

# 在IOS XR上从SNMP迁移到遥测

## 目录

[简介](#)

[SNMP](#)

[SNMP的组件](#)

[SNMP 管理器](#)

[SNMP 代理](#)

[SNMP MIB](#)

[SNMP操作](#)

[MIB和RFC](#)

[SNMP版本](#)

[阳模](#)

[OpenConfig模型](#)

[本地型号](#)

[遥测](#)

[模型驱动遥测](#)

[事件驱动遥测](#)

[传输](#)

[TCP](#)

[gRPC](#)

[gNMI/gNOI](#)

[编码](#)

[JSON](#)

[GPB-KV](#)

[GPB](#)

[IOS XR中的MDT配置](#)

[拨出模式](#)

[拨入模式](#)

[SNMP迁移到MDT](#)

[MIB迁移到XPath](#)

[BGP4-MIB](#)

[CISCO-BGP4-MIB](#)

[基于思科类的QOS-MIB](#)

[CISCO-ENHANCED-MEMPOOL-MIB](#)

[CISCO-ENTITY-FRU-CONTROL-MIB](#)

[CISCO-ENTITY-SENSOR-MIB](#)

[CISCO-FLASH-MIB](#)

[CISCO-PROCESS-MIB](#)

[ENTITY-MIB](#)

[IF-MIB](#)

[IP-MIB](#)

[IPMIB-COMMON](#)

[LLDP-MIB](#)

[MPLS-TE-STD-MIB](#)

[RFC2465-MIB](#)

[SNMP-MIB](#)

[TCP-MIB](#)

[UDP-MIB](#)

[SNMP陷阱迁移](#)

[安全考虑](#)

## 简介

介绍了简单网络管理协议(SNMP)的组成，并将基于SNMP监控的当前实现与模型驱动遥测(MDT)方法进行了关联。

## SNMP

SNMP 是一种应用层协议，可以为 SNMP 管理器和代理之间的通信提供消息格式。SNMP提供标准化框架和通用语言，用于监控和管理网络中的设备

### SNMP的组件

SNMP框架具有以下组件，这些组件在以下各节中介绍：

- [SNMP 管理器](#)
- [SNMP 代理](#)
- [SNMP MIB](#)

### SNMP 管理器

SNMP管理器是使用SNMP控制和监控网络主机活动的系统。最常见的管理系统是网络管理系统(NMS)。术语NMS可以应用于用于网络管理的专用设备，也可以应用于此类设备上使用的应用。

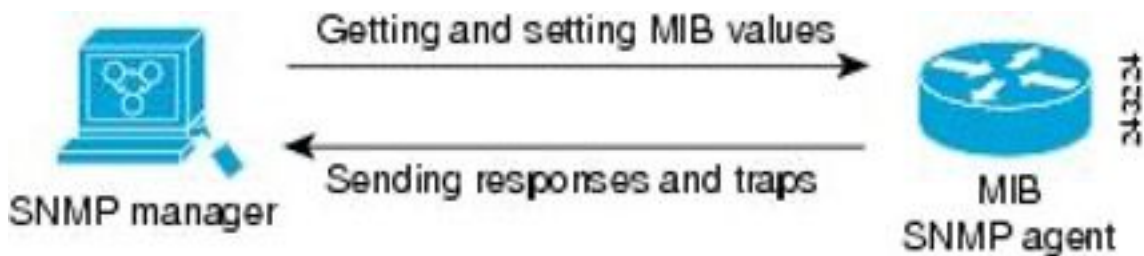
### SNMP 代理

SNMP代理是受管设备内的软件组件，用于维护设备的数据并根据需要向管理系统报告这些数据。代理驻留在路由设备（路由器、接入服务器或交换机）上。

### SNMP MIB

SNMP代理包含MIB变量，SNMP管理器可通过“获取”或“设置”操作请求或更改其值。管理员可以从代理获取值或将值存储在该代理中。代理从SNMP MIB收集数据，SNMP MIB是有关设备参数和网络数据的信息存储库。代理还可以响应管理器请求以获取或设置数据。

下图说明SNMP管理器和代理之间的通信。管理器发送代理请求以获取和设置SNMP MIB值。代理响应这些请求。与此交互无关，代理可以发送管理器未经请求的通知（陷阱或通知），以通知管理器有关网络条件。



## SNMP操作

SNMP应用执行以下操作以检索数据、修改SNMP对象变量和发送通知：

- [SNMP获取](#)
- [SNMP SET](#)
- [SNMP通知](#)

### SNMP获取

NMS执行SNMP GET操作以检索SNMP对象变量。GET操作有三种类型：

- GET — 从SNMP代理检索确切的对象实例。
- GETNEXT — 检索下一个对象变量，该对象变量是指定变量的字典继承。
- GETBULK — 检索大量对象变量数据，无需重复的GETNEXT操作。

### SNMP SET

NMS执行SNMP SET操作以修改对象变量的值。

### SNMP通知

SNMP的主要功能是能够从SNMP代理生成未经请求的通知。

未经请求的（异步）通知可以作为陷阱或通知请求（通知）生成。陷阱是向简单网络管理协议（SNMP）管理器发出网络状况警报的消息。通知是陷阱，包括从SNMP管理器请求确认接收。通知可能表示用户身份验证不正确、重新启动、连接关闭、与邻居设备的连接丢失或其他重要事件。

陷阱的可靠性比通知低，因为接收方在收到陷阱时不发送确认。发送方不知道是否收到陷阱。接收通知的SNMP管理器使用SNMP响应协议数据单元(PDU)确认消息。如果发送方从未收到响应，则可以再次发送通知。因此，通知更可能到达其预期目的地。

虽然陷阱不太可靠，但陷阱通常是首选，因为通知会消耗设备和网络中的更多资源。与陷阱不同，陷阱在发送后即被丢弃，在收到响应或请求超时之前，必须将通知保存在内存中。此外，陷阱仅发送一次，而通知可能重发多次。重试会增加流量并增加网络开销。使用陷阱和通知需要在可靠性和资源之间做出权衡。

## MIB和RFC

管理信息库(MIB)模块通常在提交给国际标准机构互联网工程任务组(IETF)的征求意见(RFC)文档中定义。RFC由个人或团体编写，供Internet协会和整个Internet社区考虑，通常旨在建立推荐的Internet标准。在获得RFC状态之前，建议将作为Internet草案(I-D)文档发布。已成为推荐标准的RFC也标记为标准文档(STD)。您可以通过Internet Society网站[http:// www.isoc.org](http://www.isoc.org)了解IETF的标

准流程和[活动](#)。您可以阅读IETF网站[http:// www.ietf.org](http://www.ietf.org)上Cisco文档中引用的所有RFC、I-D和STD的[全文](#)。

SNMP的思科实施使用RFC 1213中描述的MIB II变量定义和RFC 1215中描述的SNMP陷阱定义。

Cisco 对每个系统提供其自己的专用 MIB 扩展。除非文件另行通知，否则Cisco企业MIB应符合相关RFC描述的指南规定。您可以在Cisco.com上的Cisco MIB网站上找到MIB模块定义文件和每个思科平台支持的MIB列表。

## SNMP版本

目前，思科设备支持以下版本的SNMP:

- SNMPv1 — 简单网络管理协议：RFC 1157中定义的完整Internet标准。（RFC 1157取代了以RFC 1067和RFC 1098发布的早期版本。）安全性基于社区字符串。
- SNMPv2c - SNMPv2的基于社区字符串的管理框架。SNMPv2c（“c”表示“社区”）是RFC 1901、RFC 1905和RFC 1906中定义的实验性Internet协议。SNMPv2c是SNMPv2p(SNMPv2 Classic)协议操作和数据类型的更新，使用SNMPv1基于社区的安全模型。
- SNMPv3 - SNMP版本3。SNMPv3是RFC 3413到3415中定义的基于标准的可互操作协议。SNMPv3通过验证和加密网络上的数据包来提供对设备的安全访问。

SNMPv3中提供的安全功能如下：

- 消息完整性 — 确保数据包在传输过程中未被篡改。
- 身份验证 — 确定消息来自有效源。
- 加密 — 加扰数据包的内容，防止其被未授权的源获取。

SNMPv1 和 SNMPv2c 都使用基于社区形式的安全性。SNMP管理器的团体能够访问代理MIB，这是由团体字符串定义的。

SNMPv2c支持包括批量检索机制和向管理站报告详细错误消息。批量检索机制支持检索表和大量信息，从而最大限度地减少所需的往返次数。SNMPv2c改进的错误处理支持包括扩展的错误代码，可区分不同类型的错误；这些情况通过SNMPv1中的单个错误代码报告。还报告了以下三种类型的异常：没有此类对象、没有此类实例和MIB视图结束。

SNMPv3是一种安全模型，其中为用户和用户所在的组设置了身份验证策略。安全等级是安全模型中允许的安全级别。安全模型和安全级别的组合确定在处理SNMP分组时采用的安全机制。

有三种安全模式：SNMPv1、SNMPv2c和SNMPv3。下表列出了安全模型和级别的组合及其含义。

型号	级别	身份验证	加密	发生的情况
v1	noAuthNoPriv	公用字符串	无	使用团体字符串匹配进行身份验证。
v2c	noAuthNoPriv	公用字符串	无	使用团体字符串匹配进行身份验证。
v3	noAuthNoPriv	用户名	无	使用用户名匹配进行身份验证。
v3	authNoPriv	消息摘要5(MD5)或安全散列算法(SHA)	无	提供基于 HMAC-MD5 或 HMAC-SHA 算法的身份验证。
v3	authPriv	MD5或SHA	数据加密标准 (DES)	提供基于 HMAC-MD5 或 HMAC-SHA 算法的身份验证。除了提供基于 CBC-DES (DES-56) 标准的身份验证，还提供 DES 56 位加密。

应实施SNMP代理，以使用管理站支持的SNMP版本。座席可以与多个管理器通信。

SNMPv3支持RFC 1901至1908、2104、2206、2213、2214和2271至2275。有关SNMPv3的其他信息，请参阅RFC 2570, Internet标准网络管理框架第3版简介（这不是标准文档）。

## 阳模

Yang模型表示系统特定特征或硬件特征的树结构抽象。在网络元素中，杨模型可以表示路由协议、内部物理传感器阵列。RFC 6020中介绍了YANG语言和术语，RFC 7950中的下一次更新也介绍了。在高层中，杨模型将代表主结构的数据组织成子模块和容器，这些子模块和容器是与子节点相关的列表。接下来将介绍几种节点类型。

枝叶节点包含简单数据，如整数或字符串。它只有一个特定类型的值，没有子节点。

```
leaf host-name {  
    类型字符串；  
    说明“此系统的主机名”；  
}
```

枝叶列表是枝叶节点序列，每个枝叶的特定类型的值恰好为一个。

```
leaf-list domain-search {  
    类型字符串；  
    描述“要搜索的域名列表”；  
}
```

容器节点用于对子树中的相关节点进行分组。容器只有子节点，没有值。容器可以包含任意数量的任何类型的子节点（包括枝叶、列表、容器和枝叶列表）。

```
容器系统{  
    container login {  
        叶消息{  
            类型字符串；  
            描述  
                “登录会话开始时提供的消息”；  
        }  
    }  
}
```

列表定义列表条目的序列。每个条目都像一个结构或记录实例，由其关键枝叶的值唯一标识。列表可以定义多个关键枝叶，并且可以包含任意数量的任何类型的子节点（包括枝叶、列表、容器等）。

最后，将所有这些注释类型绑定在一起的示例模型类似于以下示例：

```
## Contents of "example-system.yang" module example-system { yang-version 1.1; namespace  
"urn:example:system"; prefix "sys"; organization "Example Inc."; contact "joe@example.com";  
description "The module for entities implementing the Example system."; revision 2007-06-09 {  
description "Initial revision."; } container system { leaf host-name { type string; description  
"Hostname for this system."; } leaf-list domain-search { type string; description "List of  
domain names to search."; } container login { leaf message { type string; description "Message  
given at start of login session."; } list user { key "name"; leaf name { type string; } leaf
```

```
full-name { type string; } leaf class { type string; } } } }
```

