SNMP-Migration zu Telemetrie auf IOS XR

Inhalt

Einführung
SNMP
SNMP-Komponenten
SNMP-Manager
SNMP-Agent
SNMP MIB
SNMP-Betrieb
MIBs und RFCs
SNMP-Versionen
Yang-Modelle
OpenConfig-Modelle
Native Modelle
Telemetrie
Modellgesteuerte Telemetrie
Ereignisgesteuerte Telemetrie
Transport
TCP
gRPC
gNMI/gNOI
Kodierung
JSON
GPB-KV
GPB
MDT-Konfiguration in IOS XR
Wählmodus
Einwahlmodus
SNMP-Migration auf MDT
MIB-Migration in XPATH
BGP4-MIB
CISCO-BGP4-MIB
CISCO-KLASSE-BASED-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-COMMMON

LLDP-MIB
MPLS-TE-STD-MIB
RFC2465-MIB
SNMP-MIB
TCP-MIB
UDP-MIB
Migration von SNMP-Traps
Sicherheitsüberlegungen

Einführung

In diesem Artikel werden SNMP-Komponenten (Simple Network Management Protocol) vorgestellt, und es wird eine Korrelation zwischen aktuellen Implementierungen auf der Grundlage der SNMP-Überwachung in einem MDT-Ansatz (Model Driven Telemetry) bereitgestellt.

SNMP

SNMP ist ein Protokoll auf Anwendungsebene, das ein Nachrichtenformat für die Kommunikation zwischen SNMP-Managern und -Agenten bereitstellt. SNMP bietet ein standardisiertes Framework und eine gemeinsame Sprache, die zur Überwachung und Verwaltung von Geräten in einem Netzwerk verwendet wird.

SNMP-Komponenten

Das SNMP-Framework umfasst die folgenden Komponenten, die in den folgenden Abschnitten beschrieben werden:

- SNMP-Manager
- SNMP-Agent
- SNMP MIB

SNMP-Manager

Der SNMP-Manager ist ein System, das die Aktivitäten von Netzwerkhosts mithilfe von SNMP steuert und überwacht. Das häufigste Managementsystem ist ein Netzwerkmanagementsystem (NMS). Der Begriff NMS kann entweder auf ein dediziertes Gerät angewendet werden, das für die Netzwerkverwaltung verwendet wird, oder auf die Anwendungen, die auf einem solchen Gerät verwendet werden.

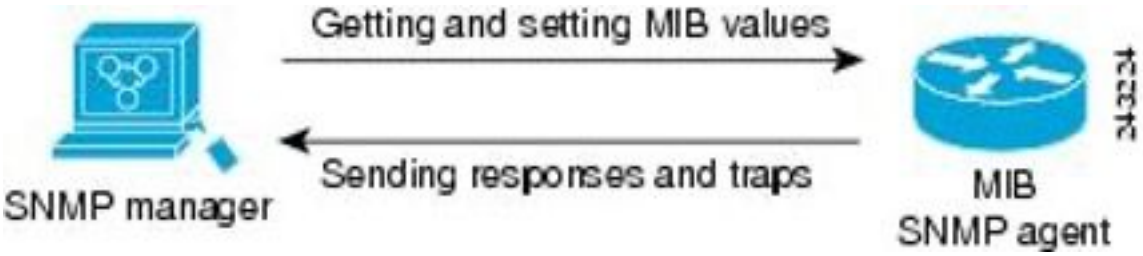
SNMP-Agent

Der SNMP-Agent ist die Softwarekomponente innerhalb eines verwalteten Geräts, das die Daten für das Gerät verwaltet und diese Daten bei Bedarf an das Management der Systeme meldet. Der Agent befindet sich auf dem Routing-Gerät (Router, Zugriffsserver oder Switch).

SNMP MIB

Ein SNMP-Agent enthält MIB-Variablen, deren Werte vom SNMP-Manager über 'Get' oder 'Set' angefordert oder geändert werden können. Ein Manager kann einen Wert von einem Agenten erhalten oder einen Wert in diesem Agenten speichern. Der Agent sammelt Daten von der SNMP MIB, dem Repository für Informationen über Geräteparameter und Netzwerkdaten. Der Agent kann auch auf Manager-Anfragen reagieren, um Daten abzurufen oder festzulegen.

In der folgenden Abbildung wird die Kommunikation zwischen dem SNMP-Manager und dem SNMP-Agent veranschaulicht. Ein Manager sendet Agent-Anfragen, um die SNMP-MIB-Werte abzurufen und festzulegen. Der Agent antwortet auf diese Anfragen. Unabhängig von dieser Interaktion kann der Agent dem Manager unaufgefordert Benachrichtigungen (Traps oder Informationen) senden, um den Manager über die Netzwerkbedingungen zu informieren.



SNMP-Betrieb

Die SNMP-Anwendungen führen die folgenden Vorgänge aus, um Daten abzurufen, SNMP-Objektvariablen zu ändern und Benachrichtigungen zu senden:

- SNMP abrufen
- SNMP-SET
- SNMP-Benachrichtigungen

SNMP abrufen

Der SNMP GET-Vorgang wird von einem NMS ausgeführt, um SNMP-Objektvariablen abzurufen. Es gibt drei Arten von GET-Vorgängen:

- GET - Ruft die genaue Objektinstanz vom SNMP-Agent ab.
- GETNEXT - Ruft die nächste Objektvariable ab, die ein lexikografischer Nachfolger der angegebenen Variablen ist.
- GETBULK - Ruft eine große Menge von Objektvariablen-Daten ab, ohne dass sich GETNEXT-Operationen wiederholen müssen.

SNMP-SET

Der SNMP SET-Vorgang wird von einem NMS ausgeführt, um den Wert einer Objektvariablen zu ändern.

SNMP-Benachrichtigungen

Eine wichtige Funktion von SNMP ist die Erzeugung unerwünschter Benachrichtigungen von einem SNMP-Agent.

Unerwünschte (asynchrone) Benachrichtigungen können als Traps oder Inform Requests

(Informs) generiert werden. Traps sind Meldungen, die den SNMP-Manager (Simple Network Management Protocol) auf den Zustand im Netzwerk hinweisen. Informs sind Traps, die eine Anfrage zur Bestätigung des Empfangs vom SNMP-Manager enthalten. Benachrichtigungen können auf unsachgemäße Benutzerauthentifizierung, Neustarts, das Schließen einer Verbindung, den Verlust der Verbindung zu einem Nachbargerät oder andere bedeutende Ereignisse hinweisen.

Traps sind weniger zuverlässig als Informationen, da der Empfänger beim Empfang einer Trap keine Bestätigung sendet. Der Absender weiß nicht, ob das Trap empfangen wurde. Ein SNMP-Manager, der eine Benachrichtigung empfängt, bestätigt die Nachricht mit einer SNMP Response Protocol Data Unit (PDU). Wenn der Absender nie eine Antwort erhält, kann die Benachrichtigung erneut gesendet werden. Infolgedessen erreichen Informationen mit höherer Wahrscheinlichkeit ihr beabsichtigtes Ziel.

Traps werden häufig bevorzugt, obwohl sie weniger zuverlässig sind, da Informationen mehr Ressourcen auf dem Gerät und im Netzwerk beanspruchen. Anders als bei einer Falle, die nach dem Senden verworfen wird, muss eine Benachrichtigung im Gedächtnis gehalten werden, bis eine Antwort eingeht oder die Anfrage abstürzt. Traps werden auch nur einmal gesendet, während eine Information möglicherweise mehrmals gesendet wird. Die erneuten Versuche erhöhen den Datenverkehr und tragen zu höheren Netzwerkkosten bei. Die Verwendung von Traps und Informationen erfordert einen Kompromiss zwischen Zuverlässigkeit und Ressourcen.

MIBs und RFCs

Management Information Base (MIB)-Module werden in der Regel in Request for Comments (RFC)-Dokumenten definiert, die an die internationale Normenorganisation Internet Engineering Task Force (IETF) übermittelt werden. RFCs werden von Einzelpersonen oder Gruppen geschrieben, die von der Internet Society und der Internet Community als Ganzes berücksichtigt werden sollen, in der Regel mit der Absicht, einen empfohlenen Internetstandard zu etablieren. Bevor Empfehlungen den RFC-Status erhalten, werden sie als Internetentwurf (I-D)-Dokumente veröffentlicht. RFCs, die zu empfohlenen Standards geworden sind, werden auch als Standard-Dokumente (STDs) bezeichnet. Informationen zum Standardisierungsprozess und zu den Aktivitäten der IETF finden Sie auf der Website der Internet Society unter http:/ / www.isoc.org. Sie können den vollständigen Text aller RFCs, I-Ds und STDs, auf die in der Cisco Dokumentation verwiesen wird, auf der IETF-Website unter http:/ / www.ietf.org lesen.

Die Cisco-Implementierung von SNMP verwendet die Definitionen der in RFC 1213 beschriebenen MIB-II-Variablen sowie die in RFC 1215 beschriebenen Definitionen der SNMP-Traps.

Cisco stellt für jedes System eigene private MIB-Erweiterungen bereit. Cisco Enterprise MIBs erfüllen die in den jeweiligen RFCs beschriebenen Richtlinien, sofern in der Dokumentation nicht anders angegeben. Die MIB-Moduldefinitionsdateien und die Liste der auf jeder Cisco Plattform unterstützten MIBs finden Sie auf der Cisco MIB-Website unter Cisco.com.

SNMP-Versionen

Derzeit unterstützen Cisco Geräte die folgenden SNMP-Versionen:

- SNMPv1 - Simple Network Management Protocol: ein vollständiger Internetstandard, definiert in RFC 1157. (RFC 1157 ersetzt die früheren Versionen, die als RFC 1067 und RFC 1098

veröffentlicht wurden.) Sicherheit basiert auf Community-Strings.
- SNMPv2c - Das Community String-basierte Administrations-Framework für SNMPv2. SNMPv2c (der "c" steht für "community") ist ein experimentelles Internetprotokoll, das in RFC 1901, RFC 1905 und RFC 1906 definiert ist. SNMPv2c ist eine Aktualisierung der Protokollvorgänge und Datentypen von SNMPv2p (SNMPv2 Classic) und verwendet das Community-basierte Sicherheitsmodell von SNMPv1.
- SNMPv3 - Version 3 des SNMP. SNMPv3 ist ein interoperables, standardbasiertes Protokoll, das in den RFCs 3413 bis 3415 definiert ist. SNMPv3 bietet sicheren Zugriff auf Geräte, indem Pakete über das Netzwerk authentifiziert und verschlüsselt werden.

SNMPv3 bietet folgende Sicherheitsfunktionen:

- Nachrichtenintegrität - Sicherstellen, dass ein Paket bei der Übertragung nicht manipuliert wurde.
- Authentication (Authentifizierung): Bestimmen, ob die Nachricht von einer gültigen Quelle stammt.
- Verschlüsselung - Scrambling des Inhalts eines Pakets, um zu verhindern, dass es von einer nicht autorisierten Quelle abgerufen wird.

SNMPv1 und SNMPv2c verwenden eine Community-basierte Form der Sicherheit. Die Community von SNMP-Managern kann auf die Agent-MIB zugreifen. Diese wird durch einen Community-String definiert.

Die SNMPv2c-Unterstützung umfasst einen Mechanismus zum Sammelabruf und detaillierte Fehlermeldungen, die den Verwaltungsstationen gemeldet werden. Der Massenabruf-Mechanismus unterstützt das Abrufen von Tabellen und großen Informationsmengen, wodurch die Anzahl der erforderlichen Rundreisen minimiert wird. Die verbesserte Fehlerbehandlungsunterstützung von SNMPv2c beinhaltet erweiterte Fehlercodes, die verschiedene Fehlertypen unterscheiden. Diese Bedingungen werden in SNMPv1 durch einen einzigen Fehlercode gemeldet. Folgende drei Arten von Ausnahmen werden ebenfalls gemeldet: Kein solches Objekt, keine solche Instanz und Ende der MIB-Ansicht.

SNMPv3 ist ein Sicherheitsmodell, in dem eine Authentifizierungsstrategie für einen Benutzer und die Gruppe, in der sich der Benutzer befindet, eingerichtet wird. Eine Sicherheitsstufe ist der zulässige Sicherheitsgrad innerhalb eines Sicherheitsmodells. Eine Kombination aus Sicherheitsmodell und Sicherheitsstufe bestimmt, welcher Sicherheitsmechanismus bei der Verarbeitung eines SNMP-Pakets verwendet wird.

Es stehen drei Sicherheitsmodelle zur Verfügung: SNMPv1, SNMPv2c und SNMPv3. In der folgenden Tabelle sind die Kombinationen von Sicherheitsmodellen und -ebenen sowie deren Bedeutung aufgeführt.

Modell	Stufe	Authentifizierung	Verschlüsselung	Was geschieht
V1	noAuthNoPriv	Community-String	Nein	Verwendet einen Community-String-Abgleich für d Authentifizierung.
v2c	noAuthNoPriv	Community-String	Nein	Verwendet einen Community-String-Abgleich für d Authentifizierung.
V3	noAuthNoPriv	Benutzername	Nein	Verwendet einen Benutzernamen-Abgleich für die Authentifizierung.
V3	authNoPriv	Message Digest 5 (MD5) oder Secure Hash Algorithm (SHA)	Nein	Bietet Authentifizierung auf der Basis von HMAC-M oder HMAC-SHA-Algorithmen.

