy)

- Model for routing policy configuration (no oper data)
- Two top-level containers:
  - Defined sets
  - Policy definitions
- Supports policy chains and nested policies
- Each policy has one or more statements with condition-action tuples
- BGP policy augmentations defined in openconfig-bgp-policy

```yaml
module: openconfig-routing-policy
   +--rw routing-policy
      +--rw defined-sets
          | ... 
      +--rw policy-definitions
          ... 
```
OpenConfig Routing Policy: Evaluation Rules

- Policies in a chain are evaluated in order
- Policy statements are evaluated in order
- Actions are executed when conditions are satisfied. Otherwise, next statement is evaluated.
- Policy chain evaluation stops if accept-route or reject-route action is executed
- If no policy statement conditions are satisfied, the next policy definition in the chain is evaluated
- Default route disposition action is performed when the end of the policy chain is reached
OpenConfig Routing Policy: Defined Sets and Policy Definitions

module: openconfig-routing-policy
tree-path /routing-policy/defined-sets
  +--rw routing-policy
     +--rw defined-sets
        +--rw prefix-sets
        |   ...
        +--rw neighbor-sets
        |   ...
        +--rw tag-sets
        ...

module: openconfig-routing-policy
tree-path /routing-policy/policy-definitions
  +--rw routing-policy
     +--rw policy-definitions
        +--rw policy-definition* [name]
        |   +--rw name
        |   +--rw statements
        |      +--rw statement* [name]
        |         |   +--rw name
        |         +--rw conditions
        |            |   +--rw actions
        |            ...
OpenConfig Routing Policy: BGP Augmentations for Defined Sets

module: openconfig-bgp-policy
augment /rpol:routing-policy/rpol:defined-sets:
  +-rw bgp-defined-sets
    +-rw community-sets
      |   +-rw community-set* [community-set-name]
      |   |   +-rw community-set-name
      |   +-rw community-member*
    +-rw ext-community-sets
      |   +-rw ext-community-set* [ext-community-set-name]
      |   |   +-rw ext-community-set-name
      |   +-rw ext-community-member*
    +-rw as-path-sets
      +-rw as-path-set* [as-path-set-name]
        +-rw as-path-set-name
        +-rw as-path-set-member*
OpenConfig Routing Policy: BGP Augmentations for Conditions and Actions

module: openconfig-routing-policy
augment /routing-policy/policy-definitions/policy-definition/statements/statement/conditions:
  +-rw bgp-conditions
    |  +-rw match-community-set!
    |      |  ...  
    |  +-rw match-ext-community-set!
    |      |  ...  
    |  +-rw match-as-path-set!
    |      |  ...  
    |  +-rw med-eq?
    |  +-rw origin-eq?
    |  +-rw next-hop-in*
    |  +-rw afi-safi-in*
    |  +-rw local-pref-eq?
    |  +-rw community-count!
    |      |  ...  
    |  +-rw as-path-length!
    |      |  ...  
    |  +-rw route-type?

module: openconfig-routing-policy
augment /routing-policy/policy-definitions/policy-definition/statements/statement/actions:
  +-rw bgp-actions
    |  +-rw set-as-path-prepend!
    |      |  ...  
    |  +-rw set-community!
    |      |  ...  
    |  +-rw set-ext-community!
    |      |  ...  
    |  +-rw set-route-origin?
    |  +-rw set-local-pref?
    |  +-rw set-next-hop?
    |  +-rw set-med?
OpenConfig Interfaces (openconfig-interfaces)

- Model for interface configuration and operational data
- Four top-level containers under each interface:
  - Config
  - State
  - Hold time
  - subinterfaces
- Four main augmentations:
  - Bundles – openconfig-if-aggregate
  - Ethernet – openconfig-if-Ethernet
  - IP – openconfig-if-ip
  - VLAN – openconfig-vlan
- All protocol configuration resides on subinterfaces
- IP augmentations include static ARP and VRRP

```
module: openconfig-interfaces
  +-rw interfaces
    +-rw interface* [name]
      +-rw name
      +-rw config
        |
        ...  
      +-ro state
        |
        ...  
    +-rw hold-time
        |
        ...  
    +-rw subinterfaces
        ...  
```
OpenConfig Interfaces: Bundle and Ethernet Augmentations

module: openconfig-if-aggregate
augment /ocif:interfaces/ocif:interface:
  +-rw aggregation!
    |   +-rw config
    |       |   +-rw lag-type?
    |       |   +-rw min-links?
    |   +-ro state
    |       |   +-ro lag-type?
    |       |   +-ro min-links?
    |       |   +-ro members*
    +-rw lacp!
      |   +-rw config
      |       |   ...
      |   +-ro state
      |       |   ...
      |   +-ro members