然而，Yang Models上使用的Yang语言并不表示数据是否组织为容器/列表/枝叶。这就是为什么网络元素上的某些特征可以用不同的杨模型来表示的原因。此挑战已通过以下Yang模型类型得到解决：

- [OpenConfig模型](#)
- [本地型号](#)

## OpenConfig模型

OpenConfig模型是使用不可知供应商组织为代表特定功能的模型开发的，此方法的优点是NMS可以使用这些模型与多供应商甚至多平台环境中的网络元素交互。

如名称所示，这些模型是打开的，可公开检查此链接上的存储库（如github）：

<https://github.com/openconfig/public/tree/master/release/models>

例如，您可以找到边界网关协议(BGP)的openconfig模型、链路聚合控制协议(LACP)的openconfig模型和ISIS的openconfig模型，具有不同的特定模型。对于BGP，您可以找到BGP错误的模型、BGP策略等的模型。模型可以是相关的，有些模型可以称之为另一个杨包。例如，openconfig-bgp-neighbor.yang属于openconfig-bgp.yang：

```
module openconfig-bgp { yang-version "1"; ## namespace namespace
"http://openconfig.net/yang/bgp"; prefix "oc-bgp"; ## import some basic inet types import
openconfig-extensions { prefix oc-ext; } import openconfig-rib-bgp { prefix oc-bgprib; } ##
Include the OpenConfig BGP submodules ## Common: defines the groupings that are common across
more than ## one context (where contexts are neighbor, group, global) include openconfig-bgp-
common; ## Multiprotocol: defines the groupings that are common across more ## than one context,
and relate to Multiprotocol include openconfig-bgp-common-multiprotocol; ## Structure: defines
groupings that are shared but are solely used for ## structural reasons. include openconfig-bgp-
common-structure; ## Include peer-group/neighbor/global - these define the groupings ## that are
specific to one context include openconfig-bgp-peer-group; include openconfig-bgp-neighbor;
include openconfig-bgp-global;
```

总之，OpenConfig模型面向所有平台通用的协议，如IETF或RFC标准化功能。

## 本地型号

相反，本地模型是面向供应商的模型，涵盖特定平台的深度结构。例如，将网络元素内的物理值传感器分组的型号，如电压、温度、ASIC计数器、交换矩阵计数器等。由于它们依赖于平台，因此通常会查找NCS6K、ASR9K或Cisco 8000的特定型号。

作为OpenConfig模型，Github存储库中也提供本地模型：

<https://github.com/YangModels/yang/tree/master/vendor/cisco/xr>

由于这些型号往往比OpenConfig型号更具体、更完整，因此它们与特定软件版本相关联，并且可能会在软件版本之间发生变化。

本机模型有两个主要类别：

- “操作”模型，用于从元素检索信息。

例如，[Cisco-IOS-XR-eigrp-oper.yang](#)

- “Cfg”型号，用于配置网络元素

例如，[Cisco-IOS-XR-eigrp-cfg.yang](#)

一般来说，模型驱动遥测使用“操作”模型来传输基础设施中的数据，而像NSO这样的NMS使用“cfg”模型来更改网络元素上的配置。

本地和OpenConfig Yang型号存在于/pkg/yang文件夹上的XR软件中，可以列出，以确定平台上是否有任何Yang型号可用。本示例适用于运行cXR 6.4.2的XRrv9k:

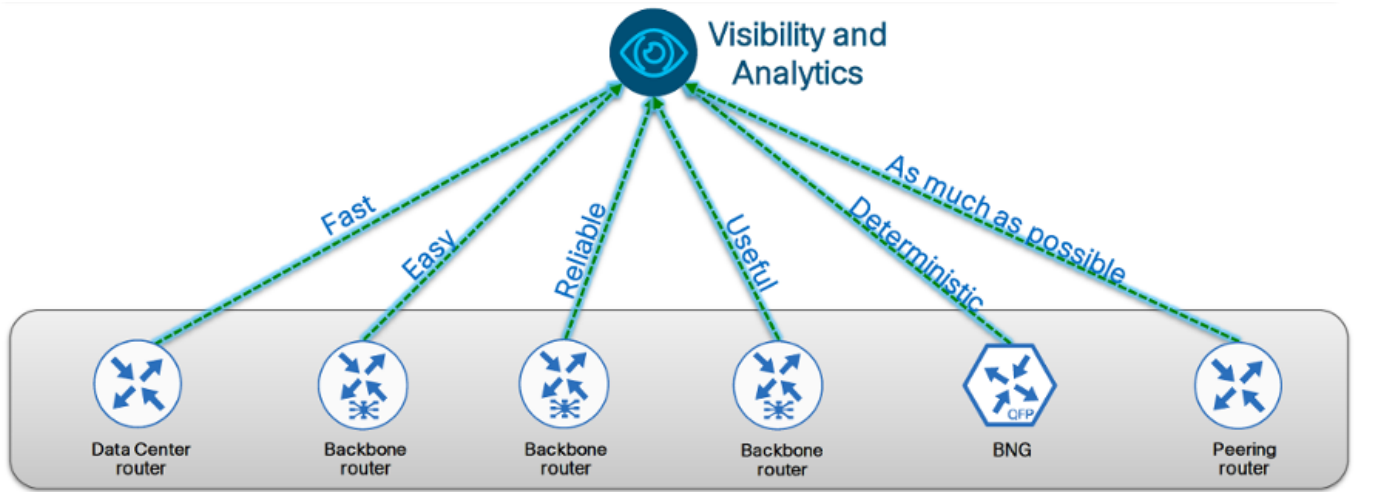
```
RP/0/RP0/CPU0:xrv9k1#run ls /pkg/yang | grep isis
星期二9月22日14:21:27.471 CLST
Cisco-IOS-XR-clns-isis-cfg.yang
Cisco-IOS-XR-clns-isis-datatypes.yang
Cisco-IOS-XR-clns-isis-oper-sub1.yang
Cisco-IOS-XR-clns-isis-oper-sub2.yang
Cisco-IOS-XR-clns-isis-oper-sub3.yang
Cisco-IOS-XR-clns-isis-oper.yang
Cisco-IOS-XR-isis-act.yang
openconfig-isis-lsdb-types.yang
openconfig-isis-lsp.yang
openconfig-isis-policy.yang
openconfig-isis-routing.yang
openconfig-isis-types.yang
openconfig.isis.yang
RP/0/RP0/CPU0:xrv9k1#
```

## 遥测

遥测是一种允许从不同远程元素收集信息到汇聚可视性和分析层的中心位置的过程。

在网络环境中，数据可能由网络中的每个元素、路由器、其他元素之间的交换生成，信息可能与物理传感器中的一组非常大的特定协议、性能计数器或度量相关。





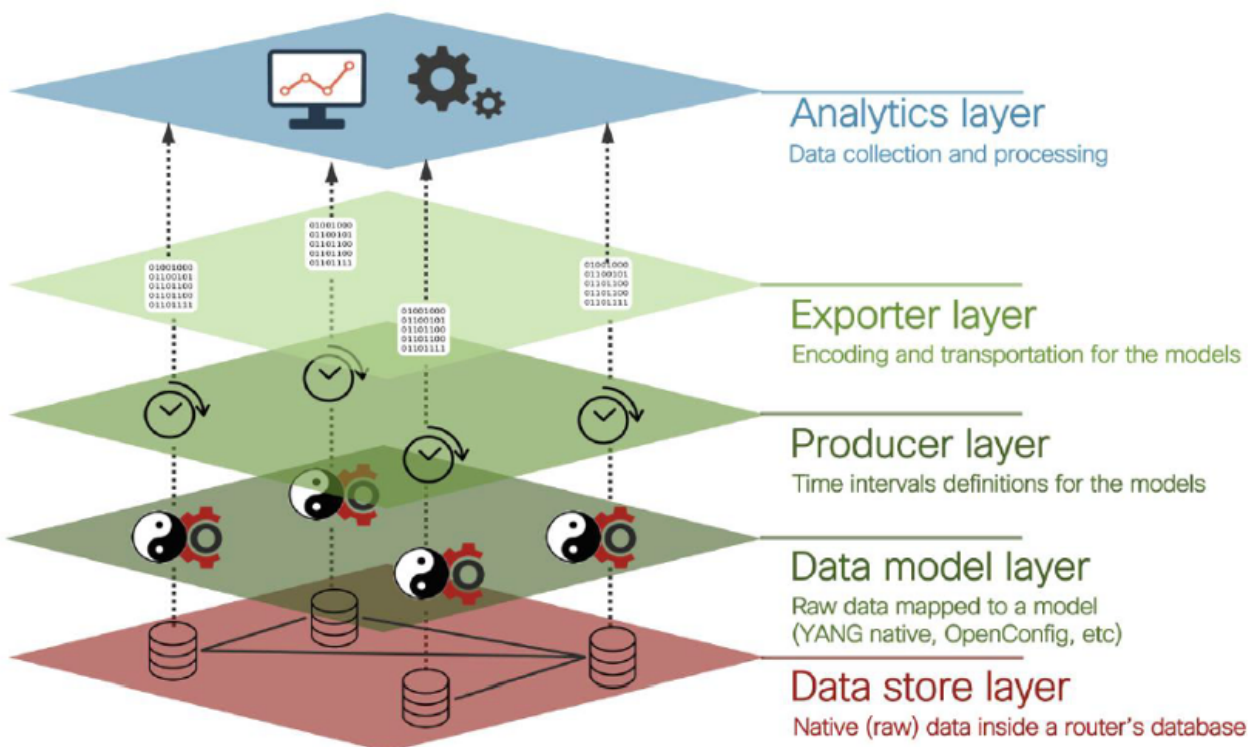
通常，可视性和分析功能位于网络中的中心点，遥测信息使用网络传输机制进行流传输，因此遥测信息应尽可能快地扩展。

与SNMP传统机制相反，遥测采用推式模式，网络应调配以流传输自己的数据，而不必定期轮询，这是基于SNMP的监控的主要特征。此调配通常称为订用，它基于一组要监控的变量、数据收集的采样间隔的常规间隔以及通过网络发送此数据的远程系统。

### 模型驱动遥测

模型驱动遥测的MDT状态，如名称所示，它基于杨模型。网络设备的每个方面都可以用Yang模型来表示，例如，模块化系统上每个组件的OSPF邻居表、RIB或温度传感器。

关于MDT架构，它可分为以下几层：



注意：对于生成层，在模型驱动遥测中有一个采样间隔定义，它控制设备查询内部原始数据库并将这些数据组织到数据模型层的频率。

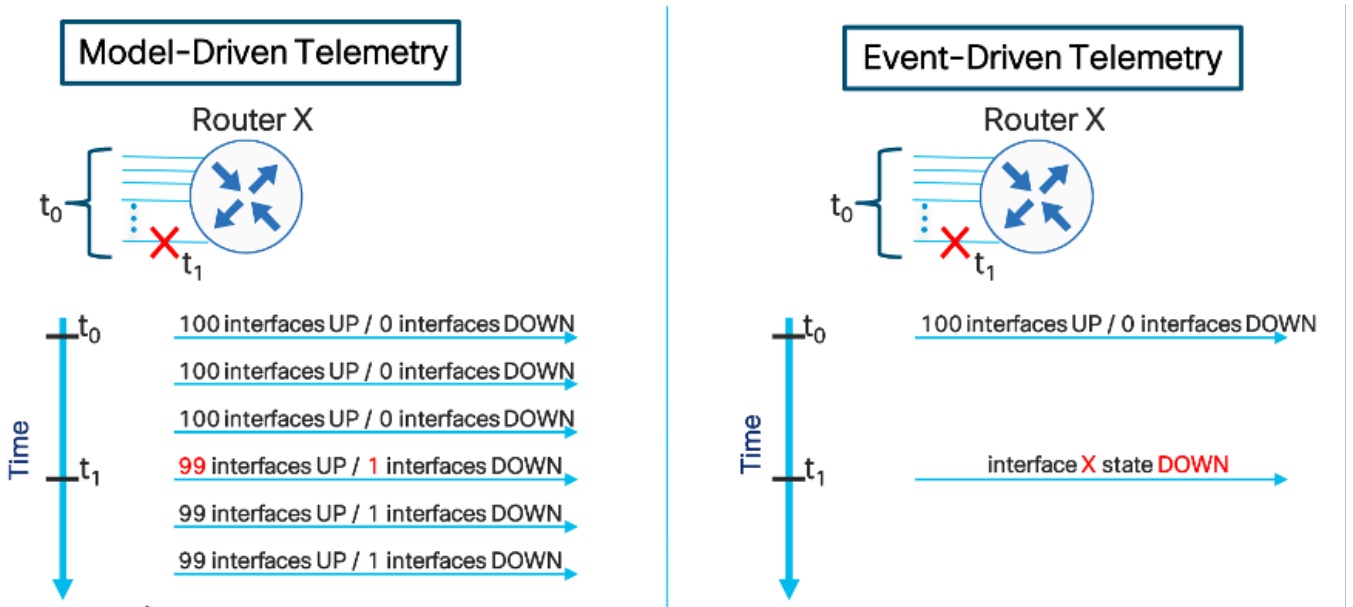


遥测订用还定义了哪些型号和容器/路径将产生要流入分析层的数据。此定义将影响与业务相关的信息。此传感器路径的MDT定义是模拟定义OID以通过SNMP检索，因为这两种技术都以定义的采样率生成结构化数据。