V3	authPriv	MD5 oder SHA	DES (Data Encryption Standard)	Bietet Authentifizierung auf der Basis von HMAC-M oder HMAC-SHA-Algorithmen. DES 56-Bit-Verschlüsselung sowie Authentifizierung nach dem DES-56-Standard (CBC-DES)

Ein SNMP-Agent sollte implementiert werden, um die von der Managementstation unterstützte SNMP-Version zu verwenden. Ein Mitarbeiter kann mit mehreren Managern kommunizieren.

SNMPv3 unterstützt die RFCs 1901 bis 1908, 2104, 2206, 2213, 2214 sowie 2271 bis 2275. Weitere Informationen zu SNMPv3 finden Sie unter RFC 2570, Einführung in Version 3 des Internet-Standard-Netzwerkmanagement-Frameworks (dies ist kein Standarddokument).

Yang-Modelle

Yang-Modelle stellen eine strukturierte Abstraktion einer bestimmten Funktion oder Hardware-Eigenschaften eines Systems dar. In Netzwerkelementen könnte ein Yang-Modell ein Routing-Protokoll, interne physische Sensoren-Arrays, darstellen. Die YANG-Sprache und -Terminologie wird in RFC 6020 beschrieben und als Nächstes auf RFC 7950 aktualisiert. Auf hoher Ebene organisiert ein Yang-Modell die Daten, die die Hauptstruktur darstellen, in Untermodule und Container, die eine Liste von Unterknoten verwandt sind. Im Folgenden werden verschiedene Knotentypen erläutert.

Ein Leaf-Knoten enthält einfache Daten wie eine ganze Zahl oder eine Zeichenfolge. Es hat genau einen Wert eines bestimmten Typs und keine untergeordneten Knoten.

```
leaf host name {
        Typzeichenfolge;
        Beschreibung "Hostname für dieses System";
    }
```

Eine Leaf-Liste ist eine Folge von Leaf-Knoten mit genau einem Wert pro Leaf.

```
leaf list domain-search {
      Typzeichenfolge;
      Beschreibung "Liste der zu durchsuchenden Domänennamen";
  }
```

Ein Containerknoten wird verwendet, um verknüpfte Knoten in einer Unterstruktur zu gruppieren. Ein Container hat nur untergeordnete Knoten und keinen Wert. Ein Container kann eine beliebige Anzahl untergeordneter Knoten jeder Art enthalten (einschließlich Leafs, Listen, Container und Leaf-Listen).

```
Containersystem {
      Containeranmeldung {
         leaf message {
            Typzeichenfolge;
            Beschreibung
               "Meldung zu Beginn der Anmeldesitzung";
         }
      }
  }
```

Eine Liste definiert eine Sequenz von Listeneinträgen. Jeder Eintrag ist wie eine Struktur oder eine Datensatzinstanz und wird durch die Werte der Schlüsselblätter eindeutig identifiziert. Eine Liste kann mehrere Schlüsselblätter definieren und eine beliebige Anzahl untergeordneter Knoten aller Art enthalten (einschließlich Leafs, Listen, Container usw.).

Schließlich sieht ein Beispielmodell, das alle diese Notentypen zusammenbindet, wie im folgenden Beispiel aus:

```
## 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; } } } } }
```

Die Yang-Sprache, die auf Yang Modellen verwendet wird, zeigt jedoch nicht die Organisation der Daten in Container/Listen/Leafs an. Aus diesem Grund könnte eine bestimmte Funktion in einem Netzwerkelement mit verschiedenen Yang-Modellen dargestellt werden. Diese Herausforderung wurde mit den folgenden Yang-Modellen adressiert:

- OpenConfig-Modelle
- Native Modelle

OpenConfig-Modelle

OpenConfig-Modelle wurden unter Verwendung von unabhängigen Anbietern für das Modell entwickelt, das eine bestimmte Funktion darstellt. Der Vorteil dieses Ansatzes besteht darin, dass ein NMS diese Modelle für die Interaktion mit Netzwerkelementen in Umgebungen mit Geräten verschiedener Anbieter oder sogar mit mehreren Plattformen verwenden kann.

Wie der Name schon sagt, sind diese Modelle offen und öffentlich zugänglich für die Prüfung von Repositories wie github auf diesem Link:

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

Als Beispiel finden Sie ein openconfig-Modell für Border Gateway Protocol (BGP), ein anderes für Link Aggregation Control Protocol (LACP) und ein anderes für ISIS mit einem anderen spezifischen Modell. Im Fall von BGP finden Sie ein Modell für BGP-Fehler, ein weiteres für BGP-Richtlinien usw. Die Modelle könnten miteinander verknüpft sein, und einige Modelle können ein anderes Yang-Paket nennen. Zum Beispiel gehört openconfig-bgp-neighbor.yang zu 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;
```

Zusammenfassend lässt sich sagen, dass OpenConfig-Modelle für Protokolle ausgerichtet sind, die allen Plattformen gemeinsam sind, wie z. B. IETF- oder RFC-standardisierte Funktionen.

Native Modelle

Native Modelle dagegen sind anbieterorientierte Modelle, die detaillierte Strukturen für eine bestimmte Plattform abdecken. Beispielsweise Modelle, bei denen Sensoren physischer Werte in einem Netzwerkelement gruppiert werden, z. B. Spannungen, Temperaturen, ASIC-Zähler, Fabric-Zähler usw. Da sie von der Plattform abhängig sind, werden häufig Modelle speziell für NCS6K, ASR9K oder Cisco 8000 gefunden.

Als OpenConfig-Modelle sind auch native Modelle im Github-Repository verfügbar:

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

Da diese Modelle im Vergleich zu OpenConfig-Modellen sehr viel spezifischer und vollständiger sind, sind sie an eine bestimmte Softwareversion gebunden und können zwischen den Softwareversionen geändert werden.

Es gibt zwei Hauptkategorien für native Modelle:

- "Oper"-Modelle, die zum Abrufen von Informationen aus einem Element verwendet werden.

Beispiel: Cisco-IOS-XR-eigrp-oper.yang

- "Cfg"-Modelle zur Konfiguration eines Netzwerkelements

Beispiel: Cisco-IOS-XR-eigrp-cfg.yang

Modellgesteuerte Telemetrie nutzt allgemein "Oper"-Modelle, um Daten von der Infrastruktur zu streamen, und NMS wie der NSO verwendet "cfg"-Modelle, um Änderungen an der Konfiguration von Netzwerkelementen vorzunehmen.

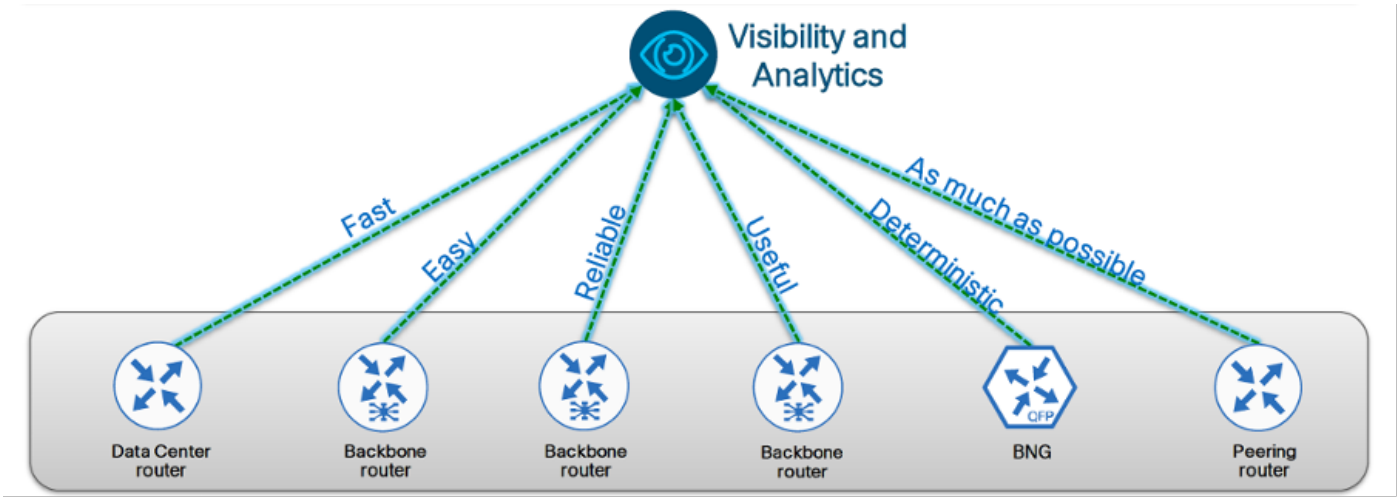
Native und OpenConfig Yang-Modelle sind auf der XR-Software im Ordner /pkg/yang vorhanden und können aufgelistet werden, um herauszufinden, ob Yang-Modelle auf einer Plattform verfügbar sind. Dieses Beispiel gilt für XRrv9k mit cXR 6.4.2:

```
RP/0/RP0/CPU0:xrv9k1#run ls /pkg/yang | grep isis
Dienstag, 22. September 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#
```

Telemetrie

Telemetrie ist ein Prozess, der das Sammeln von Informationen aus verschiedenen Remote-Elementen an einem zentralen Standort ermöglicht, der die Transparenz und die Analyseebene aggregiert.

In Netzwerkumgebungen könnten die Daten von jedem Element im Netzwerk, Routern, Switches zwischen anderen generiert werden. Die Informationen könnten mit einer sehr großen Anzahl spezifischer Protokolle, Leistungsindikatoren oder Messungen von physischen Sensoren in Zusammenhang stehen.



Im Allgemeinen befinden sich die Funktionen für Transparenz und Analyse an zentralen Punkten in Netzwerken. Das Streaming von Telemetrieinformationen erfolgt über die Übertragungsmechanismen des Netzwerks, sodass Telemetrieinformationen so schnell wie

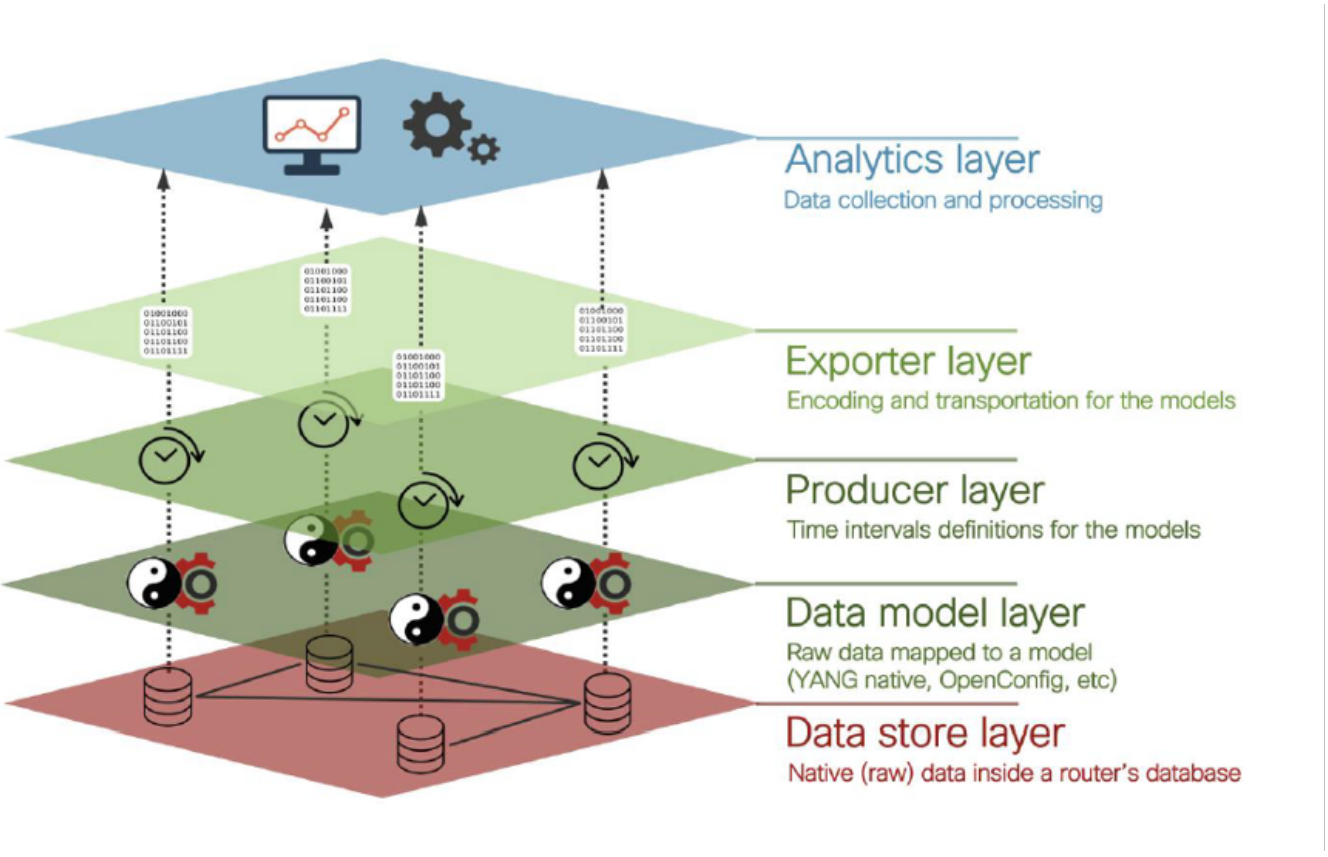
möglich skalierbar sein sollten.

Im Gegensatz zu älteren SNMP-Mechanismen verwendet Telemetrie ein Push-Paradigma, bei dem das Netzwerk bereitgestellt werden sollte, um seine eigenen Daten zu streamen, ohne in regelmäßigen Abständen abgefragt zu werden. Dies ist das Hauptmerkmal der SNMP-basierten Überwachung. Diese Bestimmung wird häufig als Abonnement bezeichnet und basiert auf einer Reihe von zu überwachenden Variablen, dem regelmäßigen Intervall für das Sampling-Intervall für die Datenerfassung und dem Remote-System, das diese Daten über das Netzwerk sendet.

Modellgesteuerte Telemetrie

MDT steht für Model Driven Telemetry und basiert, wie der Name schon sagt, auf Yang Modellen. Jeder Aspekt der Netzwerkgeräte kann mit Yang-Modellen dargestellt werden, z. B. mit der Tabelle der OSPF-Nachbarn, RIB- oder Temperatursensoren für jede Komponente modularer Systeme.

Was die MDT-Architektur angeht, so kann sie in folgende Ebenen unterteilt werden:



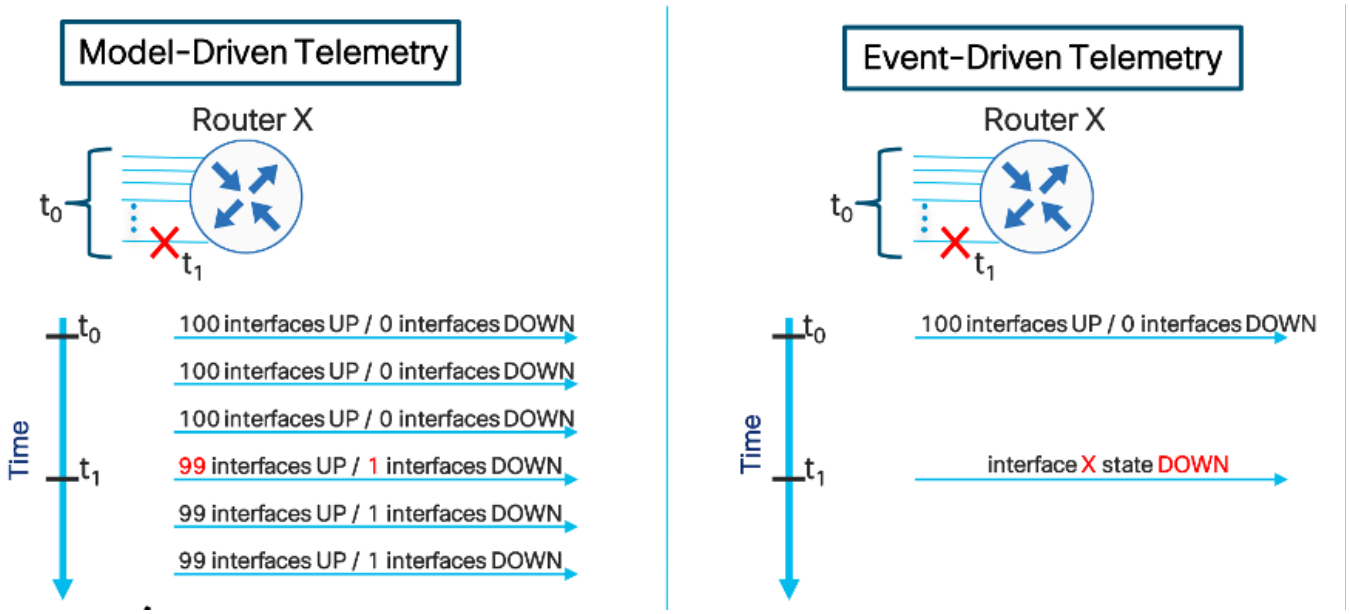
Hinweis: In Bezug auf die Produzenten-Ebene gibt es bei der modellgesteuerten Telemetrie eine Definition für das Abtastintervall, die festlegt, wie oft das Gerät die interne Datenbank nach Rohdaten abruft und diese Daten in der Datenmodellebene organisiert.

Das Telemetrie-Abonnement definiert außerdem, welche Modelle und mit Containern/Pfad Daten erzeugen, die in die Analytics-Schicht gestreamt werden sollen. Diese Definition würde sich auf relevante Informationen für Geschäftszwecke auswirken. Die MDT-Definition dieses Sensorpfads ist analog zur Definition der OID, die über SNMP abgerufen werden soll, da beide Techniken strukturierte Daten mit definierter Abtastrate erstellen.

Ereignisgesteuerte Telemetrie

EDT steht für Event Driven Telemetry und basiert für die Struktur auch auf Yang Modellen. Der Hauptunterschied besteht darin, dass der Trigger für die Erfassung und den Datenstrom kein reguläres Intervall ist, sondern ein bestimmtes Ereignis ist, z. B. Grenzwertüberschreitung, Verknüpfungsereignisse, Hardwarefehler usw.

Im Folgenden wird ein Vergleich einer Veranstaltung mit modellbasierter Telemetrie und ereignisgesteuerter Telemetrie dargestellt:



Tipp: Diese Abbildung zeigt redundante Nachrichten mithilfe von MDT, aber nur Meldungen, die Änderungen mithilfe von EDT darstellen.

Transport

Die Telemetrie sollte so zuverlässig wie möglich sein. Daher ist es sinnvoll, den Transmission Control Protocol (TCP)-basierten Transport für die Verwendung sitzungsorientierter Sockets zwischen der Infrastruktur und der Analytik-Ebene zu verwenden, die Collectors für die Sitzungserstellung implementieren sollten.

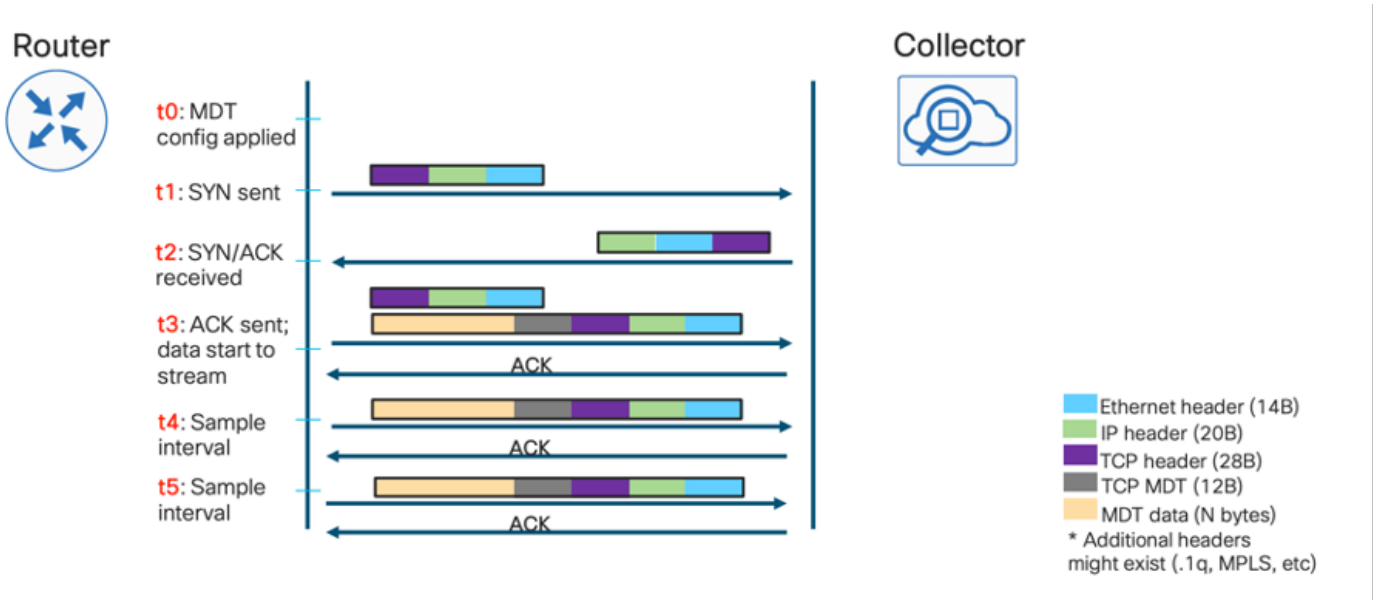
Beim Einsatz von Telemetrie gibt es zwei Hauptansätze, die sich im 3-Wege-Handshake-Erstfluss voneinander unterscheiden.

Hinweis: Im Dial-Out-Modus wird die Einrichtung der Sitzung auf der Infrastrukturseite gestartet.

Dies bedeutet, dass die Sensoren von Interesse für die Netzwerkelemente konfiguriert werden sollten. Im Gegensatz dazu ermöglicht der Einwahlansatz eine einfachere Konfiguration von Netzwerkelementen, da der Collector während der Einrichtungsphase spezifische Sensorpfade anfordern sollte.

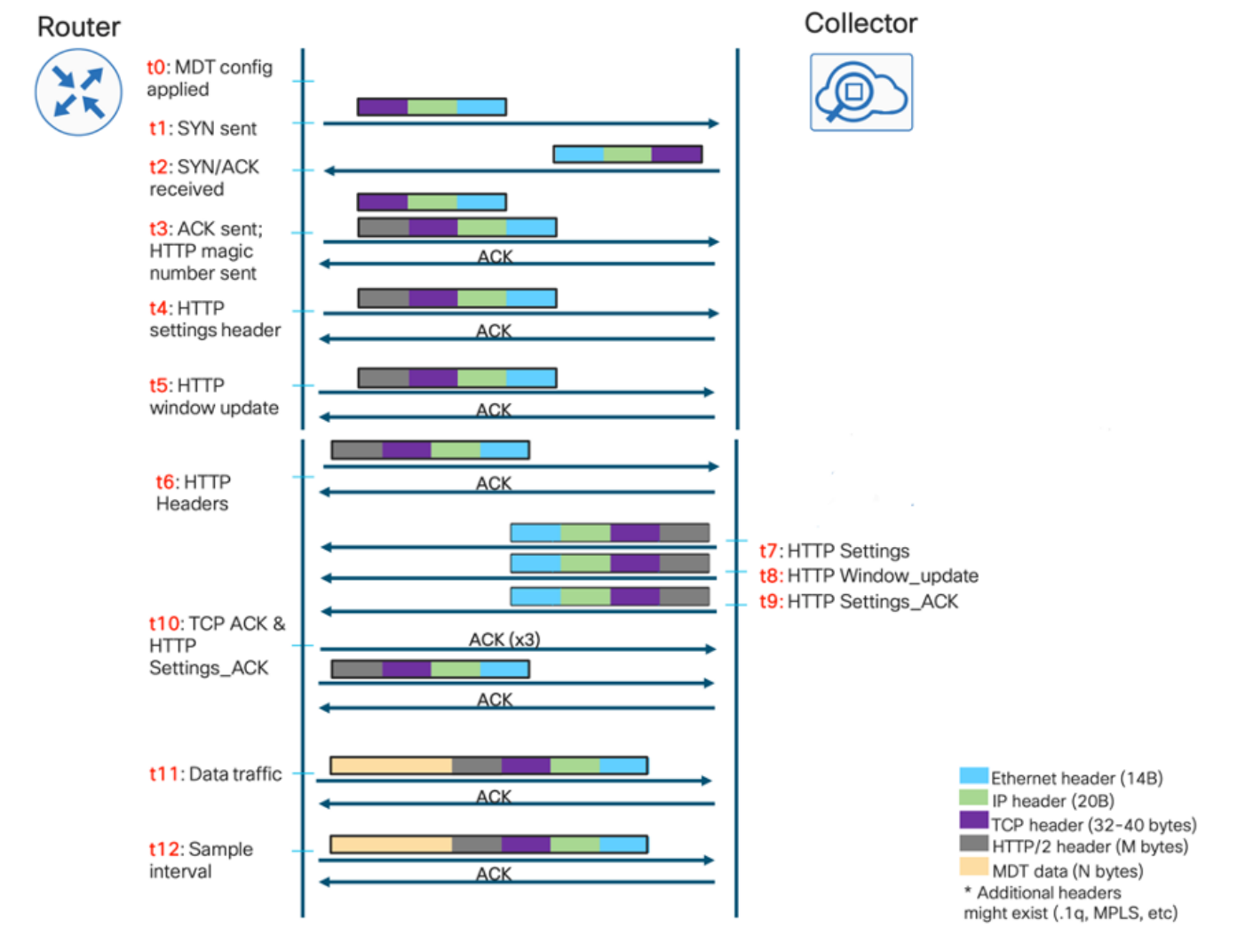
TCP

TCP ist die einfachste Methode, um eine verbindungsorientierte Sitzung zwischen einem Netzwerkelement und einem Telemetrie-Collector herzustellen. Der Datenstrom beginnt vom Router zum Collector, der das ACK zu Zuverlässigkeitszwecken an den Router zurückgesendet hat:



gRPC

Da Google Protocol RPC (gRPC) über Hypertext Transfer Protocol/2 (HTTP/2) arbeitet, sollte die Sitzung selbst bei der Einrichtung aufgebaut werden und die Geschwindigkeitssteuerung von der Collector-Seite aus nativ ermöglicht werden:



gNMI/gNOI

gRPC Network Management Interface (gNMI) ist ein von Google entwickeltes gRPC-Netzwerkverwaltungsprotokoll. gNMI bietet die Möglichkeit, die Konfiguration von Netzwerkgeräten zu installieren, zu bearbeiten und zu löschen und Betriebsdaten anzuzeigen. Die von gNMI bereitgestellten Inhalte können mithilfe von YANG modelliert werden.

gNMI verwendet gRPC-HTTP/2 zum Einrichten einer Verbindung und bietet einen bidirektionalen Kanal zwischen Netzwerkelementen und einem NMS, der ebenfalls ein Telemetrie-Collector sein könnte, aber auch eine Schnittstelle zum Verwalten der Geräte bereitstellt.

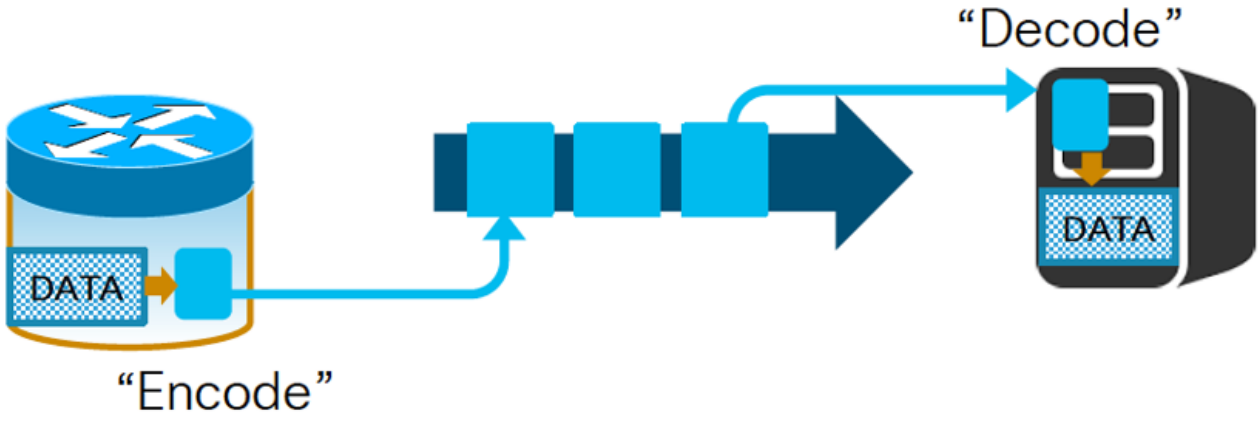
Zwischen den Operationen, die von diesem Protokoll unterstützt werden, finden Sie gNMI Get, gNMI Set, die die angeforderten Informationen zurückgeben, Erfolgsmeldungen oder Fehlermeldungen.

gRPC Network Operations Interface (gNOI) ist eine Sammlung von Mikrodiensten, die denselben Kommunikationskanal wie gNMI verwenden, jedoch generische Operationen zulassen, die nicht mit der Konfiguration selbst in Zusammenhang stehen, wie Ping, Neustart, Ändern von SSL-Zertifikaten, Löschen usw.

Kodierung

Yang-Modelle definieren die Struktur der Daten, ihre Hierarchie und den Typ jedes Leaf-Knotens

darauf. Die Modellierung zeigt jedoch nicht an, wie diese Daten serialisiert werden sollen. Dieser Prozess steuert die Konvertierung strukturierter Daten in einen Bytestream, der über die TCP-Verbindung gesendet werden soll (roh TCP, gRPC, gNMI usw.).



Hinweis: Dieser Prozess sollte mit einem entsprechenden Mechanismus im Netzwerkelement implementiert werden, der die Daten codieren soll, und der Collector sollte diese Daten decodieren.

JSON

Der erste Kodierungsmechanismus ist ein natives JavaScript Object Notation (JSON)-Format, das zwar allgemein bekannt ist, aber menschlich orientiert ist, da jeder Schlüssel als Zeichenfolge dargestellt wird, was hinsichtlich der Nachrichtengröße ineffizient ist. Der Hauptvorteil von JSON besteht darin, dass die Analyse und das Lesen in Textform als nächstes Beispiel einfach ist:

```
{ “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

Das Codierungsformat von Google Protocol Buffers-Key Value (GPB-KV) wird auch als selbstbeschreibendes GPB bezeichnet, da es Protokollpuffer verwendet, um Nachrichten zu nutzen, die auf bestimmte Elemente auf Yang-Modellen verweisen. Dies impliziert, dass nur eine PROTO-Datei zum Codieren/Decodieren von Zwecken benötigt wird und dass die Schlüssel selbst aus den Daten in selbst beschriebenen Zeichenfolgen enthalten sind.

```
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

Schließlich geht Google Protocol Buffers (GPB), auch kompakte GPB genannt, diesen Ansatz einen Schritt weiter und erfordert .proto-Dateien, um jeden Schlüssel der Struktur zu kartografieren, wodurch es viel effizienter in Bezug auf die Nachrichtengröße, da alles als binäre Werte gesendet wird. Der Nachteil besteht jedoch darin, dass jede PROTO-Datei kompiliert werden muss, die jedem Yang-Modell zugeordnet ist, das von der Infrastruktur/dem Collector unterstützt wird.

```
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\005Null0" 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\026GigabitEthernet0/0/0/0" 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
```

MDT-Konfiguration in IOS XR

Die Hauptkomponenten für das Streaming modellgesteuerter Telemetriedaten sind:

- Sitzung
- Sensorpfad
- Abonnement

- Transport und Codierung

Die Sitzungsoptionen können, wie bereits erwähnt, ein- oder ausgewählt werden. So erstellen Sie die Konfiguration in IOS XR.

Wählmodus

Im Dial-Out-Modus startet der Router eine Sitzung mit den Zielen, basierend auf dem Abonnement. Dieser Vorgang sollte folgende Schritte umfassen:

- Zielgruppe erstellen
- Erstellen einer Sensorgruppe
- Abonnement erstellen
- Dial-Out-Konfiguration validieren

Um eine Zielgruppe zu erstellen, müssen Sie die IPv4-Adresse (Internet Protocol Version 4)/IPv6-Adresse (Internet Protocol Version 6) des Collectors und den Port kennen, der diese Anwendung bedient. Außerdem müssen Sie das Protokoll und die Codierung angeben, die auf dem Netzwerkgerät und dem Collector vereinbart werden sollen.

Schließlich müssen Sie möglicherweise das Virtual Routing and Forwarding (VRF) angeben, das für die Kommunikation mit der Collector-Netzwerkadresse verwendet wird.

Als Nächstes wird ein Beispiel für eine WählOut-Konfiguration dargestellt:

```
Telemetriemodell
 Zielgruppe DG1
  VRF-MGMT
  address-family ipv4 192.168.122.20 port 5432
   codieren selbstbeschreibend-gpb
   Protokoll tcp
  !
 !
```

Als Nächstes werden die Verschlüsselungsoptionen angezeigt:

```
RP/0/RP0/CPU0:C8000-1(config-model-powered-dest-addr)#encoding?
  GPB GPB-Codierung
  JSON-Kodierung
  selbstbeschreibend-gpb Selbstbeschreibende GPB-Codierung ← Auch bekannt als GPB-KV
RP/0/RP0/CPU0:C8000-1(config-model-powered-dest-addr)#encoding
```

Die Protokolloptionen:

```
RP/0/RP0/CPU0:C8000-1(config-model-powered-dest-addr)#protocol?
  grpc gRPC
  TCP
  UDP
RP/0/RP0/CPU0:C8000-1(config-model-powered-dest-addr)#protocol grpc ?
  gzip gRPC gzip-Komprimierung
  Keine TLS TLS
  tls-hostname TLS-Hostname
  <cr>
RP/0/RP0/CPU0:C8000-1(config-model-powered-dest-addr)#protocol tcp ?
  <cr>
RP/0/RP0/CPU0:C8000-1(config-model-powered-dest-addr)#protocol udp ?
  Paketgröße UDP-Paketgröße
  <cr>
RP/0/RP0/CPU0:C8000-1(config-model-powered-dest-addr)#protocol udp
```

Das TCP-Protokoll ist einfach und benötigt nur die Port-Einstellungen, die an die IPv4/IPv6-Adresse angeschlossen sind. User Datagram Protocol (UDP) dagegen ist verbindungslos, sodass der Status der Zielgruppe immer aktiv ist.

Die Komprimierung in gRPC kann mithilfe des optionalen ***gzip***-Schlüsselworts erfolgen. gRPC verwendet standardmäßig TLS. Daher sollte für diese Verwendung ein Zertifikat lokal auf dem Router installiert werden. Dieses Verhalten kann durch die Konfiguration des ***no-tls***-Schlüsselworts überschrieben werden. Schließlich können Sie einen anderen Hostnamen für Zertifikatszwecke mithilfe des ***tls-hostname***-Schlüsselworts angeben.

Als Nächstes sollte ein Abschnitt zur Sensorgruppe hinzugefügt werden, in dem die Sensorpfade aufgelistet sind, die für uns von Interesse sind. Dieser Abschnitt ist einfach, aber es ist wichtig zu wissen, dass der Sensorpfad selbst das Filtern ermöglicht, um mehrere Ressourcen wie CPU und Bandbreite zu optimieren.

```
Telemetriemodell
 Sensorgruppe SG1
  sensor-path Cisco-IOS-XR-wdsysmon-fd-oper:Systemüberwachung/CPU-Auslastung
  sensor-path Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interface[interface-name='Mgmt*']/data-rate
 !
!
```

Hinweis: Das für einen Sensorpfad erforderliche Format ist <Modellname>:<Containerpfad>.

Dieses Dokument stellt die Zuordnung der SNMP-basierten Überwachung mithilfe der OID dar, die in diesem Legacy-Ansatz "Blätter" in YANG-Modellen darstellt, dargestellt mit XPATHs, die mit den gleichen "Blättern" übereinstimmen.

In der letzten Konfigurationsphase muss ein Abonnement konfiguriert werden, das die

Sensorgruppe mit einem Rhythmus für das Telemetriestreaming an eine Zielgruppe bindet.

```
Telemetriemodell
 Abonnement-SU1
  Sensor-Group-ID SG1 Samplingintervall 5000
  Ziel-ID DG1
 !
!
```

In diesem Beispiel wird ein Abtastintervall von 5000 Millisekunden (5 Sekunden) verwendet, das relativ zum Ende der vorherigen Auflistung ist. Um dieses Verhalten zu ändern, können Sie das ***Sample-Intervall-***Schlüsselwort mit der ***strict-timer-***Option ändern.

Zur Verifizierung können Sie den folgenden Befehl verwenden, der den Abonnementstatus abdeckt. Diese Methode ermöglicht auch die Abdeckung von Sensorikgruppen- und Zielgruppendaten.

```
RP/0/RP0/CPU0:C8000-1#sh Telemetrie-Modellabonnement SU1
Mi 18. November 15:38:01.397 UTC
Abonnement: SU1
—
  Bundesland: AKTIV
  Sensorgruppen:
  ID: SG1
    Stichprobenintervall: 5000 ms
    Taksintervall: NA
    Sensorpfad: Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interface[interface-name='Mgmt*']/data-rate
    Status des Sensorpfads: Behoben
    Sensorpfad: Cisco-IOS-XR-wdsysmon-fd-oper:Systemüberwachung/CPU-Auslastung
    Status des Sensorpfads: Behoben
  Zielgruppen:
  Gruppen-ID: DG1
    Ziel-IP: 192.168.122.10
    Zielport: 5432
    Ziel-VRF: MGMT(0x60000001)
    Codierung: selbstbeschreibend-gpb
    Transport: tcp
    Status: Aktiv
    TLS: False
    Gesendete Bytes insgesamt: 636284346
    Versendete Pakete insgesamt: 4189
    Zuletzt gesendet: 2020-11-18 15:37:58.170077650 +0000
  Erfassungsgruppen:
  —
    ID: 9
    Stichprobenintervall: 5000 ms
    Taksintervall: NA
    Heartbeat immer: Falsch
    Codierung: selbstbeschreibend-gpb
    Anzahl der Sammlungen: 1407
    Erfassungsdauer: mindestens 4 ms, max. 13 ms
    Gesamtdauer: mind. 8 ms durchschn. 10 ms max.: 20 ms
    Insgesamt aufgeschoben: 0
    Gesamtzahl der Sendefehler: 0
    Gesamtzahl der Versandverluste: 0
    Sonstige Fehler gesamt: 0
    Keine Dateninstanzen: 1407
    Letzter Erfassungsstart: 2020-11-18 15:37:57.1699545994 +0000
    Letzte Sammlung: 2020-11-18 15:37:57.1699555589 +000
    Sensorpfad: Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/data-rate
    ID: 10
    Stichprobenintervall: 5000 ms
    Taksintervall: NA
    Heartbeat immer: Falsch
    Codierung: selbstbeschreibend-gpb
    Anzahl der Sammlungen: 1391
    Erfassungsdauer: mindestens 178 ms, max.: 473 ms
    Gesamtdauer: mindestens 247 ms (durchschn. 283 ms, max.: 559 ms)
    Insgesamt aufgeschoben: 0
    Gesamtzahl der Sendefehler: 0
    Gesamtzahl der Versandverluste: 0
    Sonstige Fehler gesamt: 0
    Keine Dateninstanzen: 0
    Letzter Erfassungsstart: 2020-11-18 15:37:58.1699805906 +0000
    Letzte Sammlung Ende: 2020-11-18 15:37:58.170078415 +0000
    Sensorpfad: Cisco-IOS-XR-wdsysmon-fd-oper:Systemüberwachung/CPU-Auslastung
RP/0/RP0/CPU0:C8000-1#
```

Einwahlmodus

Im Einwahlmodus initiiert der Collector die Verbindung zu den Netzwerkelementen. Anschließend sollte der Collector das Interesse zum Erstellen eines Abonnements angeben.

Die Konfiguration umfasst die folgenden Schritte:

- gRPC-Dienst aktivieren
- Setup-Sensorgruppen
- Überprüfung

Um den gRPC-Dienst zu aktivieren, wird die Konfiguration als Nächstes angezeigt:

```
!
GRPC
 VRF-MGMT
 Port 57400
 Keine TLS
 Adressfamilie Dual
!
```

Die Optionen sind einfach, einschließlich VRF und TCP-Port. gRPC verwendet standardmäßig TLS, kann jedoch mit dem Schlüsselwort ***no-tls*** deaktiviert werden. Schließlich ermöglicht die ***doppelte*** Option der ***Adressfamilie*** die Verbindung über IPv4 und IPv6.

Als Nächstes müssen für die Einwahl Sensorgruppen lokal definiert werden, die später vom Collector zur Definition eines Abonnements verwendet werden.

Telemetriemodell

```
Sensorgruppe SG3
 sensor-path Cisco-IOS-XR-wdsysmon-fd-oper:Systemüberwachung/CPU-Auslastung
 sensor-path Cisco-IOS-XR-fib-common-oper:fib-statistics/knoten/drop
!
!
```

An diesem Punkt ist die Konfiguration für den Einwahlmodus abgeschlossen, und der Collector selbst kann mithilfe von gRPC ein Abonnement für den Router erstellen. Zur Überprüfung können Sie den gleichen Ansatz wie im Wählmodus wählen:

```
RP/0/RP0/CPU0:C8000-1#sh Telemetrie-Modellabonnement anx-1605878175837
Fr. 20. November 13:58:37.894 UTC
Abonnement: anx-1605878175837
—
  Bundesland: AKTIV
  Sensorgruppen:
  ID: SG3
    Beispielintervall: 15.000 ms
    Taksintervall: NA
    Sensorpfad: Cisco-IOS-XR-wdsysmon-fd-oper:Systemüberwachung/CPU-Auslastung
    Status des Sensorpfads: Behoben
    Sensorpfad: Cisco-IOS-XR-fib-common-oper:fib-statistics/knoten/drop
    Status des Sensorpfads: Behoben
  Zielgruppen:
  Gruppen-ID: Einwahl_1003
    Ziel-IP: 192.168.122.10
    Zielport: 46974
    Komprimierung: gzip
    Codierung: json
    Transport: Wählen
    Status: Aktiv
    TLS: False
    Gesendete Bytes insgesamt: 71000035
    Gesamtzahl der gesendeten Pakete: 509
    Zuletzt gesendet: 2020-11-20 13:58:32.1030932699 +000
  Erfassungsgruppen:
  —
    ID: 5
    Stichprobenintervall: 15.000 ms
    Taksintervall: NA
```