module: openconfig-if-ethernet
augment /ocif:interfaces/ocif:interface:
  +-rw ethernet
    |   +-rw config
    |       |   +-rw mac-address?
    |       |   +-rw auto-negotiate?
    |       |   +-rw duplex-mode?
    |       |   +-rw port-speed?
    |       |   +-rw enable-flow-control?
    |   +-ro state
    |       |   +-ro mac-address?
    |       |   +-ro auto-negotiate?
    |       |   +-ro duplex-mode?
    |       |   +-ro port-speed?
    |       |   +-ro enable-flow-control?
    |   +-ro hw-mac-address?
    |   +-ro counters
    |       |   ...

OpenConfig Interfaces: Bundle and Ethernet Augmentations
OpenConfig Interfaces: IP Augmentations

module: openconfig-if-ip
augment /ocif:interfaces/
ocif:interface/ocif:subinterfaces/
ocif:subinterface:
  +-rw ipv4!
    +-rw address* [ip]
    |  ...
    +-rw neighbor* [ip]
    |  ...
    +-rw config
    |  ...
    +-ro state
    ...

module: openconfig-if-ip
augment /ocif:interfaces/ocif:interface/
ocif:subinterfaces/ocif:subinterface:
  +-rw ipv6!
    +-rw address* [ip]
    |  ...
    +-rw neighbor* [ip]
    |  ...
    +-rw config
    |  ...
    +-ro state
    |  ...
    +-rw autoconf
    ...

OpenConfig Interfaces: VLAN Augmentations

module: openconfig-vlan
augment /ocif:interfaces/ocif:interface/
eth:ethernet:
    +-rw vlan
      +-rw config
      |    +-rw interface-mode?
      |    +-rw native-vlan?
      |    +-rw access-vlan?
      |    +-rw trunk-vlans*
      +-ro state
          +-ro interface-mode?
          +-ro native-vlan?
          +-ro access-vlan?
          +-ro trunk-vlans*

module: openconfig-vlan
augment /ocif:interfaces/ocif:interface/
lag:aggregation:
    +-rw vlan
      +-rw config
      |    +-rw interface-mode?
      |    +-rw native-vlan?
      |    +-rw access-vlan?
      |    +-rw trunk-vlans*
      +-ro state
          +-ro interface-mode?
          +-ro native-vlan?
          +-ro access-vlan?
          +-ro trunk-vlans*
OpenConfig Telemetry (openconfig-telemetry)

- Model for telemetry configuration and operational data
- Three top-level containers:
  - Sensor groups
  - Destination groups
  - Subscriptions

module: openconfig-telemetry
  +++rw telemetry-system
  | +++rw sensor-groups
  | | +++rw sensor-group* [sensor-group-id]
  | | ...
  | +++rw destination-groups
  | | +++rw destination-group* [group-id]
  | | ...
  | +++rw subscriptions
  | | +++rw persistent
  | | ...
  | | +++rw dynamic
  | | ...
Peering Use Case
Configure and Validate Peering on ASBR1

1. Configure routing policy on ASBR1
2. Configure telemetry (interface and BGP subscriptions) on ASBR1
3. Load peer configuration
4. Configure interface and validate operation
5. Configure BGP neighbor and validate operation
## Open Source Tool Chain

<table>
<thead>
<tr>
<th>YDK <code>{鼋}</code> (ydk.io)</th>
<th>Pipeline (git.io/vdnnT)</th>
<th>Kafka (kafka.apache.org)</th>
</tr>
</thead>
<tbody>
<tr>
<td>· Python/C++ bindings for OpenConfig models</td>
<td>· Collector for router streaming telemetry</td>
<td>· Distributed streaming platform (message bus)</td>
</tr>
<tr>
<td>· Detailed client-side data validation</td>
<td>· Performs basic encoding transformation</td>
<td>· Producer, consumer, stream and connector APIs</td>
</tr>
<tr>
<td>· Protocol / transport / encoding abstraction</td>
<td>· Data producer for Kafka, InfluxDB, Prometheus, etc.</td>
<td>· Rich client support (Python, Java, etc)</td>
</tr>
</tbody>
</table>
Configure Routing Policy (Prefix Set)

```python
# configure prefix set
prefix_set = routing_policy.defined_sets.prefix_sets.PrefixSet()
prefix_set.prefix_set_name = "PREFIX-SET"
prefix = prefix_set.Prefix()
prefix.ip_prefix = "10.0.0.0/8"
prefix.masklength_range = "8..32"
prefix_set.prefix.append(prefix)

prefix = prefix_set.Prefix()
prefix.ip_prefix = "172.16.0.0/12"
prefix.masklength_range = "12..32"
prefix_set.prefix.append(prefix)

routing_policy.defined_sets.prefix_sets.prefix_set.append(prefix_set)
```