## 事件驱动遥测

EDT代表事件驱动遥测，也基于杨氏模型来构建。主要区别在于，收集和流的数据的触发器不是固定间隔，而是特定事件，如阈值交叉、链路事件、硬件故障等。

下面将比较事件与模型驱动遥测和事件驱动遥测：



提示：此图显示使用MDT的冗余消息，但仅显示使用EDT表示更改的消息。

## 传输

遥测应尽可能可靠，因此使用基于传输控制协议(TCP)的传输在基础设施和分析层之间使用面向会话的套接字是有意义的，该层应实施用于建立会话的收集器。

使用遥测时有两种主要方法，在三次握手初始流中，它们之间存在差异。

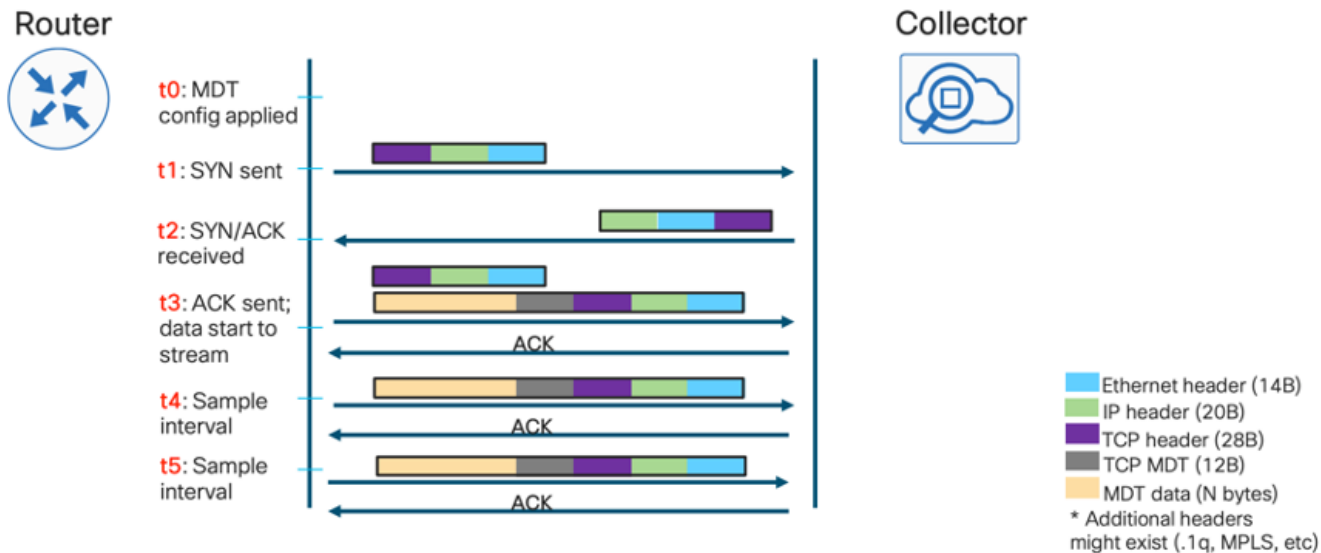


注意：在拨出模式下，会话的设置基础设施端启动，这意味着应该在网络元素上配置感兴趣的传感器。相比之下，拨号方式允许在网络元素上进行更轻的配置，因为收集器应在设置阶段要求特定

## 传感器路径。

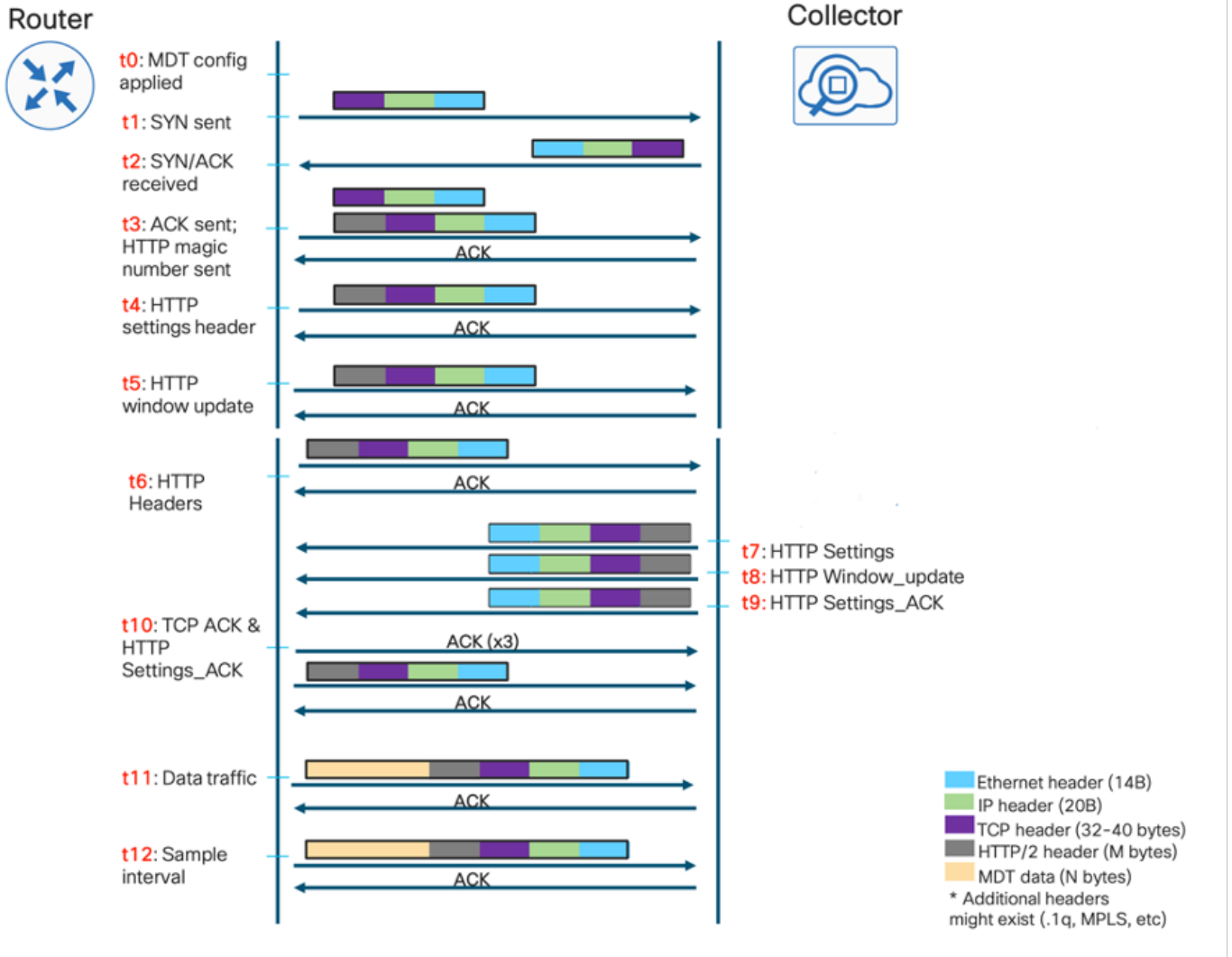
### TCP

TCP是在网络元素和遥测收集器之间建立面向连接会话的最简单方法，数据流从路由器到收集器，然后为了可靠性目的将ACK发送回路由器：



### gRPC

由于Google协议RPC(gRPC)在超文本传输协议/2(HTTP/2)上工作，因此会话本身应在设置时形成，并允许从收集器端本地进行速度控制：



## gNMI/gNOI

gRPC网络管理接口(gNMI)是Google开发的gRPC网络管理协议。gNMI提供了安装、操作和删除网络设备配置以及查看运行数据的机制。通过gNMI提供的内容可以用YANG建模。

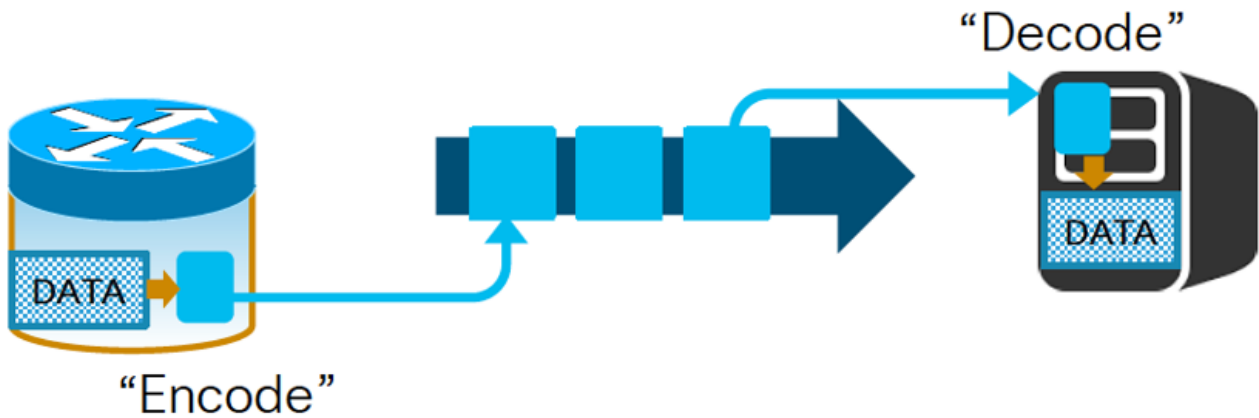
gNMI使用gRPC-HTTP/2来建立连接，并在网络元素和NMS之间提供双向通道，NMS也可以是遥测收集器，但也提供和接口来管理设备。

在此协议支持的操作之间，我们可以找到gNMI Get、gNMI Set，返回请求的信息、成功或错误消息。

gRPC网络操作接口(gNOI)是使用与gNMI相同的通信信道的微服务集合，但允许与配置本身无关的常规操作，如ping、重新启动、更改SSL证书、清除等。

## 编码

Yang模型定义了数据的结构、层次结构以及每个枝叶节点的类型。但是，建模并不表示应如何序列化此数据。此过程控制从结构化数据到要通过TCP连接（原始TCP、gRPC、gNMI等）发送的字节流的转换。



注意：此过程应在网络元素中使用等效的机制来实现，该机制应对数据进行编码，并且收集器应对此数据进行解码。

## JSON

第一种编码机制是本地JavaScript Object Notation(JSON)格式，该格式众所周知，但是以人为本，因为它将每个键都表示为字符串，在消息大小方面效率低下。使用JSON的主要优势是易于解析，并以基于文本的方式读取为下一个示例：

```
{ "node_id_str": "test-IOSXR ", "subscription_id_str": " if_rate", "encoding_path": "Cisco-IOS-XR-
infra-statsdoper:infra-statistics/interfaces/interface/latest/datarate", "collection_id": 49,
"collection_start_time": 1510716302467, "msg_timestamp": 1510716302479, "data_json": [ {
"timestamp": 1510716282334, "keys": { "interface-name": "Null0" }, "content": { "input-data-rate": 0,
"input-packet-rate": 0, "output-data-rate": 0, "output-packet-rate": 0, <> { "timestamp":
1510716282344, "keys": { "interface-name": "GigabitEthernet0/0/0/0" }, "content": { "input-data-
rate": 8, "input-packet-rate": 1, "output-data-rate": 2, "output-packet-rate": 0, <>
"collection_end_time": 1510716302372 } }
```

