Enhancements to Data Models

This section provides an overview of the enhancements made to data models.

OpenConfig Data Model Enhancements

Table 1: Feature History Table

Feature Name	Release Information	Description
LACP OpenConfig Model	Release 7.5.3	Use the `openconfig-lacp.yang` data model to manage Link Aggregation Control Protocol (LACP) aggregate interfaces by monitoring the number of LACP timeouts and the time since the last timeout. With this release, the data model is revised from version 1.1.0 to 1.2.0 to introduce the following sensor paths for the operational state of the bundle member interface lacp/interfaces/interface[name]/members/member[interface]/state/: • `last-change` • `counters/lacp-timeout-transitions` You can stream Event-driven telemetry data for the time since the last change of a timeout, and Model-driven telemetry data for the number of times the state has transitioned with a timeout. The state change is monitored since the time the device restarted or the interface was brought up, whichever is most recent.

Install Label in oc-platform Data Model

The openconfig-platform (oc-platform.yang) data model is enhanced to provide the following data:

- IOS XR software version (optionally with GISO label)
- Type, description, operational status of the component. For example, a CPU component reports its utilization, temperature or other physical properties.
- List of the committed IOS XR packages

To retrieve oc-platform information from a router via NETCONF, ensure you configured the router with the SH server and management interface:

```
Router#show run
Building configuration...
!! IOS XR Configuration version = 7.3.2
!! Last configuration change at Tue Sep  7 16:18:14 2016 by USER1
!
......
......
netconf-yang agent ssh
ssh server netconf vrf default
interface MgmtEth 0/RP0/CPU0/0
    no shut
    ipv4 address dhcp
```

The following example shows the enhanced `OPERATING_SYSTEM` node component (line card or route processor) of the oc-platform data model:

```
<component>
<name>IOSXR-NODE 0/RP0/CPU0</name>
<config>
<name>0/RP0/CPU0</name>
</config>
<state>
<name>0/RP0/CPU0</name>
<type xmlns:idx="http://openconfig.net/yang/platform-types">idx:OPERATING_SYSTEM</type>
<location>0/RP0/CPU0</location>
<description>IOS XR Operating System</description>
<software-version>7.3.2</software-version> -----------------------> Label Info
<removable>true</removable>
<oper-status xmlns:idx="http://openconfig.net/yang/platform-types">idx:ACTIVE</oper-status>
</state>
<subcomponents>
 <subcomponent>
  <name><platform>-af-ea-7.3.2v1.0.0.1</name>
  <config>
   <name><platform>-af-ea-7.3.2v1.0.0.1</name>
  </config>
  <state>
   <name><platform>-af-ea-7.3.2v1.0.0.1</name>
  </state>
 </subcomponent>
...
```

The following example shows the enhanced `OPERATING_SYSTEM_UPDATE` package component (RPMs) of the oc-platform data model:

```
<component>
<name>IOSXR-PKG/1 <platform>-isis-2.1.0.0-r732</name>
```

```
<config>
<name><platform>-isis-2.1.0.0-r732</name>
</config>
<state>
<name><platform>-isis-2.1.0.0-r732</name>
<type xmlns:idx="http://openconfig.net/yang/platform-types">idx:OPERATING_SYSTEM_UPDATE</type>
<description>IOS XR Operating System Update</description>
<software-version>7.3.2</software-version>-----------------------> Label Info
<removable>true</removable>
<oper-status xmlns:idx="http://openconfig.net/yang/platform-types">idx:ACTIVE</oper-status>
</state>
</component>
```

Associated Commands

- **show install committed**—Shows the committed IOS XR packages.
- **show install committed summary**—Shows a summary of the committed packages along with the committed IOS XR version that is displayed as a label.

OAM for MPLS and SR-MPLS in mpls-ping and mpls-traceroute Data Models

Table 2: Feature History Table

Feature Name	Release Information	Description
YANG Data Models for MPLS OAM RPCs	Release 7.3.2	This feature introduces the `Cisco-IOS-XR-mpls-ping-act` and `Cisco-IOS-XR-mpls-traceroute-act` YANG data models to accommodate operations, administration and maintenance (OAM) RPCs for MPLS and SR-MPLS. You can access these Cisco IOS XR native data models from the Github repository.