Match prefixes for 10/8

Match prefixes for 172.16/12
Configure Routing Policy (Statements)

```python
# configure policy definition
policy_definition = routing_policy.policy_definitions.PolicyDefinition()

# configure drop-prefix statement
statement = policy_definition.statements.Statement()
statement.name = "DROP-PREFIXES"
match_prefix_set = statement.conditions.MatchPrefixSet()
match_prefix_set.prefix_set = "PREFIX-SET"
match_set_options = oc_policy_types.MatchSetOptionsRestrictedTypeEnum.ANY
match_prefix_set.match_set_options = match_set_options
statement.conditions.match_prefix_set = match_prefix_set
statement.actions.reject_route = Empty()
policy_definition.statements.statement.append(statement)

# configure accept statement
statement = policy_definition.statements.Statement()
statement.name = "ACCEPT-ROUTE"
statement.actions.accept_route = Empty()
policy_definition.statements.statement.append(statement)
routing_policy.policy_definitions.policy_definition.append(policy_definition)
```

Drop prefixes matching ANY prefix in PREFIX-SET
Accept all other prefixes
Configure Interface and BGP Telemetry (Sensors)

```
# configure sensor group
sensor_group = telemetry_system.sensor_groups.SensorGroup()
sensor_group.sensor_group_id = "PEERING-SENSORS"
sensor_group.config.sensor_group_id = "PEERING-SENSORS"

# configure interface sensor path
sensor_path = sensor_group.sensor_paths.SensorPath()
sensor_path.path = "openconfig/interfaces/interfaces/interface"
sensor_group.sensor_paths.sensor_path.append(sensor_path)

# configure bgp-neighbor sensor path
sensor_path = sensor_group.sensor_paths.SensorPath()
sensor_path.path = "openconfig-bgp:bgp/neighbors"
sensor_group.sensor_paths.sensor_path.append(sensor_path)
telemetry_system.sensor_groups.sensor_group.append(sensor_group)
```

Stream interface data

Stream BGP data
Configure Interface and BGP Telemetry (Destination)

```python
# configure destination group
destination_group = telemetry_system.destination_groups.DestinationGroup()
destination_group.group_id = "PIPELINE"
destination_group.config.group_id = "PIPELINE"

# configure destination
destination = destination_group.destinations.Destination()
destination.destination_address = "10.16.10.129"
destination.destination_port = 5432
destination.config.destination_address = "10.16.10.129"
destination.config.destination_port = 5432
destination.config.destination_protocol = oc_telemetry.TelemetryStreamProtocolEnum.TCP
destination_group.destinations.destination.append(destination)

telemetry_system.destination_groups.destination_group.append(destination_group)
```

Add Pipeline host to destination group
Configure Interface and BGP Telemetry (Subscription)

```python
# configure subscription
subscription = telemetry_system.subscriptions.persistent.Subscription()
subscription.subscription_id = 10
subscription.config.subscription_id = 10
sensor_profile = subscription.sensor_profiles.SensorProfile()
sensor_profile.sensor_group = "PEERING-SENSORS"
sensor_profile.config.sensor_group = "PEERING-SENSORS"
sensor_profile.config.sample_interval = 10000
subscription.sensor_profiles.sensor_profile.append(sensor_profile)
destination_group = subscription.destination_groups.DestinationGroup()
destination_group.group_id = "PIPELINE"
destination_group.config.group_id = "PIPELINE"
subscription.destination_groups.destination_group.append(destination_group)
telemetry_system.subscriptions.persistent.subscription.append(subscription)
```

Specify a subscription (sensor + destination groups)
Peering Use Case
Peer Configuration

```json
{
  "asbr": {
    "name": "asbr1",
    "address": "198.18.1.11"
  },
  "as": 65001,
  "interface": {
    "name": "GigabitEthernet0/0/0/0",
    "description": "Peering with AS65002",
    "address": "192.168.0.1",
    "netmask": 24
  },
  "neighbor": {
    "address": "192.168.0.2",
    "as": 65002
  }
}
```
Configure Interface (Including IP Subinterface)

```python
# configure interface
interface = interfaces.Interface()
interface.name = "GigabitEthernet0/0/0/0"
interface.config.name = "GigabitEthernet0/0/0/0"
interface.config.description = "Peering with AS65002"
interface.config.enabled = True

# configure ip subinterface
subinterface = interface.subinterfaces.Subinterface()
subinterface.index = 0
subinterface.config.index = 0
subinterface.ipv4 = subinterface.Ipv4()
address = subinterface.ipv4.Address()
address.ip = "192.168.0.1"
address.config.ip = "192.168.0.1"
address.config.prefix_length = 24
subinterface.ipv4.address.append(address)
interface.subinterfaces.subinterface.append(subinterface)
interfaces.interface.append(interface)
```
Interface State Validation