## GPB-KV

Google协议缓冲密钥值(GPB-KV)编码格式也称为自描述GPB，因为它利用协议缓冲区来使用指向Yang模型上特定元素的消息。这意味着只需一个.proto文件即可进行编码/解码，并且数据中的密钥本身是自描述字符串。

```
node_id_str: "test-IOSXR" subscription_id_str: "if_rate" encoding_path: "Cisco-IOS-XR-infra-
statsd-oper:infrastatistics/interfaces/interface/latest/data-rate" collection_id: 3
collection_start_time: 1485793813366 msg_timestamp: 1485793813366 data_gpbkv { timestamp:
1485793813374 fields { name: "keys" fields { name: "interface-name" string_value: "Null0" } }
fields { name: "content" fields { name: "input-data-rate" 8: 0 } fields { name: "input-packet-
rate" 8: 0 } fields { name: "output-data-rate" 8: 0 } fields { name: "output-packet-rate" 8: 0 }
<> data_gpbkv { timestamp: 1485793813389 fields { name: "keys" fields { name: "interface-name"
string_value: "GigabitEthernet0/0/0/0" } } fields { name: "content" fields { name: "input-data-
rate" 8: 8 } fields { name: "input-packet-rate" 8: 1 } fields { name: "output-data-rate" 8: 2 }
fields { name: "output-packet-rate" 8: 0 } <> } ... collection_end_time: 1485793813405
```

## GPB

最后，Google协议缓冲区(GPB) (也称为紧凑型GPB) 进一步采用此方法，并要求.proto文件映射结构的每个密钥，从而使其在消息大小方面效率更高，因为所有信息都以二进制值发送。但是，缺

点是需编译与基础架构/收集器支持的每个Yang模型关联的每个.proto文件。

```
node_id_str: "test-IOSXR" subscription_id_str: "if_rate" encoding_path: "Cisco-IOS-XR-infra-
statsdoper:infrastatistics/interfaces/interface/latest/data-rate" collection_id: 5
collection_start_time: 1485794640452 msg_timestamp: 1485794640452 data_gpb { row { timestamp:
1485794640459 keys: "\n\n005Null10" content: "\220\003\000\230\003\000\240\003\000\250\0
03\000\260\003\000\270\003\000\300\003\000\ 310\003\000\320\003\000\330\003\t\340\003\00
0\350\003\000\360\003\377\001" } row { timestamp: 1485794640469 keys:
"\n\n026GigabitEthernet0/0/0/0" content: "\220\003\010\230\003\001\240\003\002\250\0
03\000\260\003\000\270\003\000\300\003\000\ 310\003\000\320\003\300\204=\330\003\000\34
0\003\000\350\003\000\360\003\377\001" } collection_end_time: 1485794640480
```

## IOS XR中的MDT配置

流模式驱动遥测数据中使用的核心组件包括：

- 会话
- 传感器路径
- 订用
- 传输和编码

会话选项可以是拨入或拨出，如前所述。在IOS XR中构建配置。

### 拨出模式

对于拨出模式，路由器根据订用启动到目的地的会话，该过程应包括以下步骤：

- 创建目标组
- 创建传感器组
- 创建订用
- 验证拨出配置

要创建目标组，您需要知道收集器的Internet协议版本4(IPv4)/Internet协议版本6(IPv6)地址以及为此应用程序提供服务的端口。此外，您需要指定网络设备和收集器上应同意的协议和编码。

最后，您可能需要指定用于与收集器网络地址通信的虚拟路由和转发(VRF)。

接下来，显示一个拨出配置示例：

```
遥测模型驱动
目的组DG1
VRF管理
address-family ipv4 192.168.122.20端口5432
编码自描述 — gpb
协议TCP
!
```

编码选项如下所示：

```
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#encoding ?
GPB编码
JSON编码
自描述 — gpb自描述GPB编码←也称为GPB-KV
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#encoding
```

协议选项：

```

RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#protocol ?
  grpc gRPC
  TCP
  UDP
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#protocol grpc ?
  gzip gRPC gzip消息压缩
  no-tls无TLS
  tls-hostname TLS主机名
  <cr>
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#protocol tcp ?
  <cr>
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#protocol udp ?
  数据包大小UDP数据包大小
  <cr>
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#protocol udp

```

TCP协议简单明了，只需要将端口设置附加到IPv4/IPv6地址。与之相反，用户数据报协议(UDP)是无连接协议，因此目标组状态始终处于活动状态。

gRPC中的压缩可通过使用可选的gzip关键字来实现。gRPC默认使用TLS，因此应在本地路由器上安装证书，以实现此用途。可以通过配置no-tls关键字覆盖此行为。最后，您可以使用tls-hostname关键字为证书指定不同的主机名。

接下来，应添加传感器组部分，列出我们感兴趣的传感器路径。本节内容简单明了，但必须知道传感器路径本身允许过滤以优化多个资源，如中央处理器(CPU)和带宽。

```

遥测模型驱动
  传感器组SG1
    sensor-path Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
    sensor-path Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface[interface-name='Mgmt*']/data-rate
  !
!

```

**注意：**传感器路径所需的格式为<model-name>:<container-path>

本文档介绍使用OID从基于SNMP的监控到YANG模型的映射，OID表示此传统方法中的“枝叶”，XPath表示与相同“枝叶”匹配的XPath。

最终配置阶段应配置订用，该订用将传感器组与遥测流的频率关联到目标组。

```

遥测模型驱动
  订用SU1
    sensor-group-id SG1 sample-interval 5000
    destination-id DG1
  !
!

```

此示例使用5000毫秒（5秒）的采样间隔，该间隔相对于上一个集合的结束。要更改此行为，可以使用strict-timer选项更改sample-interval关键字。

为进行验证，您可以使用以下涵盖订用状态的命令。此方法还允许覆盖传感器组和目的组信息。

```

RP/0/RP0/CPU0:C8000-1#sh遥测型号驱动订用SU1
世界协调时11月18日星期三15:38:01.397
订用： SU1
-----
  状态： 主用
  传感器组：
  ID:SG1
    示例间隔： 5000 ms
    检测信号间隔： 不适用
    传感器路径： Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface[interface-name='Mgmt*']/data-rate
    传感器路径状态： 已解决
    传感器路径： Cisco-IOS-XR-wdsysmon-fd-oper：系统监控/cpu利用率
    传感器路径状态： 已解决
  目标组：
  Group ID:DG1
  目标 IP： 192.168.122.10

```



```

目的端口： 5432
目标VRF： MGMT(0x60000001)
编码： self-describing-gpb
传输： tcp
状态： 主用
TLS： 错误
发送的总字节数： 636284346
发送的数据包总数： 4189
上次发送时间： 2020-11-18 15:37:58.1700077650 +0000
收集组：
—
ID:9
示例间隔： 5000 ms
检测信号间隔： 不适用
心跳始终： 错误
编码： self-describing-gpb
集合数： 1407
收集时间： 最小值： 最大4毫秒： 13 ms
总时间： 最小值： 平均8毫秒： 最大10毫秒： 20 毫秒
延迟总数： 0
发送错误总数： 0
发送丢弃总数： 0
其他错误总数： 0
无数据实例： 1407
上次收集开始时间：2020-11-18 15:37:57.1699545994 +0000
上次收集结束时间：2020-11-18 15:37:57.1699555589 +0000
传感器路径： Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/data-rate
ID:10
示例间隔： 5000 ms
检测信号间隔： 不适用
心跳始终： 错误
编码： self-describing-gpb
集合数： 1391
收集时间： 最小值： 最大178毫秒： 473 ms
总时间： 最小值： 247毫秒平均： 最大283毫秒： 559 ms
延迟总数： 0
发送错误总数： 0
发送丢弃总数： 0
其他错误总数： 0
无数据实例： 0
上次收集开始时间：2020-11-18 15:37:58.1699805906 +0000
上次收集结束时间：2020-11-18 15:37:58.1700078415 +0000
传感器路径： Cisco-IOS-XR-wdsysmon-fd-oper：系统监控/cpu利用率
RP/0/RP0/CPU0:C8000-1#

```

## 拨入模式

在拨入模式下，收集器启动与网络元素的连接。然后，收集器应指明构建订用的意向。

配置包括以下步骤：

- 启用gRPC服务
- 设置传感器组
- 确认

要启用gRPC服务，配置将显示在下面：

```

!
grpc
VRF管理
端口 57400
no-tls
地址系列双
!

```

选项很简单，包括VRF和TCP端口。默认情况下，gRPC使用TLS，但可以使用no-tls关键字禁用它。最后，地址系列双选项允许使用IPv4和IPv6进行连接。

接下来，拨入需要本地定义传感器组，收集器稍后将使用该传感器组来定义订用。

```

遥测模型驱动
传感器组SG3
sensor-path Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
sensor-path Cisco-IOS-XR-fib-common-oper:fib-statistics/nodes/node/drops

```

此时，指向拨入模式的配置已完成，并且收集器本身可以使用gRPC订阅路由器。为进行验证，您可以采用与拨出模式相同的方法：

```
RP/0/RP0/CPU0:C8000-1#sh遥测型号驱动订阅anx-1605878175837
11月20日星期五13:58:37.894 UTC
订阅： anx-1605878175837
```

```
—
状态： 主用
传感器组：
ID:SG3
  示例间隔： 15000 ms
  检测信号间隔： 不适用
  传感器路径： Cisco-IOS-XR-wdsysmon-fd-oper：系统监控/cpu利用率
  传感器路径状态： 已解决
  传感器路径： Cisco-IOS-XR-fib-common-oper:fib-statistics/nodes/node/drops
  传感器路径状态： 已解决
目标组：
Group ID:拨入_1003
目标 IP： 192.168.122.10
目的端口： 46974
压缩： gzip
编码： json
传输： 拨号
状态： 主用
TLS： 错误
发送的总字节数： 71000035
发送的数据包总数： 509
上次发送时间： 2020-11-20 13:58:32.1030932699 +0000
收集组：
```

```
—
ID:5
  示例间隔： 15000 ms
  检测信号间隔： 不适用
  心跳始终： 错误
  编码： json
  集合数： 170
  收集时间： 最小值： 最大273毫秒： 640 ms
  总时间： 最小值： 平均276毫秒： 最大390毫秒： 643 ms
  延迟总数： 0
  发送错误总数： 0
  发送丢弃总数： 0
  其他错误总数： 0
  无数据实例： 0
  上次收集开始时间： 2020-11-20 13:58:32.1030283276 +0000
  上次收集结束时间： 2020-11-20 13:58:32.1030910008 +0000
  传感器路径： Cisco-IOS-XR-wdsysmon-fd-oper：系统监控/cpu利用率
ID:6
  示例间隔： 15000 ms
  检测信号间隔： 不适用
  心跳始终： 错误
  编码： json
  集合数： 169
  收集时间： 最小值： 最大15毫秒： 33 ms
  总时间： 最小值： 平均17毫秒： 最大22毫秒： 33 ms
  延迟总数： 0
  发送错误总数： 0
  发送丢弃总数： 0
  其他错误总数： 0
  无数据实例： 0
  上次收集开始时间： 2020-11-20 13:58:32.1030910330 +0000
  上次收集结束时间： 2020-11-20 13:58:32.1030932787 +0000
  传感器路径： Cisco-IOS-XR-fib-common-oper:fib-statistics/nodes/node/drops
RP/0/RP0/CPU0:C8000-1#
```