The Cisco-IOS-XR-mpls-ping-act and Cisco-IOS-XR-mpls-traceroute-act YANG data models are introduced to provide the following options:

- Ping for MPLS:
  - MPLS IPv4 address
  - MPLS TE
  - FEC-129 Pseudowire
  - FEC-128 Pseudowire
  - Multisegment Pseudowire

- Ping for SR-MPLS:
    - SR policy name or BSID with LSP end-point
    - SR MPLS IPv4 address
    - SR Nil-FEC labels
    - SR Flexible Algorithm
- Traceroute for MPLS:
    - MPLS IPv4 address
    - MPLS TE
- Traceroute for SR-MPLS:
    - SR policy name or BSID with LSP end-point
    - SR MPLS IPv4 address
    - SR Nil-FEC labels
    - SR Flexible Algorithm

The following example shows the ping operation for an SR policy and LSP end-point:

```
<mpls-ping xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-mpls-ping-act">
  <sr-mpls>
  <policy>
    <name>srte_c_10_ep_10.10.10.1</name>
      <lsp-endpoint>10.10.10.4</lsp-endpoint>
  </policy>
  </sr-mpls>
  <request-options-parameters>
    <brief>true</brief>
  </request-options-parameters>
</mpls-ping>
```

Response:

```
<?xml version="1.0"?>
 <mpls-ping-response xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-mpls-ping-act">
  <request-options-parameters>
   <exp>0</exp>
   <fec>false</fec>
   <interval>0</interval>
   <ddmap>false</ddmap>
   <force-explicit-null>false</force-explicit-null>
   <packet-output>
    <interface-name>None</interface-name>
    <next-hop>0.0.0.0</next-hop>
   </packet-output>
   <pad>abcd</pad>
   <repeat>5</repeat>
   <reply>
    <dscp>255</dscp>
    <reply-mode>default</reply-mode>
    <pad-tlv>false</pad-tlv>
   </reply>
   <size>100</size>
  <source>0.0.0.0</source>
  <destination>127.0.0.1</destination>
  <sweep>
   <minimum>100</minimum>
   <maximum>100</maximum>
   <increment>1</increment>
  </sweep>
  <brief>true</brief>
  <timeout>2</timeout>
  <ttl>255</ttl>
 </request-options-parameters>
 <replies>
  <reply>
   <reply-index>1</reply-index>
   <return-code>3</return-code>
   <return-char>!</return-char>
   <reply-addr>14.14.14.3</reply-addr>
   <size>100</size>
  </reply>
  <reply>
   <reply-index>2</reply-index>
   <return-code>3</return-code>
   <return-char>!</return-char>
   <reply-addr>14.14.14.3</reply-addr>
   <size>100</size>
  </reply>
  <reply>
   <reply-index>3</reply-index>
   <return-code>3</return-code>
   <return-char>!</return-char>
   <reply-addr>14.14.14.3</reply-addr>
   <size>100</size>
  </reply>
  <reply>
   <reply-index>4</reply-index>
   <return-code>3</return-code>
   <return-char>!</return-char>
   <reply-addr>14.14.14.3</reply-addr>
   <size>100</size>
  </reply>
  <reply>
   <reply-index>5</reply-index>
   <return-code>3</return-code>
   <return-char>!</return-char>
   <reply-addr>14.14.14.3</reply-addr>
   <size>100</size>
  </reply>
 </replies>
</mpls-ping-response>
```

The following example shows the ping operation for an SR policy BSID and LSP end-point:

```
<mpls-ping xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-mpls-ping-act">
<sr-mpls>
<policy>
    <bsid>1000</bsid>
    <lsp-endpoint>10.10.10.4</lsp-endpoint>
</policy>
</sr-mpls>
<request-options-parameters>
    <brief>true</brief>
</request-options-parameters>
</mpls-ping>
```

Response:

```
<?xml version="1.0"?>
 <mpls-ping-response xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-mpls-ping-act">
  <request-options-parameters>
   <exp>0</exp>
   <fec>false</fec>
   <interval>0</interval>
   <ddmap>false</ddmap>
   <force-explicit-null>false</force-explicit-null>
   <packet-output>
    <interface-name>None</interface-name>
    <next-hop>0.0.0.0</next-hop>
   </packet-output>
   <pad>abcd</pad>
   <repeat>5</repeat>
   <reply>
    <dscp>255</dscp>
    <reply-mode>default</reply-mode>
    <pad-tlv>false</pad-tlv>
   </reply>
   <size>100</size>
   <source>0.0.0.0</source>
   <destination>127.0.0.1</destination>
   <sweep>
    <minimum>100</minimum>
    <maximum>100</maximum>
    <increment>1</increment>
   </sweep>
   <brief>true</brief>
   <timeout>2</timeout>
   <ttl>255</ttl>
  </request-options-parameters>
  <replies>
   <reply>
    <reply-index>1</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
   <reply>
    <reply-index>2</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
   <reply>
    <reply-index>3</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
   <reply>
    <reply-index>4</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
   <reply>
    <reply-index>5</reply-index>
    <return-code>3</return-code>
    <return-char>!</return-char>
    <reply-addr>14.14.14.3</reply-addr>
    <size>100</size>
   </reply>
  </replies>
 </mpls-ping-response>
```