Model: openconfig.interfaces

Path: interfaces/interface/subinterfaces/subinterface/state/oper-status

- **UP** - Ready to pass packets.
- **DOWN** - The interface does not pass any packets.
- **TESTING** - In some test mode. No operational packets can be passed.
- **UNKNOWN** - Status cannot be determined for some reason.
- **DORMANT** - Waiting for some external event.
- **NOT_PRESENT** - Some component (typically hardware) is missing.
- **LOWER_LAYER_DOWN** - Down due to state of lower-layer interface(s).
Configure Global BGP and Peer Group

# configure global bgp
bgp.global_.config.as_ = 65001
afi_safi = bgp.global_.afi_safis.AfiSafi()
afi_safi.afi_safi_name = oc_bgp_types.Ipv4UnicastIdentity()
afi_safi.config.afi_safi_name = oc_bgp_types.Ipv4UnicastIdentity()
afi_safi.config.enabled = True
bgp.global_.afi_safis.afi_safi.append(afi_safi)

# configure EBGP peer group
peer_group = bgp.peer_groups.PeerGroup()
peer_group.peer_group_name = "EBGP"
peer_group.config.peer_group_name = "EBGP"
afi_safi = peer_group.afi_safis.AfiSafi()
afi_safi.afi_safi_name = oc_bgp_types.Ipv4UnicastIdentity()
afi_safi.config.afi_safi_name = oc_bgp_types.Ipv4UnicastIdentity()
afi_safi.config.enabled = True
afi_safi.apply_policy.config.import_policy.append("ROUTE-POLICY")
afi_safi.apply_policy.config.export_policy.append("ROUTE-POLICY")
peer_group.afi_safis.afi_safi.append(afi_safi)
bgp.peer_groups.peer_group.append(peer_group)
Configure BGP Neighbor

# configure EBGP neighbor
neighbor = bgp.neighbors.Neighbor()
neighbor.neighbor_address = "192.168.0.2"
neighbor.config.neighbor_address = "192.168.0.2"
neighbor.config.peer_as = 65001
neighbor.config.peer_group = "EBGP"
bgp.neighbors.neighbor.append(neighbor)
BGP Neighbor State Validation

Model: openconfig-bgp

Path: openconfig-bgp:bgp/neighbors/neighbor/state/session-state

- IDLE - neighbor is down, and in the Idle state of the FSM
- CONNECT - neighbor is down, and the session is waiting for the underlying transport session to be established
- ACTIVE - neighbor is down, and the local system is awaiting a connection from the remote peer
- OPENSENT - neighbor is in the process of being established. The local system has sent an OPEN message.
- OPENCONFIRM - neighbor is in the process of being established. The local system is awaiting a NOTIFICATION or KEEPALIVE message.
- ESTABLISHED - neighbor is up - the BGP session with the peer is established.
Peering Use Case Execution

Loading peer configuration .... [ OK ]
Configure interface .......... [ OK ]
Configure BGP ................. [ OK ]
## Demo Components

**Vagrant Box at** [https://github.com/CiscoDevNet/ydk-py-samples](https://github.com/CiscoDevNet/ydk-py-samples)

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>openconfig-routing-policy</td>
<td>1.1.0</td>
</tr>
<tr>
<td>openconfig-telemetry</td>
<td>0.2.0</td>
</tr>
<tr>
<td>openconfig-interfaces</td>
<td>0.2.0</td>
</tr>
<tr>
<td>openconfig-bgp</td>
<td>1.1.0</td>
</tr>
<tr>
<td>YDK</td>
<td>0.5.5</td>
</tr>
<tr>
<td>Pipeline</td>
<td>1.0.0</td>
</tr>
<tr>
<td>Kafka</td>
<td>0.10.0.0</td>
</tr>
<tr>
<td>zookeeper</td>
<td>3.4.6</td>
</tr>
</tbody>
</table>
Summary

• OpenConfig provides a growing number of vendor-neutral data models

• Available implementations support real use cases

• Data models and streaming telemetry simplify automation of operational workflows
Resources