**提示：请注意，路由器上没有对拨号模式的节奏、编码、收集器IP或传输进行硬编码。**

## SNMP迁移到MDT

要完成从传统SNMP到遥测模型的迁移，应从以下方面进行介绍：

- MIB迁移到XPATH
- 陷阱迁移到遥测

- 安全考虑

## MIB迁移到XPATH

为此，我们可以使用MIB自己的层次结构对MIB进行分类，这些层次结构可以（至少在高级别上）映射到特定功能。

### BGP4-MIB

下表表示OID名称和编号，以及要在与BGP对等会话相关的模型驱动遥测传感器组上设置的相应XPATH。

OID名称	OID编号	OID说明	XPATH
bgpPeerLastError	1.3.6.1.2.1.15.3.1.14	此对等体在此连接上看到的最后一个错误代码和子代码。如果未发生错误，则此字段为零。否则，这两个字节的二进制八位数字字符串的第一个字节包含错误代码，而第二个字节包含子代码。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/neighbor-missing-error-table/neighbor/last-notify-error-code
bgpPeerOutUpdates	1.3.6.1.2.1.15.3.1.11	此连接上传输的BGP UPDATE消息数。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor/update-messages-out
bgpPeerInUpdates	1.3.6.1.2.1.15.3.1.10	此连接上收到的BGP UPDATE消息数。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor/update-messages-in
bgpPeerNegotiatedVersion	1.3.6.1.2.1.15.3.1.4	两个对等体之间运行的BGP的协商版本。除非bgpPeerState处于openconfirm或established状态，否则此条目必须为零(0)。请注意，此对象的法定值介于0和255之间。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor/neighbor-protocol-version
bgpPeerState	1.3.6.1.2.1.15.3.1.2	BGP对等连接状态。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor/connection-state
bgpPeerRemoteAddr	1.3.6.1.2.1.15.3.1.7	此条目的BGP对等体的远程IP地址。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor/connection-remote-address

bgpPeerLocalAddr	1.3.6.1.2.1.15.3.1.5	此条目的BGP连接的本地IP地址。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor/connection-local-address
bgpPeerFsmEstablishedTime	1.3.6.1.2.1.15.3.1.16	此计时器指示此对等体处于已建立状态的时间（以秒为单位）或自此对等体上次处于已建立状态以来的时间。当配置新对等体或启动路由器时，该值设置为零。BGP连接的所需状态。从“停止”到“启动”的过渡将导致生成BGP手动启动事件。从“start”到“stop”的转换将导致生成BGP手动停止事件。此参数可用于重新启动BGP对等连接。在不提供足够身份验证的情况下，应谨慎提供对此对象的写访问。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor/connection-established-time
bgpPeerAdminStatus	1.3.6.1.2.1.15.3.1.3		Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor/connection-admin-status

## CISCO-BGP4-MIB

下表表示OID名称和编号以及要在与BGP会话状态和前缀互换相关的模型驱动遥测传感器组上设置的相应XPath。

OID名称	OID编号	OID说明	XPATH
cbgpPeer2RemoteAs	1.3.6.1.4.1.9.9.187.1.2.5.1.11	在BGP OPEN消息中收到的远程自治系统编号。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/sessions/session/remote-as
cbgpPeer2PrevState	1.3.6.1.4.1.9.9.187.1.2.5.1.29	BGP对等体连接以前的状态。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor/previous-connection-state
cbgpPeer2State	1.3.6.1.4.1.9.9.187.1.2.5.1.3	BGP对等连接状态。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor/connection-state
cbgpPeer2LocalAddr	1.3.6.1.4.1.9.9.187.1.2.5.1.6	此条目的BGP连接的本地IP地址。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor/connection-local-address
cbgpPeer2AdvertisedPrefixes	1.3.6.1.4.1.9.9.187.1.2.8.1.6	当在此连接上通告属于地址系列的路由前	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor/connection-advertised-prefixes

			缀时，此计数器将递增。当连接经历硬重置时，它将初始化为零。	instance-active/default-vrf/af/neighbor-af-table/neighbor-af-data/prefixes-advertised
cbgpPeer2AcceptedPrefixes	1.3.6.1.4.1.9.9.187.1.2.8.1.1		此连接上属于地址系列的已接受路由前缀的数量。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor-af-data/prefixes-accepted
cbgpPeerPrefixLimit	1.3.6.1.4.1.9.9.187.1.2.1.1.3		此连接上接受的最大路由前缀数	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/af/neighbor-af-table/neighbor-af-data/prefix-limit
cbgpPeer2PrefixThreshold	1.3.6.1.4.1.9.9.187.1.2.8.1.4		此连接上地址系列的前缀阈值(%), 在此时, 会生成警告消息, 说明前缀计数超过阈值或生成相应的SNMP通知。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/config-instances/config-instance/default-vrf/entity-configurations/entity-configuration/af-depend-config/max-prefix-warning-threshold

## 基于思科类的QOS-MIB

下表表示OID名称和编号，以及要在与服务质量(QoS)类/策略中的统计信息相关的模型驱动遥测传感器组上设置的相应XPath。

OID名称	OID编号	OID说明	XPATH
cbQosCMDropBitRate	1.3.6.1.4.1.9.9.166.1.15.1.1.18	每个类的丢包的比特率，因为所有可产生丢包的功能（例如，警察、随机检测等）。	Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/input/service-policy-names/service-policy-instance/statistics/class-stats/general-stats/total-rate
cbQosCMDropPkt64	1.3.6.1.4.1.9.9.166.1.15.1.1.14	每类丢弃数据包的64位计数器，这是可产生丢包的所有功能（例如，警察、随机检测等）的结果。	Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/input/service-policy-names/service-policy-instance/statistics/class-stats/general-stats/total-packets

cbQosCMPrePolicyPkt64	1.3.6.1.4.1.9.9.166.1.15.1.1.3	在执行任何QoS策略之前，64位会计入入站数据包。	Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/output/se- policy-names/service-po- instance/statistics/class- stats/general-stats/total- packets Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/input/serv- policy-names/service-po- instance/statistics/class- stats/general-stats/pre-p- matched-packets Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/output/se- policy-names/service-po- instance/statistics/class- stats/general-stats/pre-p- matched-packets
cbQosCMName	1.3.6.1.4.1.9.9.166.1.7.1.1.1	类映射的名称。	Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/interface/ service-policy-names/se- policy- instance/statistics/class- stats/class-name Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/interface/ service-policy-names/se- policy- instance/statistics/class- stats/child-policy/class- stats/general-stats/trans- bytes
cbQosCMPostPolicyByte64	1.3.6.1.4.1.9.9.166.1.15.1.1.1 0	执行QoS策略后，64位的出站八位组计数。	Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/output/se- policy-names/service-po- instance/statistics/class- stats/child-policy/class- stats/general-stats/trans- bytes
cbQosIfIndex	1.3.6.1.4.1.9.9.166.1.1.1.1.4	ifIndex，用于此服务所连接的接口。仅当逻辑接口具有snmp ifIndex时，此字段才有意义。例如，当cbQosIfType为controlPlane时，此字	Cisco-IOS-XR-infra- policymgr-oper:policy- manager/global/policy- map/policy-map-types/p- map-type/policy-maps



cbQosConfigIndex	1.3.6.1.4.1.9.9.166.1.5.1.1.2	段的值是无意义的。 每个对象的任意（系统分配的）配置（独立于实例）索引。具有相同配置的每个对象共享相同的配置索引。	Cisco-IOS-XR-infra-policymgr-oper:policy-manager/global/policy-map/policy-map-types/policy-map-type/policy-maps
cbQosCMPrePolicyByte64	1.3.6.1.4.1.9.9.166.1.15.1.1.6	在执行任何QoS策略之前，64位的入站八位组计数。	Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/interface/service-policy-names/service-policy-instance/statistics/class-stats/child-policy/class-stats/general-stats/pre-patched-bytes  Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/output/service-policy-names/service-policy-instance/statistics/class-stats/child-policy/class-stats/general-stats/pre-patched-bytes

## CISCO-ENHANCED-MEMPOOL-MIB

下表表示OID名称和编号，以及要在与内存使用情况相关的模型驱动遥测传感器组上设置的相应XPATH。

OID名称	OID编号	OID说明	XPATH
cempMemPoolUsed	1.3.6.1.4.1.9.9.221.1.1.1.1.7	指示物理实体上的应用程序当前正在使用的内存池中的字节数。	Cisco-IOS-XR-nto-misc-oper:memory-summary/nodes/node/sur-y
cempMemPoolHCUsed	1.3.6.1.4.1.9.9.221.1.1.1.1.18	指示物理实体上的应用程序当前正在使用的内存池中的字节数。此对象是cempMemPoolUsed的64位版本。	Cisco-IOS-XR-nto-misc-oper:memory-summary/nodes/node/det-al-used
cempMemPoolHCFree	1.3.6.1.4.1.9.9.221.1.1.1.1.20	指示当前在物理实体上未使用的内存池的字节数。此对象是cempMemPoolFree的64位版本。	Cisco-IOS-XR-nto-misc-oper:memory-summary/nodes/node/det-e-physical-memory

## CISCO-ENTITY-FRU-CONTROL-MIB

下表表示OID名称和编号以及要在模型驱动的遥测传感器组上设置的相应XPATH，这些传感器组与监控系统上的现场可更换单元相关。

OID名称	OID编号	OID说明	XPATH
cefcFRUPowerOperStatus	1.3.6.1.4.1.9.9.117.1.1.2.1.2	运行FRU电源状态。	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/fru-info/power-operational-state
cefcFRUPowerAdminStatus	1.3.6.1.4.1.9.9.117.1.1.2.1.1	管理性所需的FRU电源状态。	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/fru-info/power-administrative-state
cefcModuleStatusLastChangeTime	1.3.6.1.4.1.9.9.117.1.2.1.1.4	更改 cefcModuleOperStatus时的sysUpTime值。	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/fru-info/last-operational-state-change-time
cefcModuleUpTime	1.3.6.1.4.1.9.9.117.1.2.1.1.8	此对象提供自上次重新初始化以来模块的正常运行时间。此对象不持久；如果模块重置、重新启动、关闭，则启动时间从零开始。	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/fru-info/card-up-time
cefcModuleResetReason	1.3.6.1.4.1.9.9.117.1.2.1.1.3	此对象标识上次在模块上执行重置的原因。	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/fru-info/card-reset-reason
cefcModuleOperStatus	1.3.6.1.4.1.9.9.117.1.2.1.1.2	此对象显示模块的运行状态。	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/fru-info/card-operational-state
cefcModuleAdminStatus	1.3.6.1.4.1.9.9.117.1.2.1.1.1	此对象提供对模块的管理控制。	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/fru-info/card-administrative-state

## CISCO-ENTITY-SENSOR-MIB

下表表示OID名称和编号以及要在与节点上的传感器实体相关的模型驱动遥测传感器组上设置的相应XPATH。

OID名称	OID编号	OID说明	XPATH
entSensorValue	1.3.6.1.4.1.9.9.91.1.1.1.1.4	此变量报告传感器所看到的最新测量。要正确显示或解释此变量的值，您还必须知道entSensorType、entSensorScale和entSensorPrecision。但是，您可以将entSensorValue与entSensorThresholdTable中给定的阈值进行比较，而无需任何语义知识。	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/env-sensor-info/value
entSensorThresholdEvaluation	1.3.6.1.4.1.9.9.91.1.2.1.1.5	此变量表示最近对阈值的评估的结果。如果阈值条件为true，则	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/threshold-evaluation

entSensorThresholdEvaluation为true(1)。如果阈值条件为false，则entSensorThresholdEvaluation为false(2)。阈值按entSensorValueUpdateRate指示的速率进行评估。