The following example shows the traceroute operation for an SR policy and LSP end-point:

```
<mpls-traceroute xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-mpls-traceroute-act">
<sr-mpls>
<policy>
    <name>srte_c_10_ep_10.10.10.1</name>
    <lsp-endpoint>10.10.10.4</lsp-endpoint>
</policy>
</sr-mpls>
<request-options-parameters>
    <brief>true</brief>
</request-options-parameters>
</mpls-traceroute>
```

Response:

```
<?xml version="1.0"?>
 <mpls-traceroute-response xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-mpls-traceroute-act">

  <request-options-parameters>
   <exp>0</exp>
   <fec>false</fec>
   <ddmap>false</ddmap>
   <force-explicit-null>false</force-explicit-null>
   <packet-output>
    <interface-name>None</interface-name>
    <next-hop>0.0.0.0</next-hop>
   </packet-output>
   <reply>
    <dscp>255</dscp>
    <reply-mode>default</reply-mode>
   </reply>
   <source>0.0.0.0</source>
   <destination>127.0.0.1</destination>
   <brief>true</brief>
   <timeout>2</timeout>
   <ttl>30</ttl>
  </request-options-parameters>
  <paths>
   <path>
    <path-index>0</path-index>
    <hops>
    <hop>
        <hop-index>0</hop-index>
        <hop-origin-ip>11.11.11.1</hop-origin-ip>
        <hop-destination-ip>11.11.11.2</hop-destination-ip>
        <mtu>1500</mtu>
        <dsmap-label-stack>
            <dsmap-label>
                <label>16003</label>
            </dsmap-label>
        </dsmap-label-stack>
        <return-code>0</return-code>
        <return-char> </return-char>
    </hop>
  <hop>
      <hop-index>1</hop-index>
      <hop-origin-ip>11.11.11.2</hop-origin-ip>
      <hop-destination-ip>14.14.14.3</hop-destination-ip>
      <mtu>1500</mtu>
      <dsmap-label-stack>
          <dsmap-label>
              <label>3</label>
          </dsmap-label>
      </dsmap-label-stack>
      <return-code>8</return-code>
      <return-char>L</return-char>
  </hop>
  <hop>
      <hop-index>2</hop-index>
      <hop-origin-ip>14.14.14.3</hop-origin-ip>
      <hop-destination-ip></hop-destination-ip>
      <mtu>0</mtu>
      <dsmap-label-stack/>
      <return-code>3</return-code>
      <return-char>!</return-char>
  </hop>
  </hops>
 </path>
</paths>
</mpls-traceroute-response>
```

OpenConfig YANG Model:MACsec

Table 3: Feature History Table

Feature Name	Release Information	Description
OpenConfig YANG Model:MACsec	Release 7.5.2	You can now use the OpenConfig YANG data model to define the MACsec key chain and policy, and apply MACsec encryption on a router interface. You can access the OC data model from the Github repository.

With the OpenConfig YANG Model:MACsec, you can also retrieve operational data from the NETCONF agent using gRPC. By automating processes that are repeated across multiple network elements, you can leverage the YANG models for MACsec.

You can use the following operations to stream Telemetry data by sending a request to the NETCONF agent:

- <get>
- <get-config>
- <edit-config>

Subscribe to the following sensor paths to send a pull request to the YANG leaf, list, or container:

- mka/key-chains/key-chain/mka-keys/mka-key

- interfaces/interface/mka
- interfaces/interface
- mka/policies/policy
- interfaces/interface/scsa-rx/scsa-rx
- interfaces/interface/scsa-tx/scsa-tx

Limitation

- The current implementation of Cisco IOS XR supports only the local time zone configuration in the YYYY-MM-DDTHH:MM:SS format for the following paths:
  - /macsec/mka/key-chains/key-chain/mka-keys/mka-key/config/valid-date-time
  - /macsec/mka/key-chains/key-chain/mka-keys/mka-key/config/expiration-date-time
  - /macsec/mka/key-chains/key-chain/mka-keys/mka-key/state/valid-date-time
  - /macsec/mka/key-chains/key-chain/mka-keys/mka-key/state/expiration-date-time
- Under the MACsec policy, you can disable the delay-protection and include-icv-indicator leaves only by using the delete operation. You cannot modify the configuration by updating the default field value, from true to false. This codeblock shows a sample delete operation:

  ```
  <config>
  <delay-protection nc:operation="delete"/>
  <include-icv-indicator nc:operation="delete"/>
  </config>
  ```