```
  Takt immer:     Falsch
  Codierung: json
  Anzahl der Sammlungen: 170
  Erfassungsdauer: mindestens 273 ms, max.: 640 ms
  Gesamtdauer: mind. 276 ms (durchschn. 390 ms) max.: 643 ms
  Insgesamt aufgeschoben: 0
  Gesamtzahl der Sendefehler: 0
  Gesamtzahl der Versandverluste: 0
  Sonstige Fehler gesamt: 0
  Keine Dateninstanzen: 0
  Letzter Erfassungsbeginn: 2020-11-20 13:58:32.1030283276 +0000
  Letzte Sammlung Ende: 2020-11-20 13:58:32.1030910008 +0000
  Sensorpfad: Cisco-IOS-XR-wdsysmon-fd-oper:Systemüberwachung/CPU-Auslastung
  ID: 6
  Stichprobenintervall: 15.000 ms
  Taksintervall: NA
  Heartbeat immer: Falsch
  Codierung: json
  Anzahl der Sammlungen: 169
  Erfassungsdauer: mindestens 15 ms, max.: 33 ms
  Gesamtdauer: mindestens 17 ms (durchschn. 22 ms, max.: 33 ms)
  Insgesamt aufgeschoben: 0
  Gesamtzahl der Sendefehler: 0
  Gesamtzahl der Versandverluste: 0
  Sonstige Fehler gesamt: 0
  Keine Dateninstanzen: 0
  Letzter Erfassungsbeginn: 2020-11-20 13:58:32.1030910330 +000
  Letzte Sammlung: 2020-11-20 13:58:32.1030932787 +000
  Sensorpfad: Cisco-IOS-XR-fib-common-oper:fib-statistics/knoten/drop
RP/0/RP0/CPU0:C8000-1#
```

Tipp: Beachten Sie, dass auf dem Router für den Einwahlmodus keine Prioritätsstufe, Kodierung, Collector-IP oder kein Transport hardcodiert ist.

SNMP-Migration auf MDT

Um die Migration vom traditionellen SNMP zum Telemetriemodell durchzuführen, sollten folgende Aspekte behandelt werden:

- MIB-Migration in XPATH
- Trap-Migration in Telemetrie
- Sicherheitsüberlegungen

MIB-Migration in XPATH

Zu diesem Zweck können wir MIB nach eigenen Hierarchien kategorisieren, die (mindestens auf hoher Ebene) einer bestimmten Funktionalität zugeordnet werden können.

BGP4-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensoren für BGP-Peering-Sitzungen eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
bgpPeerLastError	1.3.6.1.2.1.15.3.1.14	Der letzte Fehlercode und Untercode, der von diesem Peer für diese Verbindung angezeigt wird. Wenn kein Fehler aufgetreten ist, ist dieses Feld 0. Andernfalls enthält das erste Byte	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instance/instan active/default-vrf/neighbo table/neighbor/neighbor/l notify-error-code

		dieser beiden Byte OCTET STRING den Fehlercode, das zweite Byte den Untercode.	
bgpPeerOutUpdates	1.3.6.1.2.1.15.3.1.11	Die Anzahl der über diese Verbindung übertragenen BGP-UPDATE-Nachrichten	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instance/instan active/default-vrf/afs/af/neighbor-af-table/neighbor/update-messages-out
bgpPeerInUpdates	1.3.6.1.2.1.15.3.1.10	Die Anzahl der BGP UPDATE-Nachrichten, die über diese Verbindung empfangen wurden.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instance/instan active/default-vrf/afs/af/neighbor-af-table/neighbor/update-messages-in
bgpPeerNegotiatedVersion	1.3.6.1.2.1.15.3.1.4	Die ausgehandelte Version von BGP, das zwischen den beiden Peers ausgeführt wird. Dieser Eintrag MUSS null (0) sein, es sei denn, der bgpPeerState befindet sich im openconfirm-Zustand oder im etablierten Zustand. Beachten Sie, dass die rechtlichen Werte für dieses Objekt zwischen 0 und 255 liegen.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instance/instan active/default-vrf/afs/af/neighbor-af-table/neighbor/negotiatio protocol-version
bgpPeerState	1.3.6.1.2.1.15.3.1.2	Der BGP-Peer-Verbindungsstatus.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instanz/instanc active/default-vrf/afs/af/neighbor-af-table/neighbor/connectio state
bgpPeerRemoteAddr	1.3.6.1.2.1.15.3.1.7	Die Remote-IP-Adresse des BGP-Peers dieses Eintrags.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instanz/instanc active/default-vrf/afs/af/neighbor-af-table/neighbor/connectio remote-address
bgpPeerLocalAddr	1.3.6.1.2.1.15.3.1.5	Die lokale IP-Adresse der BGP-Verbindung dieses Eintrags.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instanz/instanc active/default-vrf/afs/af/neighbor-af-table/neighbor/connectio local-address
bgpPeerFsmEtablishedTime	1.3.6.1.2.1.15.3.1.16	Dieser Timer gibt an, wie lange (in Sekunden) dieser Peer sich im etablierten Zustand befindet oder wie lange seit dem letzten Peer im etablierten Zustand war.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instance/instan active/default-vrf/afs/af/neighbor-af-table/neighbor/connectio down-time

bgpPeerAdminStatus	1.3.6.1.2.1.15.3.1.3	Wenn ein neuer Peer konfiguriert wird oder der Router gestartet wird, wird er auf Null gesetzt. Der gewünschte Status der BGP-Verbindung. Bei einem Übergang von 'stop' zu 'start' wird das BGP Manual Start Event generiert. Bei einem Übergang von 'start' zu 'stop' wird das BGP Manual Stopp Event generiert. Dieser Parameter kann verwendet werden, um BGP-Peer-Verbindungen neu zu starten. Bei der Bereitstellung des Schreibzugriffs auf dieses Objekt ohne ausreichende Authentifizierung sollte Vorsicht geboten sein.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instance/instan active/default-vrf/afs/af/neighbor-af-table/neighbor/connectio admin-status

CISCO-BGP4-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensensoren für den BGP-Sitzungsstatus und das ausgetauschte Präfix eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
cbgpPeer2RemoteAs	1.3.6.1.4.1.9.9.187.1.2.5.1.11	Die in der BGP OPEN-Nachricht erhaltene Nummer des autonomen Remote-Systems.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instance/instan active/default-vrf/sessions/session/rem as
cbgpPeer2PrevState	1.3.6.1.4.1.9.9.187.1.2.5.1.29	Der vorherige Status der BGP-Peer-Verbindung.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instanz/instanc active/default-vrf/afs/af/neighbor-af-table/neighbor/previous-connection-state
cbgpPeer2State	1.3.6.1.4.1.9.9.187.1.2.5.1.3	Der BGP-Peer-Verbindungsstatus.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instanz/instanc active/default-vrf/afs/af/neighbor-af-table/neighbor/connecti state
cbgpPeer2LocalAddr	1.3.6.1.4.1.9.9.187.1.2.5.1.6	Die lokale IP-Adresse der BGP-Verbindung dieses Eintrags.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instanz/instanc active/default-vrf/afs/af/neighbor-af-table/neighbor/connecti local-address

cbgpPeer2AdvertisedPrefixes	1.3.6.1.4.1.9.9.187.1.2.8.1.6	Dieser Zähler wird erhöht, wenn für diese Verbindung ein Routenpräfix angekündigt wird, das zu einer Adressfamilie gehört. Sie wird auf Null initialisiert, wenn die Verbindung einem harten Zurücksetzen unterzogen wird.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instanz/instanc active/default-vrf/afs/af/neighbor-af-table/neighbor/af-data/prefixes-insered
cbgpPeer2AcceptedPrefixes	1.3.6.1.4.1.9.9.187.1.2.8.1.1	Anzahl der akzeptierten Routenpräfixe für diese Verbindung, die zu einer Adressfamilie gehören.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instanz/instanc active/default-vrf/afs/af/neighbor-af-table/neighbor/af-data/prefixes-accept
cbgpPeerPrefixLimit	1.3.6.1.4.1.9.9.187.1.2.1.1.3	Max. Anzahl von Routing-Präfixen, die für diese Verbindung akzeptiert werden	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instanz/instanc active/default-vrf/afs/af/neighbor-af-table/neighbor/af-data/m prefix-limit
cbgpPeer2PrefixThreshold	1.3.6.1.4.1.9.9.187.1.2.8.1.4	Der Präfixgrenzwert (%) für eine Adressfamilie dieser Verbindung, bei dem die Warnmeldung, die die Präfixanzahl angibt, den Schwellenwert überschreitet oder eine entsprechende SNMP-Benachrichtigung generiert wird.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/config-instance/config-instance default-vrf/entity-configuration/af-depend config/max-prefix-warn-threshold

CISCO-KLASSE-BASED-QOS-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensensoren für Statistiken in QoS-Klassen/-Richtlinien eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
cbQosCMDropBitRate	1.3.6.1.4.1.9.9.166.1.15.1.1.18	Die Bitrate der Verwerfungen pro Klasse ergibt sich aus allen Features, die zu Verwerfungen führen können (z. B. Polizei, willkürliche	Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/input/serv policy-names/service-po instance/statistics/class-stats/general-stats/total-rate

		Erkennung usw.).	Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/output/se policy-names/service-po instance/statistics/class-stats/general-stats/total-rate
cbQosCMDropPkt64	1.3.6.1.4.1.9.9.166.1.15.1.1.14	Der 64-Bit-Zähler der verlorenen Pkte pro Klasse aufgrund aller Funktionen, die zu Verlusten führen können (z. B. Polizei, willkürliche Erkennung usw.).	Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/input/serv policy-names/service-po instance/statistics/class-stats/general stats/total-packages Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/output/se policy-names/service-po instance/statistics/class-stats/general stats/total-packages
cbQosCMPrePolicyPkt64	1.3.6.1.4.1.9.9.166.1.15.1.1.3	Die 64-Bit-Anzahl eingehender Pakete vor der Ausführung von QoS-Richtlinien.	Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/input/serv policy-names/service-po instance/statistics/class-stats/general-stats/pre-p matching-pakete Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/output/se policy-names/service-po instance/statistics/class-stats/general-stats/pre-p matching-pakete
cbQosCMName	1.3.6.1.4.1.9.9.166.1.7.1.1.1	Name der Klassenzuordnung.	Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/input/serv policy-names/service-po instance/statistics/class-stats/class-name
cbQosCMPostPolicyByte64	1.3.6.1.4.1.9.9.166.1.15.1.1.10	Die 64-Bit-Anzahl ausgehender Oktette nach der Ausführung von QoS-Richtlinien.	Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/input/serv policy-names/service-po instance/statistics/class-stats/class-stats/general stats/Transmit-bytes Cisco-IOS-XR-qos-ma-oper:qos/interface/interfa utput/service-policy-names/service-policy-

			instance/statistics/class- stats/class-stats/general stats/Transmit-bytes
cbQosIfIndex	1.3.6.1.4.1.9.9.166.1.1.1.1.4	ifIndex für die Schnittstelle, an die dieser Dienst angeschlossen ist. Dieses Feld ist nur sinnvoll, wenn die logische Schnittstelle über einen snmp ifIndex verfügt. Beispielsweise ist der Wert dieses Felds bedeutungslos, wenn cbQosIfType die Kontrollebene ist.	Cisco-IOS-XR-infra- policymgr-oper:policy- manager/global/policy- map/policy-map-types/p map-type/policy-maps
cbQoConfigIndex	1.3.6.1.4.1.9.9.166.1.5.1.1.2	Ein beliebiger (systemzugewiesener) Konfigurationsindex (instanzunabhängig) für jedes Object. Alle Objekte mit derselben Konfiguration verwenden denselben Konfigurationsindex.	Cisco-IOS-XR-infra- policymgr-oper:policy- manager/global/policy- map/policy-map-types/p map-type/policy-maps
cbQosCMPrePolicyByte64	1.3.6.1.4.1.9.9.166.1.15.1.1.6	Die 64-Bit-Anzahl eingehender Oktette vor der Ausführung von QoS-Richtlinien.	Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/input/serv policy-names/service-po instance/statistics/class- stats/class-stats/general stats/pre-policy-matching bytes Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/output/se policy-names/service-po instance/statistics/class- stats/class-stats/general stats/pre-policy-matching bytes

CISCO ENHANCED-MEMPOOL-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensensoren für die Speichernutzung eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
cempMemPoolVerwendet	1.3.6.1.4.1.9.9.221.1.1.1.1.7	Gibt die Anzahl der Byte aus dem Speicherpool an, die derzeit von Anwendungen auf der physischen Einheit verwendet werden.	Cisco-IOS-XR-nto-misc-oper:Arbeitsspeicherübers Knoten/Knoten/Zusamme ung
cempMemPoolHCUsed	1.3.6.1.4.1.9.9.221.1.1.1.1.18	Gibt die Anzahl der Byte aus dem Speicherpool an, die derzeit von Anwendungen auf der physischen Einheit verwendet werden. Dieses Objekt ist eine 64-Bit-Version von cempMemPoolUsed.	Cisco-IOS-XR-nto-misc-oper:Speicherübersicht/K /Knoten/Detail/insgesamt verwendet
cempMemPoolHCFree	1.3.6.1.4.1.9.9.221.1.1.1.1.20	Gibt die Anzahl der Byte aus dem Speicherpool an, die derzeit nicht für die physische Einheit verwendet werden. Dieses Objekt ist eine 64-Bit-Version von cempMemPoolFree.	Cisco-IOS-XR-nto-misc-oper:Arbeitsspeicherübers Knoten/Knoten/Detail/freie physischer Speicher

CISCO-ENTITY-FRU-CONTROL-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensensoren für die vor Ort austauschbaren Einheiten im überwachten System eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
cefcFRUPowerOperStatus	1.3.6.1.4.1.9.9.117.1.1.2.1.2	Betriebszustand der FRU.	Cisco-IOS-XR-invmgr-oper:Bestand/Entitäten bute/fru-info/power-Operational-state
cefcFRUPowerAdminStatus	1.3.6.1.4.1.9.9.117.1.1.2.1.1	Vom Administrator gewünschter FRU-Stromversorgungszustand.	Cisco-IOS-XR-invmgr-oper:Bestand/Entitäten bute/fru-info/power-administrative-state
cefcModuleStatusLastChangeTime	1.3.6.1.4.1.9.9.117.1.2.1.1.4	Der Wert von sysUpTime zum Zeitpunkt der Änderung von cefcModuleOperStatus.	Cisco-IOS-XR-invmgr-oper:Bestand/Entitäten bute/fru-info/last-Operational-state-chan
cefcModuleUpTime	1.3.6.1.4.1.9.9.117.1.2.1.1.8	Dieses Objekt stellt die Betriebszeit für das Modul seit der letzten	Cisco-IOS-XR-invmgr-oper:Bestand/Einheiten eiten/Attribute/FRU-INFO/Kartenverfügbark

		Neuinitialisierung bereit. Dieses Objekt ist nicht persistent. Wenn ein Modul zurückgesetzt, neu gestartet, ausgeschaltet wird, beginnt die Betriebszeit von Null.	
cefcModuleResetGrund	1.3.6.1.4.1.9.9.117.1.2.1.1.3	Dieses Objekt identifiziert den Grund für das letzte Zurücksetzen, das für das Modul durchgeführt wurde.	Cisco-IOS-XR-invmgr-oper:Bestand/Einheiten/Attribute/fru-info/creset-reason
cefcModuleOperStatus	1.3.6.1.4.1.9.9.117.1.2.1.1.2	Dieses Objekt zeigt den Betriebszustand des Moduls an.	Cisco-IOS-XR-invmgr-oper:Bestand/Entitäten bute/fru-info/card-Operational-State
cefcModuleAdminStatus	1.3.6.1.4.1.9.9.117.1.2.1.1.1	Dieses Objekt ermöglicht die administrative Steuerung des Moduls.	Cisco-IOS-XR-invmgr-oper:Bestand/Entitäten bute/fru-info/card-administrative-state

CISCO-ENTITY-SENSOR-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensensoren-Gruppen für Sensorentitäten auf dem Knoten eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
entSensorValue	1.3.6.1.4.1.9.9.91.1.1.1.1.4	Diese Variable berichtet über die aktuelle Messung, die der Sensor gemacht hat. Um den Wert dieser Variablen korrekt anzuzeigen oder zu interpretieren, müssen Sie auch entSensorType, entSensorScale und entSensorPrecision kennen. Sie können entSensorValue jedoch ohne semantische Kenntnisse mit den in entSensorThresholdTable angegebenen Schwellenwerten vergleichen.	Cisco-IOS-XR-invmg oper:Bestand/Entität tribute/env-sensor-info/value
entSensorThresholdEvaluation	1.3.6.1.4.1.9.9.91.1.2.1.1.5	Diese Variable gibt das Ergebnis der letzten Bewertung des Schwellenwerts an. Wenn	Cisco-IOS-XR-invmg oper:Bestand/Entität tribute/Schwellenwe

		die Schwellenbedingung true ist, ist entSensorThresholdEvaluation true(1). Wenn die Schwellenbedingung false ist, ist entSensorThresholdEvaluation false(2). Schwellenwerte werden mit der von entSensorValueUpdateRate angegebenen Rate ausgewertet.	

CISCO-FLASH-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensensoren für Flash-Speicher im System eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