## CISCO-FLASH-MIB

下表表示OID名称和编号以及要在与系统闪存相关的模型驱动遥测传感器组上设置的相应XPATH。

OID名称	OID编号	OID说明	XPATH
ciscoFlashPartitionName	1.3.6.1.4.1.9.9.10.1.1.4.1.1.10	闪存分区名称，用于指代系统的分区。这可以是AAAAAAnn形式的任何字母数字字符串，其中A表示可选的字母字符和数字字符。任何数字字符必须始终构成字符串的尾部部分。系统将删除字母字符并使用数字部分映射到分区索引。根据此名称，闪存操作会定向到设备分区。系统具有默认分区概念。这将是设备中的第一个分区。当未指定分区名称时，系统会将操作定向到默认分区。因此，分区名称是必填项，除非在默认分区上执行操作，或者设备只有一个分区（未分区）。	Cisco-IOS-XR-shellutil-filesystem/oper:filesystem/node/filesystem/type
ciscoFlashPartitionSizeExtended	1.3.6.1.4.1.9.9.10.1.1.4.1.1.13	闪存分区大小。它应是ciscoFlashDeviceMinPartitionSize的整数倍。如果有单个分区，此大小将等于ciscoFlashDeviceSize。此对象是64位版本的ciscoFlashPartitionSize	Cisco-IOS-XR-shellutil-filesystem/oper:filesystem/node/filesystem/size
ciscoFlashPartitionFreeSpaceExtended	1.3.6.1.4.1.9.9.10.1.1.4.1.1.14	闪存分区中的可用空间。请注意，闪存中文件的实际大小包括代表文件系统文件头的少量开销。某些文件系统在计算可用空间时可能还会考虑分区或设备报头开销。可用空间的计算方式是：总分区大小减去所有现有文件（有效/无效/删除的文件，包括每个文件的文件头）的大小，减少任何分区头的大小，减少要复制的下一个文件的头大小。简而言之，此对象将提供可复制的最大文件的大小。管理实体不需要知	Cisco-IOS-XR-shellutil-filesystem/oper:filesystem/node/filesystem/free

道或使用任何开销，例如文件和分区报头长度，因为此类开销可能因文件系统而异。闪存中删除的文件不会释放空间。为了回收文件所占用的空间，可能必须擦除分区。此对象是64位版本的 ciscoFlashPartitionFreeSpace

## CISCO-PROCESS-MIB

下表表示OID名称和编号，以及要在模型驱动遥测传感器组相关的CPU使用率和进程资源分配上设置的相应XPATH。

OID名称	OID编号	OID说明	XPATH
cpmCPUTotal1minRev	1.3.6.1.4.1.9.9.109.1.1.1.1.7	过去1分钟内的总CPU占线百分比。此对象弃用了对象 cpmCPUTotal1min，并将值范围增加到(0..100)。	Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization/total-cpu-minute
cpmCPUTotal5minRev	1.3.6.1.4.1.9.9.109.1.1.1.1.8	过去5分钟内的总CPU占线百分比。此对象弃用了对象 cpmCPUTotal5min，并将值范围增加到(0..100)。	Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization/total-cpu-5min
cpmCPUTotal15minRev	1.3.6.1.4.1.9.9.109.1.1.1.1.31	过去15分钟内的总CPU占线百分比。此对象弃用了对象 cpmCPUTotal15min，并将值范围增加到(0..100)。	Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization/total-cpu-15min
cpmProcessName	1.3.6.1.4.1.9.9.109.1.2.1.1.2	与此进程关联的名称。如果名称超过32个字符，则会将其截断为前31个字符，并且“*”将作为最后一个字符添加，以暗示这是截断的进程名称。	Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization/process-cpu/process-name
cpmProcessTextSegmentSize	1.3.6.1.4.1.9.9.109.1.2.3.1.15	这表示进程及其所有共享对象的文本内存。	Cisco-IOS-XR-process-memory/nodes/node-id/process-ids/process-id-seg-size
cpmProcessDynamicMemorySize	1.3.6.1.4.1.9.9.109.1.2.3.1.18	这表示进程使用的动态内存量。	Cisco-IOS-XR-process-memory/nodes/node-id/process-ids/process-id-limit
cpmProcessDataSegmentSize	1.3.6.1.4.1.9.9.109.1.2.3.1.16	这表示进程的数据段及其所有共享对象。	Cisco-IOS-XR-process-memory/nodes/node-id/process-ids/process-id-data-seg-size

			cess-ids/process-id seg-size
cpmProcExtMemAllocatedRev	1.3.6.1.4.1.9.9.109.1.2.3.1.1	此进程已从系统接收的所有动态分配内存的总和。这包括可能已返回的内存。释放的内存总和由 cpmProcExtMemFreedRev 提供。此对象弃用 cpmProcExtMemAllocated。	Cisco-IOS-XR-processes-memory/nodes/nod cess-ids/process-id
cpmProcExtMemFreedRev	1.3.6.1.4.1.9.9.109.1.2.3.1.2	此进程返回给系统的所有内存的总和。此对象弃用 cpmProcExtMemFreed。	Cisco-IOS-XR-processes-memory/nodes/nod cess-ids/process-id

## ENTITY-MIB

下表表示要在系统上模型驱动的遥测传感器组相关物理实体上设置的OID名称和编号以及相应的XPATH。

OID名称	OID编号	OID说明	XPATH
entPhysicalName	1.3.6.1.2.1.47.1.1.1.1.7	物理实体的文本名称。此对象的值应是由本地设备分配的组件名称，并且应适合用于在设备的“控制台”中输入的命令。这可能是文本名称，如“console”或简单组件编号（如端口或模块编号），如“1”，具体取决于设备的物理组件命名语法。如果没有本地名称，或者此对象不适用，则此对象包含零长度字符串。请注意，如果控制台接口不区分两个物理实体（如插槽1和插槽1中的卡），则两个物理实体的 entPhysicalName 值将相同。	Cisco-IOS-XR-snmp-entit mib-oper:entity-physical-index
entLogicalDescr	1.3.6.1.2.1.47.1.2.1.1.2	逻辑实体的文本说明。此对象应包含一个字符串，用于标识逻辑实体的制造商名称，并应设置为逻辑实体每个版本的不同值。	Cisco-IOS-XR-snmp-agent-oper:snmp/information-name/
entPhysicalDescr	1.3.6.1.2.1.47.1.1.1.1.2	物理实体的文本说明。此对象应包含一个字符串，该字符串用于标识物理实体的制造商名称，并应设置为物理实体的每个版本或型号的不同值。“包含”此物理实体的物理实体的 entPhysicalIndex 值。值为零表示此物理实体不包含在任何其他物理实体中。请注意，“遏制”关系集定义了严格的层次结构；即不允许递归。如果物理实体由多个物理实体	Cisco-IOS-XR-snmp-agent-oper:snmp/Cisco-IOS-XR-snmp-entit mib-oper:entity-mib/entity-physical-indexes/
entPhysicalConnedIn	1.3.6.1.2.1.47.1.1.1.1.4	“包含”此物理实体的物理实体的 entPhysicalIndex 值。值为零表示此物理实体不包含在任何其他物理实体中。请注意，“遏制”关系集定义了严格的层次结构；即不允许递归。如果物理实体由多个物理实体	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/inv-basic-bag/unique-id

entPhysicalClass	1.3.6.1.2.1.47.1.1.1.1.5	<p>(例如, 双宽度模块) 包含, 则此对象应标识具有最低 entPhysicalIndex 值的包含实体。</p> <p>物理实体的一般硬件类型的指示。代理应将此对象设置为最准确地指示物理实体的一般类的标准枚举值, 或主类 (如果有多个)。如果此实体不存在适当的标准注册标识符, 则返回值“other(1)”。如果此代理未知该值, 则返回值“unknown(2)”。</p>	Cisco-IOS-XR-invmgr-oper:inventory/entities
entPhysicalHardwareRev	1.3.6.1.2.1.47.1.1.1.1.8	<p>物理实体的供应商特定硬件修订字符串。首选值是实际打印在组件本身 (如果存在) 上的硬件修订版标识符。请注意, 如果修订信息以不可打印 (例如, 二进制) 格式在内部存储, 则代理必须以特定于实现的方式将这些信息转换为可打印格式。如果没有特定硬件修订字符串与物理组件关联, 或者代理不知道此信息, 则此对象将包含零长度字符串。</p>	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/inv-basic-bag/hardware-revision
entPhysicalFirmwareRev	1.3.6.1.2.1.47.1.1.1.1.9	<p>物理实体的供应商特定固件修订字符串。请注意, 如果修订信息以不可打印 (例如, 二进制) 格式在内部存储, 则代理必须以特定于实现的方式将这些信息转换为可打印格式。如果没有特定固件程序与物理组件关联, 或者代理不知道此信息, 则此对象将包含零长度字符串。</p>	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/inv-basic-bag/firmware-revision
entPhysicalSoftwareRev	1.3.6.1.2.1.47.1.1.1.1.10	<p>物理实体的供应商特定软件修订字符串。请注意, 如果修订信息以不可打印 (例如, 二进制) 格式在内部存储, 则代理必须以特定于实现的方式将这些信息转换为可打印格式。如果没有特定软件程序与物理组件关联, 或者代理不知道此信息, 则此对象将包含零长度字符串。</p>	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/inv-basic-bag/software-revision
entPhysicalSerialNum	1.3.6.1.2.1.47.1.1.1.1.11	<p>物理实体的供应商特定序列号字符串。首选值是实际打印在组件本身 (如果存在) 上的序列号字符串。在物理实体的首次实例化时, 如果座席可以获得此信息, 则与该实体关联的 entPhysicalSerialNum 的值将设置为正确的供应商分配的序</p>	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/inv-basic-bag/serial-number



列号。如果序列号未知或不  
存在，则

entPhysicalSerialNum将设置  
为零长度字符串。请注意，能  
够正确识别所有已安装物理实  
体的序列号的实施不需要提供  
对entPhysicalSerialNum对象  
的写访问。为

entPhysicalSerialNum字符串  
提供非易失性存储的代理不需  
要为此对象实施写访问。并非  
每个物理组件都会有序列号  
，甚至需要一个。其

entPhysicalsFRU对象的关联  
值等于“false(2)”（例如中继器  
模块中的中继器端口）的物理  
实体不需要其自己的唯一序列  
号。代理不必为此类实体提供  
写访问，并且可返回零长度字  
符串。如果为

entPhysicalSerialNum实例实  
施写访问，并且将值写入实例  
，则只要该实体保持实例化状  
态，代理就必须保留与同一物  
理实体关联的

entPhysicalSerialNum实例中  
提供的值。这包括在网络管理  
系统的所有重新初始化/重新启  
动过程中进行实例化，包括导  
致物理实体的

entPhysicalIndex值更改的实  
例化。

此物理组件的制造商的名称。  
首选值是实际打印在组件本身  
（如果有）上的制造商名称字  
符串。请注意

，entPhysicalModelName、  
entPhysicalFirmwareRev、  
entPhysicalSoftwareRev和  
entPhysicalSerialNum对象实  
例之间的比较仅在具有相同  
entPhysicalMfgName值的  
entPhysicalEntries中有意义。  
如果代理未知与物理组件关联  
的制造商名称字符串，则此对  
象将包含零长度字符串。

与此物理组件关联的供应商特  
定型号名称标识符字符串。首  
选值是客户可见的部件号，可  
以打印在组件本身上。如果代  
理未知与物理组件关联的模型  
名称字符串，则此对象将包含  
零长度字符串。

entPhysicalMfgName 1.3.6.1.2.1.47.1.1.1.1.12

entPhysicalModelName 1.3.6.1.2.1.47.1.1.1.1.13

Cisco-IOS-XR-invmgr  
oper:inventory/entities  
y/attributes/inv-basic-  
bag/manufacturer-nar

Cisco-IOS-XR-invmgr  
oper:inventory/entities  
y/attributes/inv-basic-  
bag/model-name

## IF-MIB

下表表示OID名称和编号，以及要在与接口特征和计数器相关的模型驱动遥测传感器组上设置的相应XPath。

OID名称	OID编号	OID说明	XPATH
ifMtu	1.3.6.1.2.1.2.2.1.4	接口上可以发送/接收的最大数据包的大小，以二进制八位数指定。对于用于传输网络数据报的接口，这是接口上可以发送的最大网络数据报的大小。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/mtu
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	接口的协议子层地址。例如，对于802.x接口，此对象通常包含MAC地址。接口的介质特定MIB必须定义此对象的位和字节顺序以及值的格式。对于没有此地址的接口（例如串行线路），此对象应包含长度为零的二进制八位数字符串。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface-ty
ifType	1.3.6.1.2.1.2.2.1.3	接口类型。ifType的其他值由互联网编号指派机构(IANA)通过更新IANAifType文本约定的语法来分配。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface-ty
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	高层协议请求传输且未寻址到此子层组播或广播地址的数据包总数，包括已丢弃或未发送的数据包。此计数器的值在重新初始化管理系统时以及在ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/packets-sent
ifHCOutUcastPkts	1.3.6.1.2.1.31.1.1.1.11	高层协议请求传输且未寻址到此子层组播或广播地址的数据包总数，包括已丢弃或未发送的数据包。此对象是ifOutUcastPkts的64位版本。此计数器的值在重新初始化管理系统时以及在ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/packets-sent
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	由此子层传送到更高（子）层且未寻址到此子层的组播或广播地址的数据包数。此计数器的值在重新初始化管理系统时以及在ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/packets-received
ifHCInUcastPkts	1.3.6.1.2.1.31.1.1.1.7	由此子层传送到更高（子）层且未寻址到此子层的组播或广播地址的数据包数。此对象是ifInUcastPkts的64位版本。此	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface-