Running Configuration

```
RP/0/0/CPU0:ios#show running-config
Tue Apr 19 21:36:08.882 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Thu Apr 14 16:25:17 2022 by UNKNOWN
key chain kc
 macsec
  key 1234
   key-string password
00554155500E5D5157701E1D5D4C53404A5A5E577E7E727F6B647040534355560E080A00005B554F4E080A0407070303530A54540C0252445E550958525A771B16
 cryptographic-algorithm aes-256-cmac
   lifetime 00:01:01 january 01 2021 infinite
  netconf-yang agent
 ssh
interface GigabitEthernet0/0/0/0
 shutdown
interface GigabitEthernet0/0/0/1
 macsec psk-keychain kc
interface GigabitEthernet0/0/0/2
 macsec psk-keychain kc policy mp
interface GigabitEthernet0/0/0/3
 shutdown
interface GigabitEthernet0/0/0/4
 shutdown
macsec-policy mp
 cipher-suite GCM-AES-XPN-256
 key-server-priority 4
```

```
ssh server v2
end
```

RPC Request for get-config

```
<get-config>
   <source>
      <running/>
   </source>
        <filter>
          <macsec xmlns="http://openconfig.net/yang/macsec">
          </macsec>
        </filter>
</get-config>
```

RPC Response for get-config

```
<?xml version="1.0"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 <data>
  <macsec xmlns="http://openconfig.net/yang/macsec">
   <mka>
    <policies>
     <policy>
      <name>mp</name>
      <config>
       <name>mp</name>
       <macsec-cipher-suite>gcm-aes-xpn-256</macsec-cipher-suite>
       <key-server-priority>4</key-server-priority>
      </config>
     </policy>
    </policies>
    <key-chains>
     <key-chain>
      <name>kc</name>
      <config>
       <name>kc</name>
      </config>
      <mka-keys>
       <mka-key>
        <id>1234</id>
        <config>
         <id>1234</id>
         <cryptographic-algorithm>AES_256_CMAC</cryptographic-algorithm>
         <valid-date-time>2021-01-01T00:01:01</valid-date-time>
         <expiration-date-time>NO_EXPIRATION</expiration-date-time>
        </config>
       </mka-key>
      </mka-keys>
     </key-chain>
    </key-chains>
   </mka>
   <interfaces>
    <interface>
     <name>GigabitEthernet0/0/0/1</name>
     <config>
      <name>GigabitEthernet0/0/0/1</name>
     </config>
     <mka>
      <config>
       <key-chain>kc</key-chain>
      </config>
     </mka>
    </interface>
    <interface>
     <name>GigabitEthernet0/0/0/2</name>
     <config>
      <name>GigabitEthernet0/0/0/2</name>
     </config>
     <mka>
      <config>
       <key-chain>kc</key-chain>
       <mka-policy>mp</mka-policy>
      </config>
     </mka>
    </interface>
   </interfaces>
  </macsec>
 </data>
</rpc-reply>
```

RPC Request for get

```
<get>
   <filter>
     <macsec xmlns="http://openconfig.net/yang/macsec">
     </macsec>
     </filter>
</get>
```