ciscoFlashPartitionName	1.3.6.1.4.1.9.9.10.1.1.4.1.1.10	Der Name der Flash-Partition, der verwendet wird, um auf eine Partition des Systems zu verweisen. Dabei kann es sich um eine beliebige alphanumerische Zeichenfolge im Formular AAAAAAAAAnn handeln, wobei A ein optionales alphanumerisches Zeichen und ein numerisches Zeichen darstellt. Alle numerischen Zeichen müssen immer der nachfolgende Teil der Zeichenfolge sein. Das System entfernt die alphanumerischen Zeichen und verwendet den numerischen Teil, um einem Partitionsindex zuzuordnen. Flash-Operationen werden an eine Gerätepartition weitergeleitet, die auf diesem Namen basiert. Das System hat ein Konzept für eine Standardpartition. Dies ist die erste Partition auf dem Gerät. Das System leitet einen Vorgang an die Standardpartition weiter, wenn kein Partitionsname angegeben wird. Der	Cisco-IOS-XR-shellutil-filesyst oper:file-system/node/fil system/type

		Partitionsname ist daher obligatorisch, es sei denn, der Vorgang wird auf der Standardpartition ausgeführt, oder das Gerät hat nur eine Partition (ist nicht partitioniert).	
ciscoFlashPartitionSizeExtended	1.3.6.1.4.1.9.9.10.1.1.4.1.1.13	Größe der Flash-Partition. Sie sollte ein ganzes Vielfaches von ciscoFlashDeviceMinPartitionSize sein. Wenn eine einzelne Partition vorhanden ist, entspricht diese Größe ciscoFlashDeviceSize. Dieses Objekt ist eine 64-Bit-Version von ciscoFlashPartitionSize	Cisco-IOS-XR-shellutil-filesyst oper:file-system/node/fil system/size
ciscoFlashPartitionFreeSpaceExtended	1.3.6.1.4.1.9.9.10.1.1.4.1.1.14	Freier Speicherplatz innerhalb einer Flash-Partition. Beachten Sie, dass die tatsächliche Größe einer Datei in Flash einen kleinen Overhead enthält, der den Dateikopf des Dateisystems darstellt. Bestimmte Dateisysteme verfügen möglicherweise auch über einen Partition- oder Geräte-Header-Overhead, der bei der Berechnung des freien Speicherplatzes berücksichtigt werden muss. Freier Speicherplatz wird berechnet als Gesamtpartitionsgröße weniger Größe aller vorhandenen Dateien (gültige/ungültige/gelöschte Dateien und einschließlich Dateiheader jeder Datei), weniger Größe aller Partitionsheader, weniger Größe des Headers der nächsten Datei, in die kopiert werden soll. Kurz gesagt, dieses Objekt gibt die Größe der größten Datei, in die kopiert werden kann. Von der Verwaltungseinheit wird nicht erwartet, dass sie Overhead wie Datei- und Partitionsheaderlängen	Cisco-IOS-XR-shellutil-filesyst oper:file-system/node/fil system/free

		kennt oder nutzt, da diese Overhead von Dateisystem zu Dateisystem variieren kann. Gelöschte Dateien in Flash lassen keinen Speicherplatz frei. Eine Partition muss möglicherweise gelöscht werden, um den von Dateien belegten Speicherplatz freizugeben. Dieses Objekt ist eine 64-Bit-Version von ciscoFlashPartitionFreeSpace	

CISCO-PROCESS-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensoren in Bezug auf CPU-Nutzung und Ressourcenzuweisung für Prozesse eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
cpmCPUTotal1minRev	1.3.6.1.4.1.9.9.109.1.1.1.1.7	Der prozentuale Gesamtanteil der CPU-Auslastung in der letzten Minute. Dieses Objekt veraltet das Objekt cpmCPUTotal1min und erhöht den Wertebereich auf (0.100).	Cisco-IOS-XR-wdsysmon-fd-oper:Systemüberwa g/CPU-Auslastung/ cpu-a-minute
cpmCPUTotal5minRev	1.3.6.1.4.1.9.9.109.1.1.1.1.8	Der prozentuale Gesamtanteil der CPU-Auslastung in den letzten fünf Minuten. Dieses Objekt veraltet das Objekt cpmCPUTotal5min und erhöht den Wertebereich auf (0.100).	Cisco-IOS-XR-wdsysmon-fd-oper:Systemüberwa g/CPU-Auslastung/ cpu-five minute
cpmCPUTotal15minRev	1.3.6.1.4.1.9.9.109.1.1.1.1.31	Der prozentuale Gesamtanteil der CPU-Auslastung in den letzten 15 Minuten. Dieses Objekt veraltet das Objekt cpmCPUTotal15min und erhöht den Wertebereich auf (0.100).	Cisco-IOS-XR-wdsysmon-fd-oper:Systemüberwa g/CPU-Auslastung/ cpu-fünfzehn Minut
cpmProcessName	1.3.6.1.4.1.9.9.109.1.2.1.1.2	Der diesem Prozess zugeordnete Name. Wenn der Name länger als 32 Zeichen ist, wird er auf die ersten 31 Zeichen gekürzt, und ein `*' wird als letztes Zeichen angehängt, um	Cisco-IOS-XR-wdsysmon-fd-oper:Systemüberwa g/CPU-Auslastung/Prozess CPU/Prozessname

		darauf hinzuweisen, dass es sich um einen abgeschnittenen Prozessnamen handelt.	
cpmProcessTextSegmentSize	1.3.6.1.4.1.9.9.109.1.2.3.1.15	Zeigt den Textspeicher eines Prozesses und alle gemeinsam genutzten Objekte an.	Cisco-IOS-XR-proc oper:process-memory/knoten/noc cess-ids/process-id seg-size
cpmProcessDynamicMemorySize	1.3.6.1.4.1.9.9.109.1.2.3.1.18	Dies gibt die Menge des dynamischen Speichers an, der vom Prozess verwendet wird.	Cisco-IOS-XR-proc oper:process-memory/knoten/noc cess-ids/process-id limit
cpmProcessDataSegmentSize	1.3.6.1.4.1.9.9.109.1.2.3.1.16	Dies gibt das Datensegment eines Prozesses und alle seine freigegebenen Objekte an.	Cisco-IOS-XR-proc oper:process-memory/knoten/noc cess-ids/process-id seg-size
cpmProcExtMemAllocationsRev	1.3.6.1.4.1.9.9.109.1.2.3.1.1	Die Summe des dynamisch zugewiesenen Speichers, den dieser Prozess vom System erhalten hat. Dies schließt Speicher ein, die möglicherweise zurückgegeben wurden. Die Summe des freien Speichers wird von cpmProcExtMemFreedRev bereitgestellt. Dieses Objekt veraltet cpmProcExtMemAllocation.	Cisco-IOS-XR-proc oper:process-memory/knoten/noc cess-ids/process-id
cpmProcExtMemFreedRev	1.3.6.1.4.1.9.9.109.1.2.3.1.2	Die Summe des Speichers, den dieser Prozess an das System zurückgegeben hat. Dieses Objekt veraltet cpmProcExtMemFreed.	Cisco-IOS-XR-proc oper:process-memory/knoten/noc cess-ids/process-id

ENTITY-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensoren für mit dem System verbundene physische Einheiten eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
entPhysicalName	1.3.6.1.2.1.47.1.1.1.1.7	Der Textname der physischen Einheit. Der Wert dieses Objekts sollte der Name der Komponente sein, der vom lokalen Gerät zugewiesen wurde, und für die Verwendung in Befehlen	Cisco-IOS-XR-snmp-entitymib-oper:entity-physisch-index

		geeignet sein, die in der "Konsole" des Geräts eingegeben werden. Dies kann ein Textname sein, z. B. `console' oder eine einfache Komponentennummer (z. B. Port- oder Modulnummer), z. B. `1', abhängig von der Syntax für die physische Komponentenbenennung des Geräts. Wenn kein lokaler Name vorhanden ist oder dieses Objekt anderweitig nicht anwendbar ist, enthält dieses Objekt eine Zeichenfolge mit 0-Länge. Beachten Sie, dass der Wert von entPhysicalName für zwei physische Einheiten derselbe ist, falls die Konsolenschnittstelle nicht zwischen ihnen unterscheidet, z. B. Steckplatz-1 und die Karte in Steckplatz-1.	
entLogicalDescr	1.3.6.1.2.1.47.1.2.1.1.2	Eine Textbeschreibung der logischen Einheit. Dieses Objekt sollte eine Zeichenfolge enthalten, die den Namen des Herstellers für die logische Einheit identifiziert, und für jede Version der logischen Einheit sollte ein eindeutiger Wert festgelegt werden.	Cisco-IOS-XR-snmp-agent-oper:snmp/information em name/
entPhysicalDescr	1.3.6.1.2.1.47.1.1.1.1.2	Eine Textbeschreibung der physischen Einheit. Dieses Objekt sollte eine Zeichenfolge enthalten, die den Namen des Herstellers für die physische Einheit identifiziert, und sollte für jede Version oder jedes Modell der physischen Einheit auf einen eindeutigen Wert festgelegt werden.	Cisco-IOS-XR-snmp-agent-oper:snmp/Cisc IOS-XR-snmp-entitym oper:entity-mib/entity-physische indizes/
entPhysicalContainedIn	1.3.6.1.2.1.47.1.1.1.1.4	Der Wert von entPhysicalIndex für die physische Einheit, die diese physische Einheit "enthält". Ein Wert von Null bedeutet, dass diese physische Einheit nicht in einer anderen	Cisco-IOS-XR-invmgr oper:Bestand/Entitäte bute/inv-basic-bag/un id

		physischen Einheit enthalten ist. Beachten Sie, dass der Satz von Containment-Beziehungen eine strikte Hierarchie definiert; d. h. Rekursion ist nicht zulässig. Falls eine physische Einheit von mehr als einer physischen Einheit enthalten ist (z. B. Module mit doppelter Breite), sollte dieses Objekt die enthaltenden Einheiten mit dem niedrigsten Wert von entPhysicalIndex identifizieren.	
entPhysicalClass	1.3.6.1.2.1.47.1.1.1.1.5	Angabe des allgemeinen Hardwaretyps der physischen Einheit. Ein Agent sollte dieses Objekt auf den Standardenumerationswert festlegen, der die allgemeine Klasse der physischen Einheit am genauesten angibt, oder die primäre Klasse, wenn mehr als eine vorhanden ist. Wenn für diese physische Einheit keine geeignete Standardregistrierungskennung vorhanden ist, wird der Wert 'other(1)' zurückgegeben. Wenn der Wert von diesem Agent unbekannt ist, wird der Wert 'known(2)' zurückgegeben.	Cisco-IOS-XR-invmgr oper:Bestand/Einheite
entPhysicalHardwareRev	1.3.6.1.2.1.47.1.1.1.1.8	Die anbieterspezifische Hardware-Revisionszeichenfolge für die physische Einheit. Der bevorzugte Wert ist die Hardware-Revisionskennung, die tatsächlich auf der Komponente selbst gedruckt wurde (sofern vorhanden). Wenn Revisionsinformationen intern in einem nicht druckbaren (z. B. binären) Format gespeichert werden, muss der Agent diese Informationen in ein druckbares Format umwandeln, und zwar auf implementierungsspezifische Weise. Wenn der physischen	Cisco-IOS-XR-invmgr oper:Bestand/Entitäte ibute/inv-basic-bag/hardwarerevision

		Komponente keine spezielle Hardware-Revisionszeichenfolge zugeordnet ist oder diese Informationen dem Agenten nicht bekannt sind, enthält dieses Objekt eine Zeichenfolge mit der Länge 0 (null).	
entPhysicalFirmwareRev	1.3.6.1.2.1.47.1.1.1.1.9	Die anbieterspezifische Firmware-Revisionszeichenfolge für die physische Einheit. Wenn Revisionsinformationen intern in einem nicht druckbaren (z. B. binären) Format gespeichert werden, muss der Agent diese Informationen in ein druckbares Format umwandeln, und zwar auf implementierungsspezifische Weise. Wenn der physischen Komponente keine spezifischen Firmware-Programme zugeordnet sind oder diese Informationen dem Agenten nicht bekannt sind, enthält dieses Objekt eine Zeichenfolge mit 0-Länge.	Cisco-IOS-XR-invmgr oper:Bestand/Entitäte ibute/inv-basic-bag/firmware-revision
entPhysicalSoftwareRev	1.3.6.1.2.1.47.1.1.1.1.10	Die anbieterspezifische Versionszeichenfolge der Software für die physische Einheit. Wenn Revisionsinformationen intern in einem nicht druckbaren (z. B. binären) Format gespeichert werden, muss der Agent diese Informationen in ein druckbares Format umwandeln, und zwar auf implementierungsspezifische Weise. Wenn der physischen Komponente keine spezifischen Softwareprogramme zugeordnet sind oder diese Informationen dem Agenten nicht bekannt sind, enthält dieses Objekt eine Zeichenfolge mit der Länge 0 (null).	Cisco-IOS-XR-invmgr oper:Bestand/Entitäte ibute/inv-basic-bag/software-revision

entPhysicalSerialNum	1.3.6.1.2.1.47.1.1.1.1.11	Die anbieterspezifische Seriennummer-Zeichenfolge für die physische Einheit. Der bevorzugte Wert ist die Seriennummer-Zeichenfolge, die tatsächlich auf der Komponente selbst gedruckt wurde (sofern vorhanden). Bei der ersten Instanziierung einer physischen Einheit wird der Wert von entPhysicalSerialNum, der dieser Entität zugeordnet ist, auf die richtige Seriennummer des Anbieters festgelegt, sofern diese Informationen dem Agenten zur Verfügung stehen. Wenn eine Seriennummer unbekannt ist oder nicht vorhanden ist, wird stattdessen die entPhysicalSerialNum-Zeichenfolge auf eine Zeichenfolge der Länge 0 (null) festgelegt. Beachten Sie, dass Implementierungen, die die Seriennummern aller installierten physischen Einheiten korrekt identifizieren können, keinen Schreibzugriff auf das entPhysicalSerialNum-Objekt bereitstellen müssen. Agenten, die keinen nichtflüchtigen Speicher für die entPhysicalSerialNum-Zeichenfolgen bereitstellen können, müssen den Schreibzugriff für dieses Objekt nicht implementieren. Nicht jede physische Komponente verfügt über eine Seriennummer oder benötigt sogar eine. Physische Einheiten, für die der zugeordnete Wert des entPhysicalIsFRU-Objekts 'false(2)' ist (z. B. die Repeater-Ports in einem Repeater-Modul), benötigen keine eigene eindeutige Seriennummer. Ein Agent muss für diese Entitäten	Cisco-IOS-XR-invmgr oper:Bestand/Einheite heiten/Attribute/inv-ba bag/Seriennummer

		keinen Schreibzugriff bereitstellen und kann eine Zeichenfolge mit 0-Länge zurückgeben. Wenn Schreibzugriff für eine Instanz von entPhysicalSerialNum implementiert wird und ein Wert in die Instanz geschrieben wird, muss der Agent den angegebenen Wert in der mit derselben physischen Entität verknüpften entPhysicalSerialNum-Instanz beibehalten, solange diese Entität instanziiert bleibt. Dies umfasst Instanziierungen für alle Neuinitialisierungen/Neustarts des Netzwerkmanagementsystems, einschließlich solcher, die zu einer Änderung des entPhysicalIndex-Werts der physischen Einheit führen.	
entPhysicalFgName	1.3.6.1.2.1.47.1.1.1.1.12	Der Name des Herstellers dieser physischen Komponente. Der bevorzugte Wert ist der Name des Herstellers, der tatsächlich auf der Komponente selbst gedruckt wurde (sofern vorhanden). Beachten Sie, dass Vergleiche zwischen Instanzen von entPhysicalModelName, entPhysicalFirmwareRev, entPhysicalSoftwareRev und den entPhysicalSerialNum-Objekten nur unter entPhysicalEntries mit dem gleichen Wert von entPhysicalMfgName sinnvoll sind. Wenn dem Agenten die Zeichenfolge des Herstellernamens, die der physischen Komponente zugeordnet ist, nicht bekannt ist, enthält dieses Objekt eine Zeichenfolge mit 0-Länge.	Cisco-IOS-XR-invmgr oper:Bestand/Einheite heiten/Attribute/inv-ba bag/Herstellername
entPhysicalModelName	1.3.6.1.2.1.47.1.1.1.1.13	Die anbieterspezifische Modellbezeichnung-ID-Zeichenfolge, die dieser	Cisco-IOS-XR-invmgr oper:Bestand/Einheite heiten/Attribute/inv-ba

		physischen Komponente zugeordnet ist. Der bevorzugte Wert ist die vom Kunden angezeigte Teilenummer, die auf der Komponente selbst gedruckt werden kann. Wenn die Modellnamenszeichenfolge, die der physischen Komponente zugeordnet ist, dem Agenten nicht bekannt ist, enthält dieses Objekt eine Zeichenfolge mit der Länge 0 (null).	bag/model-name

IF-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensensoren für Schnittstellenmerkmale und Zähler eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
ifMtu	1.3.6.1.2.1.2.2.1.4	Die Größe des größten Pakets, das auf der Schnittstelle gesendet/empfangen werden kann, angegeben in Oktetten. Bei Schnittstellen, die zur Übertragung von Netzwerkdatagrammen verwendet werden, ist dies die Größe des größten Netzwerkdatagramms, das auf der Schnittstelle gesendet werden kann.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/mtu
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	Die Adresse der Schnittstelle in ihrer Protokoll-Unterschicht. Beispielsweise enthält dieses Objekt für eine 802.x-Schnittstelle normalerweise eine MAC-Adresse. Die medienspezifische MIB der Schnittstelle muss die Bit- und Bytereihenfolge sowie das Format des Werts dieses Objekts definieren. Bei Schnittstellen, die keine solche Adresse haben (z. B. eine serielle Leitung), sollte dieses Objekt eine Oktettzeichenfolge von Null-Länge enthalten.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface-ty information/bündel-information/member/m address