		计数器的值在重新初始化管理系统时以及在 ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。	stats/packets-received
ifOutErrors	1.3.6.1.2.1.2.2.1.20	对于面向数据包的接口，是指由于错误而无法传输的出站数据包数。对于面向字符或固定长度的接口，是指由于错误而无法传输的出站传输单元数。此计数器的值在重新初始化管理系统时以及在 ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/output-errors
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	选择丢弃的出站数据包数，即使未检测到任何错误以阻止其传输。丢弃这个信息包的一个可能原因是信息包能自由释放缓冲空间。此计数器的值在重新初始化管理系统时以及在 ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/output-drops
ifOutMulticastPkts	1.3.6.1.2.1.31.1.1.1.4	较高层协议请求传输的、发往此子层组播地址的数据包总数，包括被丢弃或未发送的数据包。对于MAC层协议，这包括组地址和功能地址。此计数器的值在重新初始化管理系统时以及在 ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/multicast-packets
ifHCOutMulticastPkts	1.3.6.1.2.1.31.1.1.1.12	较高层协议请求传输的、发往此子层组播地址的数据包总数，包括被丢弃或未发送的数据包。对于MAC层协议，这包括组地址和功能地址。此对象是ifOutMulticastPkts的64位版本。此计数器的值在重新初始化管理系统时以及在 ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/multicast-packets
ifInMulticastPkts	1.3.6.1.2.1.31.1.1.1.2	由此子层传送到更高（子）层的数据包数，这些数据包被寻址到此子层的组播地址。对于MAC层协议，这包括组地址和功能地址。此计数器的值在重新初始化管理系统时以及在 ifCounterDiscontinuationTime值指示的其他时间可能发生不	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/multicast-packets received

ifHCInMulticastPkts	1.3.6.1.2.1.31.1.1.1.8	<p>连续。</p> <p>由此子层传送到更高（子）层的数据包数，这些数据包被寻址到此子层的组播地址。对于MAC层协议，这包括组地址和功能地址。此对象是ifInMulticastPkts的64位版本。此计数器的值在重新初始化管理系统时以及在ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。</p> <p>对于面向数据包的接口，是指包含错误的入站数据包数量，这些错误导致无法将它们传送到更高层协议。对于面向字符或固定长度的接口，包含错误的入站传输单元的数量，这些错误导致无法将传输到更高层协议。此计数器的值在重新初始化管理系统时以及在ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。</p>	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/multicast-packets received
ifInErrors	1.3.6.1.2.1.2.2.1.14	<p>选择被丢弃的入站数据包数，即使未检测到任何错误以阻止其传输到更高层协议。丢弃这个信息包的一个可能原因是信息包能自由释放缓冲空间。此计数器的值在重新初始化管理系统时以及在ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。</p>	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/input-errors
ifInDiscards	1.3.6.1.2.1.2.2.1.13	<p>从接口传输的二进制八位数总数，包括成帧字符。此计数器的值在重新初始化管理系统时以及在ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。</p>	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/input-drops
ifOutOctets	1.3.6.1.2.1.2.2.1.16	<p>从接口传输的二进制八位数总数，包括成帧字符。此对象是ifOutOctets的64位版本。此计数器的值在重新初始化管理系统时以及在ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。</p>	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/bytes-sent
ifHCOctets	1.3.6.1.2.1.31.1.1.1.10	<p>接口上接收的二进制八位数总数，包括成帧字符。此计数器的值在重新初始化管理系统时以及在ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。</p>	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/bytes-sent
ifInOctets	1.3.6.1.2.1.2.2.1.10	<p>接口上接收的二进制八位数总数，包括成帧字符。此计数器的值在重新初始化管理系统时以及在ifCounterDiscontinuationTime值指示的其他时间可能发生不连续。</p>	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface-

			ifCounterDiscontinuationTime 值指示的其他时间可能发生不连续。	stats/bytes-received
ifHCInOctets	1.3.6.1.2.1.31.1.1.1.6		接口上接收的二进制八位数总数，包括成帧字符。此对象是 ifInOctets 的 64 位版本。此计数器的值在重新初始化管理系统时以及在 ifCounterDiscontinuationTime 值指示的其他时间可能发生不连续。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/bytes-received
ifOutBroadcastPkts	1.3.6.1.2.1.31.1.1.1.5		较高层协议请求传输的、发往此子层广播地址的数据包总数，包括被丢弃或未发送的数据包。此计数器的值在重新初始化管理系统时以及在 ifCounterDiscontinuationTime 值指示的其他时间可能发生不连续。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/broadcast-packete sent
ifHCOutBroadcastPkts	1.3.6.1.2.1.31.1.1.1.13		较高层协议请求传输的、发往此子层广播地址的数据包总数，包括被丢弃或未发送的数据包。此对象是 ifOutBroadcastPkts 的 64 位版本。此计数器的值在重新初始化管理系统时以及在 ifCounterDiscontinuationTime 值指示的其他时间可能发生不连续。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/broadcast-packete sent
ifInBroadcastPkts	1.3.6.1.2.1.31.1.1.1.3		由此子层传送到更高（子）层的数据包数，这些数据包被寻址到此子层的广播地址。此计数器的值在重新初始化管理系统时以及在 ifCounterDiscontinuationTime 值指示的其他时间可能发生不连续。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/broadcast-packete received
ifHCInBroadcastPkts	1.3.6.1.2.1.31.1.1.1.9		由此子层传送到更高（子）层的数据包数，这些数据包被寻址到此子层的广播地址。此对象是 ifInBroadcastPkts 的 64 位版本。此计数器的值在重新初始化管理系统时以及在 ifCounterDiscontinuationTime 值指示的其他时间可能发生不连续。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/broadcast-packete received
ifIndex	1.3.6.1.2.1.2.2.1.1		每个接口的唯一值，大于零。建议从 1 开始连续分配值。每个接口子层的值必须至少保持恒定，从实体网络管理系统的一次重新初始化到下一次重新初始化。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/if-index
ifDescr	1.3.6.1.2.1.2.2.1.2		包含接口信息的文本字符串。	Cisco-IOS-XR-pfi-im-c

		此字符串应包括制造商的名称、产品名称和接口硬件/软件版本。	oper:interfaces/interface-xr/interface/description
ifSpeed	1.3.6.1.2.1.2.2.1.5	接口当前带宽的估计值 (以比特/秒为单位)。对于带宽不变的接口或无法进行准确估计的接口,此对象应包含标称带宽。如果接口的带宽大于此对象报告的最大值,则此对象应报告其最大值 (4,294,967,295),并且如果必须使用HighSpeed报告接口的速度。对于没有带宽概念的子层,此对象应为零。接口的当前运行状态。测试(3)状态表示无法通过任何可操作的数据包。如果ifAdminStatus为down(2),则ifOperStatus应为down(2)。如果ifAdminStatus更改为up(1),则如果接口已准备好传输和接收网络流量	Cisco-IOS-XR-pfi-im-c oper:interfaces/interface-xr/interface/bandwidth
ifOperStatus	1.3.6.1.2.1.2.2.1.8	, ifOperStatus应更改为up(1);如果接口等待外部操作 (例如等待传入连接的串行线路),则应更改为休眠(5);当且仅当存在故障阻止其进入up(1)状态时,它应保持down(2)状态;如果接口缺少 (通常是硬件)组件,则应保持notPresent(6)状态。接口的所需状态。测试(3)状态表示无法通过任何可操作的数据包。当托管系统初始化时	Cisco-IOS-XR-pfi-im-c oper:interfaces/interface-non-dynamics/interface-non-dynamic/oper-stat
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	,所有接口都以ifAdminStatus开始,处于down(2)状态。由于显式管理操作或受管系统保留的每个配置信息,ifAdminStatus随后更改为up(1)或testing(3)状态(或保持down(2)状态)。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interface-non-dynamics/interface-non-dynamic/admin-st
ifName	1.3.6.1.2.1.31.1.1.1.1	接口的文本名称。此对象的值应是本地设备分配的接口名称,并且应适合用于在设备的“控制台”中输入的命令。这可能是文本名称 (如“le0”)或简单端口号 (如“1”),具体取决于设备的接口命名语法。如果ifTable中的多个条目一起表示由设备命名的单个接口,则每个接口的值将相同,即ifName。请注意,对于响应与某些其他 (代理)设备上的接	Cisco-IOS-XR-pfi-im-c oper:interfaces/interface- driefts/interface- brief/interface-name

口相关的SNMP查询的代理，此类接口的ifName值是其代理设备的本地名称。如果没有本地名称，或者此对象不适用，则此对象包含零长度字符串。

以每秒1,000,000位为单位的接口当前带宽的估计值。如果此对象报告的值为“n”，则接口速度介于“n-500,000”到“n+499,999”之间。对于带宽不变的接口或无法进行准确估计的接口，此对象应包含标称带宽。对于没有带宽概念的子层，此对象应为零。

Cisco-IOS-XR-pfi-im-c  
oper:interfaces/interfac  
crianks/interface-  
brief/bandwidth64-bit

ifHighSpeed

1.3.6.1.2.1.31.1.1.1.15

## IP-MIB

下表表示OID名称和编号，以及要在与互联网协议(IP)统计信息和操作值相关的模型驱动遥测传感器组上设置的相应XPath。

OID名称	OID编号	OID说明	XPATH
icmplnDestUnreachs	1.3.6.1.2.1.5.3	收到的ICMP目标无法到达消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmplnParmProbs	1.3.6.1.2.1.5.5	收到的ICMP参数问题消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmplnSrcQuenchs	1.3.6.1.2.1.5.6	收到的ICMP源抑制消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmplnEchos	1.3.6.1.2.1.5.8	收到的ICMP回应（请求）消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmplnEchoReps	1.3.6.1.2.1.5.9	收到的ICMP应答消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmplnTimestamps	1.3.6.1.2.1.5.10	收到的ICMP时间戳（请求）消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmplnAddrMasks	1.3.6.1.2.1.5.12	收到的ICMP地址掩码请求消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmplnAddrMaskReps	1.3.6.1.2.1.5.13	收到的ICMP地址掩码应答消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st

icmpOutMsgs	1.3.6.1.2.1.5.14	此实体尝试发送的ICMP消息总数。请注意，此计数器包括icmpOutErrors计数的所有计数器。	s/traffic/icmp-stats Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutDestUnreachs	1.3.6.1.2.1.5.16	发送的ICMP目标无法到达消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutTimeExcds	1.3.6.1.2.1.5.17	发送的ICMP超时消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutParmProbs	1.3.6.1.2.1.5.18	发送的ICMP参数问题消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutSrcQuenchs	1.3.6.1.2.1.5.19	发送的ICMP源抑制消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutRedirects	1.3.6.1.2.1.5.20	发送的ICMP重定向消息数。对于主机，此对象始终为零，因为主机不发送重定向。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutEchos	1.3.6.1.2.1.5.21	发送的ICMP回应（请求）消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutEchoReps	1.3.6.1.2.1.5.22	发送的ICMP应答消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutTimestamps	1.3.6.1.2.1.5.23	发送的ICMP时间戳（请求）消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutAddrMasks	1.3.6.1.2.1.5.25	发送的ICMP地址掩码请求消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutAddrMaskReps	1.3.6.1.2.1.5.26	发送的ICMP地址掩码应答消息数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
ipAdEntIfIndex	1.3.6.1.2.1.4.20.1.2	唯一标识此条目适用的接口的索引值。由此索引的特定值标识的接口与由RFC 1573的ifIndex的相同值标识的接口相同。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/
ipAdEntAddr	1.3.6.1.2.1.4.20.1.1	此条目的编址信息所涉及的IP地址。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/interfaces/inte



ipAdEntNetMask	1.3.6.1.2.1.4.20.1.3	与此条目的IP地址关联的子网掩码。掩码的值是IP地址，所有网络位设置为1，所有主机位设置为0。	vrf/vrf/detail/primary-address Cisco-IOS-XR-ipv4-io- oper:ipv4- network/interfaces/inte vrf/vrf/detail/prefix-len
ipAdEntBcastAddr	1.3.6.1.2.1.4.20.1.4	IP广播地址中用于在与此条目的IP地址关联的（逻辑）接口上发送数据报的最低有效位的值。例如，当使用Internet标准全1广播地址时，值将为1。此值同时适用于此（逻辑）接口上的实体使用的子网和网络广播地址。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/interfaces/inte vrf/vrf/detail/direct- broadcast
ipNetToMediaPhysAddress	1.3.6.1.2.1.4.22.1.2	介质相关的“物理”地址。	Cisco-IOS-XR-ipv4-arp oper:arp/nodes/node/e /entry/hardware-address