- OpenConfig Site
  http://www.openconfig.net/

- Vagrant box with clients and telemetry collector
  https://github.com/CiscoDevNet/ydk-py-samples

- Getting Started With OpenConfig
  https://github.com/CiscoDevNet/openconfig-getting-started
Backup
Testbed Topology

Controller

Out-of-band network

IS-IS L2 (Area 49.0001)

R1

R2

NETCONF

198.18.1.1127/24

NETCONF

198.18.1.12/24

198.18.1.12/24

198.18.1.11/24

NETCONF

lo0

172.16.255.1/32

g0/0/0/0

172.16.1.0/31

172.16.1.1/31

172.16.1.1/31

172.16.255.2/32

lo0

172.16.255.1/32

172.16.1.0/31

172.16.1.1/31

172.16.255.2/32
OpenConfig VLAN (openconfig-vlan)

- Model for VLAN configuration and operational data
- Two top-level containers under each VLAN:
  - Config
  - State
- Augments openconfig-interfaces

module: openconfig-vlan
  +-rw vlans
    +-rw vlan* [vlan-id]
      +-rw vlan-id
      +-rw config
      |  +-rw vlan-id?
      |  +-rw name?
      |  +-rw status?
      +-ro state
      |  +-ro vlan-id?
      |  +-ro name?
      |  +-ro status?
      |  +-ro member-ports*
OpenConfig MPLS (openconfig-mpls)

- Model for MPLS related configuration and operational data
- Includes data for:
  - Static MPLS
  - LDP
  - Traffic Engineering
  - RSVP
  - Segment Routing
- Five top-level containers under each interface:
  - global
  - TE global attributes
  - TE interface attributes
  - Signaling protocols
  - LSPs
- Only supports named LSP and named explicit paths

```
module: openconfig-mpls
  +++-rw mpls!
  |       +++-rw global
  |       |       ...|
  |       +---rw te-global-attributes
  |       |       ...|
  |       +---rw te-interface-attributes
  |       |       ...|
  |       +---rw signaling-protocols
  |       |       ...|
  |       +---rw lsps
  |       ... |
```
OpenConfig MPLS: Global and TE Global Attributes

module: openconfig-mpls
tree-path /mpls/global
  +--rw mpls!
    +--rw global
      +--rw config
      |    +--rw null-label?
      +--ro state
      |    +--ro null-label?
    +--rw mpls-interface-attributes
      +--rw interface* [name]
        +--rw name
      +--rw config
      |    +--rw name?
      |    +--rw mpls-enabled?
      +--ro state
        +--ro name?
        +--ro mpls-enabled?

module: openconfig-mpls
tree-path /mpls/te-global-attributes
  +--rw mpls!
    +--rw te-global-attributes
      +--rw srlgs
      |    +--rw srlg* [name]
      |    ...
      +--rw igp-flooding-bandwidth
      |    +--rw config
      |    |    +--ro state
      |    |    ...
    +--rw mpls-admin-groups
      |    +--rw admin-group* [admin-group-name]
      |    ...
    +--rw te-lsp-timers
      +--rw config
      |    ...
      +--ro state
      ...
# OpenConfig MPLS: TE Interface Attributes and Signaling Protocols

**module**: openconfig-mpls  
**tree-path**: /mpls/te-interface-attributes  
+--rw mpls!  
   +--rw te-interface-attributes  
   |   +--rw interface* [name]  
   |   |   +--rw name  
   |   +--rw config  
   |   |   +--ro state  
   |   |   +--rw igp-flooding-bandwidth  
   |   ...  
   +--ro state  
   +--ro state  
   ...  

**module**: openconfig-mpls  
**tree-path**: /mpls/signaling-protocols  
+--rw mpls!  
   +--rw signaling-protocols  
   |   +--rw rsvp-te  
   |   |   +--rw sessions  
   |   |   +--rw neighbors  
   |   |   +--rw global  
   |   |   +--rw interface-attributes  
   |   |   +--rw segment-routing  
   |   |   +--rw srgbs  
   |   |   +--rw interfaces  
   |   |   +--rw ldp  
   |   |   +--rw timers  
   |   ...  
   ...
module: openconfig-mpls

tree-path /mpls/lsp
  +--rw mpls!
    +--rw lsp
        +--rw constrained-path
            | +--rw named-explicit-paths
            |     +--rw named-explicit-path* [name]
            |     ...
            +--rw tunnels
            |     +--rw tunnel* [name source]
            |     ...
        +--rw unconstrained-path
            |     +--rw path-setup-protocol
            |         +--rw ldp!
            |         ...
            | +--rw segment-routing
            |     ...
        +--rw static-lsp