RPC Response for get

```
<?xml version="1.0"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 <data>
  <macsec xmlns="http://openconfig.net/yang/macsec">
   <mka>
    <policies>
     <policy>
      <name>mp</name>
      <config>
       <name>mp</name>
       <macsec-cipher-suite>gcm-aes-xpn-256</macsec-cipher-suite>
       <key-server-priority>4</key-server-priority>
      </config>
      <state>
       <name>mp</name>
       <key-server-priority>4</key-server-priority>
       <macsec-cipher-suite>gcm-aes-xpn256</macsec-cipher-suite>
       <confidentiality-offset>zero-bytes</confidentiality-offset>
       <delay-protection>false</delay-protection>
       <include-icv-indicator>false</include-icv-indicator>
       <sak-rekey-interval>0</sak-rekey-interval>
      </state>
     </policy>
     <policy>
      <name>DEFAULT-POLICY</name>
      <state>
       <name>DEFAULT-POLICY</name>
       <key-server-priority>16</key-server-priority>
       <macsec-cipher-suite>gcm-aes-xpn256</macsec-cipher-suite>
       <confidentiality-offset>zero-bytes</confidentiality-offset>
       <delay-protection>false</delay-protection>
       <include-icv-indicator>false</include-icv-indicator>
       <sak-rekey-interval>0</sak-rekey-interval>
      </state>
  </policy>
 </policies>
 <key-chains>
  <key-chain>
   <name>kc</name>
   <config>
    <name>kc</name>
   </config>
   <mka-keys>
    <mka-key>
     <id>1234</id>
     <config>
      <id>1234</id>
      <cryptographic-algorithm>AES_256_CMAC</cryptographic-algorithm>
      <valid-date-time>2021-01-01T00:01:01</valid-date-time>
      <expiration-date-time>NO_EXPIRATION</expiration-date-time>
     </config>
     <state>
      <id>1234</id>
      <cryptographic-algorithm>AES_256_CMAC</cryptographic-algorithm>
      <valid-date-time>2021-01-01T00:01:01</valid-date-time>
      <expiration-date-time>NO_EXPIRATION</expiration-date-time>
     </state>
    </mka-key>
   </mka-keys>
   <state>
    <name>kc</name>
   </state>
  </key-chain>
 </key-chains>
</mka>
<interfaces>
 <interface>
  <name>GigabitEthernet0_0_0_1</name>
  <state>
   <name>GigabitEthernet0_0_0_1</name>
   <counters>
    <tx-untagged-pkts>8</tx-untagged-pkts>
    <rx-untagged-pkts>0</rx-untagged-pkts>
    <rx-badtag-pkts>2</rx-badtag-pkts>
    <rx-unknownsci-pkts>3</rx-unknownsci-pkts>
    <rx-nosci-pkts>4</rx-nosci-pkts>
   </counters>
  </state>
  <mka>
   <state>
    <mka-policy>DEFAULT-POLICY</mka-policy>
    <key-chain>kc</key-chain>
    <counters>
     <in-mkpdu>0</in-mkpdu>
     <in-sak-mkpdu>0</in-sak-mkpdu>
     <out-mkpdu>225271</out-mkpdu>
     <out-sak-mkpdu>0</out-sak-mkpdu>
    </counters>
   </state>
  </mka>
  <scsa-tx>
   <scsa-tx>
    <sci-tx>024f88a08c9d0001</sci-tx>
    <state>
     <sci-tx>024f88a08c9d0001</sci-tx>
     <counters>
      <sc-encrypted>0</sc-encrypted>
      <sa-encrypted>0</sa-encrypted>
    </counters>
   </state>
  </scsa-tx>
 </scsa-tx>
</interface>
<interface>
 <name>GigabitEthernet0_0_0_2</name>
 <state>
  <name>GigabitEthernet0_0_0_2</name>
  <counters>
   <tx-untagged-pkts>8</tx-untagged-pkts>
   <rx-untagged-pkts>0</rx-untagged-pkts>
   <rx-badtag-pkts>2</rx-badtag-pkts>
   <rx-unknownsci-pkts>3</rx-unknownsci-pkts>
   <rx-nosci-pkts>4</rx-nosci-pkts>
  </counters>
 </state>
 <mka>
  <state>
   <mka-policy>mp</mka-policy>
   <key-chain>kc</key-chain>
   <counters>
    <in-mkpdu>0</in-mkpdu>
    <in-sak-mkpdu>0</in-sak-mkpdu>
    <out-mkpdu>225271</out-mkpdu>
    <out-sak-mkpdu>0</out-sak-mkpdu>
   </counters>
  </state>
 </mka>
 <scsa-tx>
  <scsa-tx>
   <sci-tx>0246c822daae0001</sci-tx>
   <state>
    <sci-tx>0246c822daae0001</sci-tx>
    <counters>
     <sc-encrypted>0</sc-encrypted>
     <sa-encrypted>0</sa-encrypted>
    </counters>
   </state>
  </scsa-tx>
 </scsa-tx>
</interface>
<interface>
 <name>GigabitEthernet0/0/0/1</name>
 <config>
  <name>GigabitEthernet0/0/0/1</name>
 </config>
 <mka>
  <config>
   <key-chain>kc</key-chain>
  </config>
 </mka>
</interface>
<interface>
 <name>GigabitEthernet0/0/0/2</name>
 <config>
  <name>GigabitEthernet0/0/0/2</name>
 </config>
 <mka>
  <config>
   <key-chain>kc</key-chain>
   <mka-policy>mp</mka-policy>
  </config>
 </mka>
```

```
    </interface>
   </interfaces>
  </macsec>
 </data>
</rpc-reply>
```