## IPMIB-COMMON

下表表示OID名称和编号，以及要在与IP统计信息相关的模型驱动遥测传感器组上设置的相应XPATH。

OID名称	OID编号	OID说明	XPATH
ipIfStatsHCOutTransmits	1.3.6.1.2.1.4.31.3.1.31	此实体提供给下层以供传输的IP数据报总数。此对象计数与ipIfStatsOutTransmits相同的数 据报，但允许更大的值。此计 数器的值在重新初始化管理系 统时可能发生中断，在 ipIfStatsDiscontinuationTime 值指示的其他时间也会发生中 断。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/s cs/traffic/ipv4-stats/pa forwarded
ipIfStatsInReceives	1.3.6.1.2.1.4.31.3.1.3	接收的输入IP数据报总数，包 括错误接收的数据报。此计数 器的值在重新初始化管理系 统时可能发生中断，在 ipIfStatsDiscontinuationTime 值指示的其他时间也会发生中 断。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/s cs/traffic/ipv4-stats/inp packets
ipIfStatsHCInReceives	1.3.6.1.2.1.4.31.3.1.4	接收的输入IP数据报总数，包 括错误接收的数据报。此对象 计数与ipIfStatsInReceives相 同的数据报，但允许更大的值。 此计数器的值在重新初始化管 理系统时可能发生中断，在 ipIfStatsDiscontinuationTime 值指示的其他时间也会发生中 断。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/s cs/traffic/ipv4-stats/inp packets

## LLDP-MIB

下表表示OID名称和编号以及要在与监控节点上的链路层发现协议(LLDP)操作数据相关的模型驱动遥测传感器组上设置的相应XPath。

OID名称	OID编号	OID说明	XPATH
IldpLocPortId	1.0.8802.1.1.2.1.3.7.1.3	用于标识与本地系统中给定端口关联的端口组件的字符串值。	Cisco-IOS-XR-etherne lldp- oper:Ildp/nodes/node/ bors/device/Ildp- neighbor/port-id-detail Cisco-IOS-XR-etherne lldp-
IldpLocPortIdSubtype	1.0.8802.1.1.2.1.3.7.1.2	关联的“IldpLocPortId”对象中使用的端口标识符编码类型。	oper:Ildp/nodes/node/ bors/device/Ildp- neighbor/mib/port-id-s type Cisco-IOS-XR-etherne lldp-
IldpLocChassisIdSubtype	1.0.8802.1.1.2.1.3.1	用于标识与本地系统关联的机箱的编码类型。	oper:Ildp/nodes/node/ bors/device/Ildp- neighbor/mib/chassis- sub-type Cisco-IOS-XR-etherne lldp-
IldpLocSysName	1.0.8802.1.1.2.1.3.3	用于标识本地系统的系统名称的字符串值。如果本地代理支持IETF RFC 3418，则IldpLocSysName对象应具有与sysName对象相同的值。	Cisco-IOS-XR-etherne lldp- oper:Ildp/nodes/node/ bors/device/Ildp- neighbor/detail/system name Cisco-IOS-XR-etherne lldp-
IldpRemSysName	1.0.8802.1.1.2.1.4.1.1.9	用于标识远程系统的系统名称的字符串值。	oper:Ildp/nodes/node/ bors/device/Ildp- neighbor/detail/system name Cisco-IOS-XR-etherne lldp-
IldpRemChassisId	1.0.8802.1.1.2.1.4.1.1.5	用于标识与远程系统关联的机箱组件的字符串值。	oper:Ildp/nodes/node/ bors/device/Ildp- neighbor/chassis-id Cisco-IOS-XR-etherne lldp-
IldpRemChassisIdSubtype	1.0.8802.1.1.2.1.4.1.1.4	用于标识与远程系统关联的机箱的编码类型。	oper:Ildp/nodes/node/ bors/device/Ildp-neigh Cisco-IOS-XR-etherne lldp-
IldpRemPortIdSubtype	1.0.8802.1.1.2.1.4.1.1.6	关联的“IldpRemPortId”对象中使用的端口标识符编码类型。	oper:Ildp/nodes/node/ bors/device/Ildp-neigh Cisco-IOS-XR-etherne lldp-
IldpRemPortId	1.0.8802.1.1.2.1.4.1.1.7	用于标识与远程系统关联的端口组件的字符串值。	oper:Ildp/nodes/node/ bors/device/Ildp-neigh Cisco-IOS-XR-etherne lldp-
IldpLocChassisId	1.0.8802.1.1.2.1.3.2	用于标识与本地系统关联的机箱组件的字符串值。	oper:Ildp/nodes/node/ bors/device/Ildp-neigh

## MPLS-TE-STD-MIB

下表表示OID名称和编号以及要在受管设备上与多协议标签交换(MPLS)流量工程操作值相关的模型驱动遥测传感器组上设置的相应XPath。

OID名称	OID编号	OID说明	XPATH
mplsTunnelName	1.3.6.1.2.1.10.166.3.2.2.1.5	分配给隧道的规范名称。此名称可用于指代LSR控制台端口上的隧道。如果mplsTunnelsIf设置为true，则与此隧道对应的接口的ifName的值应等于mplsTunnelName。另请参阅RFC 2863中对ifName的说明。	Cisco-IOS-XR-mpls-transport-neighbor/chassis-id/oper:mpls-te/p2p-p2rptunnel/tunnel-heads/tunnel-head/tunnel-name
mplsTunnelDescr	1.3.6.1.2.1.10.166.3.2.2.1.6	包含有关隧道信息的文本字符串。如果没有说明，则此对象包含零长度字符串。此对象可能不由MPLS信令协议发出信号，因此，传输时此对象的值和出口LSR可能自动生成或不	openconfig-network-instance/network-instances/network-instance/mpls/lsp/connected-path/tunnels/tunnel/sdescription
mplsTunnelPerfHCPackets	1.3.6.1.2.1.10.166.3.2.9.1.2	隧道转发的数据包数的高容量计数器。	openconfig-network-instance/network-instance/mpls/lsp/connected-path/tunnels/tunnel/scounters/packets
mplsTunnelPerfHCBytes	1.3.6.1.2.1.10.166.3.2.9.1.5	隧道转发的字节数的高容量计数器。	openconfig-network-instance/network-instance/mpls/lsp/connected-path/tunnels/tunnel/scounters/bytes
mplsTunnelHopIpAddr	1.3.6.1.2.1.10.166.3.2.4.1.5	此隧道跳的隧道跳数地址。此地址的类型由相应mplsTunnelHopAddrType的值确定。如果对应的mplsTunnelHopRowStatus对象的值为“active”，则无法更改此对象的值。	Cisco-IOS-XR-mpls-transport-neighbor/chassis-id/oper:mpls-te/p2p-p2rptunnel/tunnel-heads/tunnel-head/destination/destination-address

## RFC2465-MIB

下表表示OID名称和编号，以及要在与IPv6全局值相关的模型驱动遥测传感器组上设置的相应XPath。

OID名称	OID编号	OID说明	XPATH
ipv6addrPfxLength	1.3.6.1.2.1.55.1.8.1.2	与此条目的IPv6地址关联的前缀长度（以位为单位）。	Cisco-IOS-XR-ipv6-management:ipv6-network/nodes/node/interface-data/vrfs/vrf/brief/address-fix-length
ipv6AddrAnycastFlag	1.3.6.1.2.1.55.1.8.1.4	如果此地址是任播地址，则此对象具有值“true(1)”，否则为值“false(2)”。	Cisco-IOS-XR-ipv6-management:ipv6-network/nodes/node/interface-data/vrfs/vrf/brief/address-anycast

## SNMP-MIB

下表表示OID名称和编号，以及要在与SNMP代理本身相关的模型驱动遥测传感器组上设置的相应XPATH（如果可用）。

OID名称	OID编号	OID说明	XPATH
sysUpTime	1.3.6.1.2.1.1.3	表示系统正常运行时间的字符串	Cisco-IOS-XR-snmp-agent:snmp/information/system-up-time/
sysObjectID	1.3.6.1.2.1.2.0	表示系统OID的字符串	Cisco-IOS-XR-snmp-agent:snmp/information/system-oid/
sysDescr	1.3.6.1.2.1.1.1	表示系统说明的字符串	Cisco-IOS-XR-snmp-agent:snmp/information/system-descr

## TCP-MIB

下表表示OID名称和编号，以及要在与TCP特定计数器相关的模型驱动遥测传感器组上设置的相应XPATH。

OID名称	OID编号	OID说明	XPATH
tcpInErrs	1.3.6.1.2.1.6.14	错误接收的数据段总数（例如，错误的TCP校验和）。	Cisco-IOS-XR-ip-tcp-ipv4-traffic:tcp-checksum-error-packets
tcpInSegs	1.3.6.1.2.1.6.10	接收的数据段总数，包括错误接收的数据段。此计数包括在当前建立的连接上收到的数据段。	Cisco-IOS-XR-ip-tcp-ipv4-traffic:tcp-input-packets
tcpOutSegs	1.3.6.1.2.1.6.11	发送的数据段总数，包括当前连接上的数据段，但不包括仅包含重新传输的二进制八位数的数据段。	Cisco-IOS-XR-ip-tcp-ipv4-traffic:tcp-output-packets

## UDP-MIB

下表表示OID名称和编号，以及要在与UDP特定计数器相关的模型驱动遥测传感器组上设置的相应XPATH。

OID名称	OID编号	OID说明	XPATH
udpOutDatagrams	1.3.6.1.2.1.7.4	从此实体发送的UDP数据报总数。	Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv4-traffic/udp-output-packets Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv6-traffic/udp-output-packets
udpNoPorts	1.3.6.1.2.1.7.2	在目的端口上没有应用的已接收UDP数据报的总数。	Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv4-traffic/udp-no-port-packets Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv6-traffic/udp-no-port-packets
udpInErrors	1.3.6.1.2.1.7.3	由于目的端口上缺少应用程序而无法传送的已接收UDP数据报数。	Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv4-traffic/udp-checks-error-packets Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv6-traffic/udp-checks-error-packets
udpInDatagrams	1.3.6.1.2.1.7.1	传送给UDP用户的UDP数据报总数。	Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv4-traffic/udp-input-packets Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv6-traffic/udp-input-packets

## SNMP陷阱迁移

SNMP陷阱是受管设备上的动态事件触发的消息。因此，这些消息的行为类似于我们之前介绍的EDT的概念。

在配置方面，MDT允许EDT采用相同的结构，具体取决于遥测收集器在拨入或拨出选择或功能方面的实施。

## 安全考虑

SNMPv2仅使用团体作为身份验证/授权机制。但是，SNMPv3，如前所述，可以使用凭证进行身份验证，使用AES加密模型保护信息。

在遥测方法中，IOS XR允许使用基于证书的gRPC/TLS技术来执行身份验证。这些证书可与中心信任点（例如CA服务器）一起使用。在建立信任关系的过程之后，所有遥测消息都在gRPC会话内发送，该会话使用TLS加密，实现SNMPv3的相同优势。