ifType	1.3.6.1.2.1.2.2.1.3	Der Schnittstellentyp. Zusätzliche Werte für ifType	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac

		werden von der Internet Assigned Numbers Authority (IANA) durch Aktualisieren der Syntax der Textuellen IANAifType-Konvention zugewiesen.	xr/interface/interface/ty
ifOutUcastPkte	1.3.6.1.2.1.2.2.1.17	Die Gesamtzahl der Pakete, die von den übergeordneten Protokollen übertragen werden und nicht an eine Multicast- oder Broadcast-Adresse auf dieser Unterschicht adressiert wurden, einschließlich Pakete, die verworfen oder nicht gesendet wurden. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface-statistics/full-interface-stats/packages-sent
ifHCOutUcastPkte	1.3.6.1.2.1.31.1.1.1.11	Die Gesamtzahl der Pakete, die von den übergeordneten Protokollen übertragen werden und nicht an eine Multicast- oder Broadcast-Adresse auf dieser Unterschicht adressiert wurden, einschließlich Pakete, die verworfen oder nicht gesendet wurden. Dieses Objekt ist eine 64-Bit-Version von ifOutUcastPkts. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface-statistics/full-interface-stats/packages-sent
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	Die Anzahl der Pakete, die von dieser Unterschicht an eine höhere (Sub-)Schicht geliefert wurden und nicht an eine Multicast- oder Broadcast-Adresse auf dieser Unterschicht adressiert waren. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface-statistics/full-interface-stats/received

		angegebenen Zeiten auftreten.	
ifHCInUcastPkte	1.3.6.1.2.1.31.1.1.1.7	Die Anzahl der Pakete, die von dieser Unterschicht an eine höhere (Sub-)Schicht geliefert wurden und nicht an eine Multicast- oder Broadcast-Adresse auf dieser Unterschicht adressiert waren. Dieses Objekt ist eine 64-Bit-Version von ifInUcastPkts. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface-statistics/full-interface-stats/received
ifOutErrors	1.3.6.1.2.1.2.2.1.20	Bei paketorientierten Schnittstellen die Anzahl ausgehender Pakete, die aufgrund von Fehlern nicht übertragen werden konnten. Bei zeichenorientierten Schnittstellen oder Schnittstellen mit fester Länge die Anzahl der ausgehenden Übertragungseinheiten, die aufgrund von Fehlern nicht übertragen werden konnten. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	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	Die Anzahl der ausgehenden Pakete, die als verworfen ausgewählt wurden, obwohl keine Fehler erkannt wurden, um deren Übertragung zu verhindern. Ein möglicher Grund dafür, ein solches Paket zu verwerfen, könnte darin bestehen, Speicherplatz freizugeben. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface-statistics/full-interface-stats/output-drop
ifOutMulticastPkts	1.3.6.1.2.1.31.1.1.1.4	Die Gesamtzahl der Pakete,	Cisco-IOS-XR-pfi-im-c

		die von den übergeordneten Protokollen übertragen werden und an eine Multicast-Adresse auf dieser Unterschicht adressiert wurden, einschließlich Pakete, die verworfen oder nicht gesendet wurden. Für ein MAC-Layer-Protokoll umfasst dies sowohl Gruppen- als auch Funktionsadressen. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	oper:interfaces/interfac xr/interface/interface-statistics/full-interface-stats/multicast-packag sent
ifHCOutMulticastPkte	1.3.6.1.2.1.31.1.1.1.12	Die Gesamtzahl der Pakete, die von den übergeordneten Protokollen übertragen werden und an eine Multicast-Adresse auf dieser Unterschicht adressiert wurden, einschließlich Pakete, die verworfen oder nicht gesendet wurden. Für ein MAC-Layer-Protokoll umfasst dies sowohl Gruppen- als auch Funktionsadressen. Dieses Objekt ist eine 64-Bit-Version von ifOutMulticastPkts. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface-statistics/full-interface-stats/multicast-packag sent
ifInMulticastPkts	1.3.6.1.2.1.31.1.1.1.2	Die Anzahl der Pakete, die von dieser Unterschicht an eine höhere (Sub-)Schicht geliefert wurden und an eine Multicast-Adresse auf dieser Unterschicht adressiert wurden. Für ein MAC-Layer-Protokoll umfasst dies sowohl Gruppen- als auch Funktionsadressen. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface-statistics/full-interface-stats/multicast-packag received

ifHCInMulticastPkts	1.3.6.1.2.1.31.1.1.1.8	anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten. Die Anzahl der Pakete, die von dieser Unterschicht an eine höhere (Sub-)Schicht geliefert wurden und an eine Multicast-Adresse auf dieser Unterschicht adressiert wurden. Für ein MAC-Layer-Protokoll umfasst dies sowohl Gruppen- als auch Funktionsadressen. Dieses Objekt ist eine 64-Bit-Version von ifInMulticastPkts. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/multicast-packag received
ifInErrors	1.3.6.1.2.1.2.2.1.14	Bei paketorientierten Schnittstellen die Anzahl der eingehenden Pakete, die Fehler enthielten, die die Bereitstellung an ein übergeordnetes Protokoll verhinderten. Bei zeichenorientierten Schnittstellen oder Schnittstellen mit fester Länge die Anzahl der eingehenden Übertragungseinheiten, die Fehler enthielten, die die Bereitstellung an ein übergeordnetes Protokoll verhinderten. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	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	Die Anzahl der eingehenden Pakete, die als verworfen ausgewählt wurden, obwohl keine Fehler erkannt wurden, um deren Übermittlung an ein Protokoll der höheren Schicht zu verhindern. Ein möglicher Grund dafür, ein solches Paket zu verwerfen, könnte darin	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/input-drops

		bestehen, Speicherplatz freizugeben. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	
ifOutOctets	1.3.6.1.2.1.2.2.1.16	Die Gesamtzahl der über die Schnittstelle übertragenen Oktette, einschließlich Framing-Zeichen. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/bytes-sent
ifHCOutOctets	1.3.6.1.2.1.31.1.1.1.10	Die Gesamtzahl der über die Schnittstelle übertragenen Oktette, einschließlich Framing-Zeichen. Dieses Objekt ist eine 64-Bit-Version von ifOutOctets. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	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	Die Gesamtzahl der Oktette, die auf der Schnittstelle empfangen wurden, einschließlich Framing-Zeichen. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/bytes-received
ifHCInOctets	1.3.6.1.2.1.31.1.1.1.6	Die Gesamtzahl der Oktette, die auf der Schnittstelle empfangen wurden, einschließlich Framing-Zeichen. Dieses Objekt ist eine 64-Bit-Version von ifInOctets. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/bytes-received

ifOutBroadcastPkte	1.3.6.1.2.1.31.1.1.1.5	anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten. Die Gesamtzahl der Pakete, die von den übergeordneten Protokollen übertragen werden und an eine Broadcast-Adresse auf dieser Unterschicht adressiert wurden, einschließlich Pakete, die verworfen oder nicht gesendet wurden. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/Broadcast-Packe sent
ifHCOutBroadcast-Pkte	1.3.6.1.2.1.31.1.1.1.13	Die Gesamtzahl der Pakete, die von den übergeordneten Protokollen übertragen werden und an eine Broadcast-Adresse auf dieser Unterschicht adressiert wurden, einschließlich Pakete, die verworfen oder nicht gesendet wurden. Dieses Objekt ist eine 64-Bit-Version von ifOutBroadcastPkts. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/Broadcast-Packe sent
ifInBroadcastPkts	1.3.6.1.2.1.31.1.1.1.3	Die Anzahl der Pakete, die von dieser Unterschicht an eine höhere (Unter-)Schicht übermittelt wurden und an eine Broadcast-Adresse auf dieser Unterschicht adressiert wurden. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/Broadcast-Packe received
ifHCInBroadcast-Pkte	1.3.6.1.2.1.31.1.1.1.9	Die Anzahl der Pakete, die von dieser Unterschicht an eine höhere (Unter-)Schicht	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface-

		übermittelt wurden und an eine Broadcast-Adresse auf dieser Unterschicht adressiert wurden. Dieses Objekt ist eine 64-Bit-Version von ifInBroadcastPkts. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ifCounterDisContinuityTime angegebenen Zeiten auftreten.	statistics/full-interface-stats/Broadcast-Packe received
ifIndex	1.3.6.1.2.1.2.2.1.1	Ein eindeutiger Wert größer als Null für jede Schnittstelle. Es wird empfohlen, Werte fortlaufend ab 1 zuzuweisen. Der Wert für jede Schnittstellenunterschicht muss mindestens von einer Neuinitialisierung des Netzwerkmanagementsystems der Einheit bis zur nächsten Neuinitialisierung konstant bleiben.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/if-index
ifDescr	1.3.6.1.2.1.2.2.1.2	Eine Textzeichenfolge, die Informationen über die Schnittstelle enthält. Diese Zeichenfolge sollte den Namen des Herstellers, den Produktnamen und die Version der Schnittstellenhardware/-software enthalten.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/description
ifSpeed	1.3.6.1.2.1.2.2.1.5	Eine Schätzung der aktuellen Bandbreite der Schnittstelle in Bit pro Sekunde. Für Schnittstellen, die nicht in der Bandbreite variieren, oder für Schnittstellen, für die keine genaue Schätzung vorgenommen werden kann, sollte dieses Objekt die nominale Bandbreite enthalten. Wenn die Bandbreite der Schnittstelle den von diesem Objekt gemeldeten maximalen Wert übersteigt, sollte dieses Objekt seinen maximalen Wert (4.294.967.295) melden und ifHighSpeed muss zum Melden der Geschwindigkeit der Schnittstelle verwendet	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/bandwidth

		werden. Bei einer Unterschicht ohne Konzept der Bandbreite sollte dieses Objekt null sein.	
ifOperStatus	1.3.6.1.2.1.2.2.1.8	Der aktuelle Betriebsstatus der Schnittstelle. Der Status test(3) gibt an, dass keine betrieblichen Pakete übergeben werden können. Wenn ifAdminStatus inaktiv ist(2), dann sollte ifOperStatus deaktiviert sein(2). Wenn ifAdminStatus auf up(1) geändert wird, sollte ifOperStatus auf up(1) geändert werden, wenn die Schnittstelle für das Senden und Empfangen von Netzwerkverkehr bereit ist. Wenn die Schnittstelle auf externe Aktionen wartet (z. B. eine serielle Leitung, die auf eine eingehende Verbindung wartet), sollte sie in dormant(5) geändert werden. Der Zustand "down(2)" sollte nur dann beibehalten werden, wenn ein Fehler vorliegt, der den Wechsel in den Status "up(1)" verhindert. Es sollte im Zustand "notPresent(6)" bleiben, wenn für die Schnittstelle (in der Regel Hardware)-Komponenten fehlen.	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac non-dynamik/interface dynamic/oper-state
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	Der gewünschte Status der Schnittstelle. Der Status test(3) gibt an, dass keine betrieblichen Pakete übergeben werden können. Wenn ein verwaltetes System initialisiert wird, beginnen alle Schnittstellen mit ifAdminStatus im down(2)-Status. Als Ergebnis einer expliziten Verwaltungsaktion oder einer vom verwalteten System aufbewahrten Konfigurationsinformation wird ifAdminStatus entweder in den Status "up(1)" oder "testing(3)" geändert (oder bleibt im Zustand "down(2)").	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac non-dynamik/interface dynamic/admin-state
ifName	1.3.6.1.2.1.31.1.1.1.1	Der Textname der	Cisco-IOS-XR-pfi-im-c

		Schnittstelle. Der Wert dieses Objekts sollte der Name der Schnittstelle sein, der vom lokalen Gerät zugewiesen wurde, und für die Verwendung in Befehlen geeignet sein, die in der "Konsole" des Geräts eingegeben werden. Dies kann ein Textname sein, z. B. `le0', oder eine einfache Portnummer, z. B. `1', abhängig von der Schnittstellenbenennungssyntax des Geräts. Wenn mehrere Einträge in der ifTable zusammen eine einzige Schnittstelle darstellen, die vom Gerät benannt wird, hat jeder den gleichen Wert wie ifName. Beachten Sie, dass bei einem Agenten, der auf SNMP-Abfragen bezüglich einer Schnittstelle auf einem anderen (proxitierten) Gerät reagiert, der Wert von ifName für eine solche Schnittstelle der lokale Name des proxiierten Geräts für diese Schnittstelle ist. Wenn kein lokaler Name vorhanden ist oder dieses Objekt anderweitig nicht anwendbar ist, enthält dieses Objekt eine Zeichenfolge mit 0-Länge.	oper:interfaces/interfac briefs/interface-brief/interface-name
ifHighSpeed	1.3.6.1.2.1.31.1.1.1.15	Eine Schätzung der aktuellen Bandbreite der Schnittstelle in Einheiten von 1.000.000 Bit pro Sekunde. Wenn dieses Objekt einen Wert von 'n' meldet, liegt die Geschwindigkeit der Schnittstelle im Bereich von 'n-500,000' bis 'n+499,999'. Für Schnittstellen, die nicht in der Bandbreite variieren, oder für Schnittstellen, für die keine genaue Schätzung vorgenommen werden kann, sollte dieses Objekt die nominale Bandbreite enthalten. Bei einer Unterschicht ohne Konzept der	Cisco-IOS-XR-pfi-im-c oper:Interfaces/interfac briefs/interface-brief/bandwidth64-Bit

		Bandbreite sollte dieses Objekt null sein.	

IP-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetrie-Sensorgruppen im Zusammenhang mit IP-Statistiken und Betriebswerten eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
icmpInDestUnreachs	1.3.6.1.2.1.5.3	Die Anzahl der empfangenen ICMP-Zielmeldungen ohne Erreichbarkeit.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/scs/traffic/icmp-stats
icmpInParmProbs	1.3.6.1.2.1.5.5	Die Anzahl der empfangenen ICMP-Parameterproblemmeldungen.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/scs/traffic/icmp-stats
icmpInSrcQuenchs	1.3.6.1.2.1.5.6	Die Anzahl der empfangenen ICMP Source Quench-Nachrichten	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/scs/traffic/icmp-stats
ICMPInEchos	1.3.6.1.2.1.5.8	Die Anzahl der empfangenen ICMP-Echo-Nachrichten (Anfrage)	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/scs/traffic/icmp-stats
icmpInEchoReps	1.3.6.1.2.1.5.9	Die Anzahl der empfangenen ICMP-Echo-Antwortnachrichten.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/scs/traffic/icmp-stats
icmpInTimestamps	1.3.6.1.2.1.5.10	Die Anzahl der empfangenen ICMP-Timestamp-Nachrichten (Anfrage).	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/scs/traffic/icmp-stats
icmpInAddrMasken	1.3.6.1.2.1.5.12	Die Anzahl der empfangenen ICMP-Adressenmaske-Anforderungsnachrichten.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/scs/traffic/icmp-stats
icmpInAddrMaskReps	1.3.6.1.2.1.5.13	Die Anzahl der empfangenen ICMP-Adressenmasken-Antwortnachrichten.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/scs/traffic/icmp-stats
icmpOutMsgs	1.3.6.1.2.1.5.14	Die Gesamtzahl der ICMP-Nachrichten, die diese Entität zu senden versuchte. Beachten Sie, dass dieser Zähler alle Zähler enthält, die von icmpOutErrors gezählt werden.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/scs/traffic/icmp-stats
icmpOutDestUnreachs	1.3.6.1.2.1.5.16	Die Anzahl der nicht erreichbaren ICMP-Zielnachrichten, die gesendet werden.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/scs/traffic/icmp-stats

icmpOutTimeExds	1.3.6.1.2.1.5.17	Die Anzahl der gesendeten ICMP-Nachrichten, die die Zeit überschritten haben.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/s cs/traffic/icmp-stats
icmpOutParmProbs	1.3.6.1.2.1.5.18	Die Anzahl der gesendeten ICMP-Parameterproblemmeldungen.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/s cs/traffic/icmp-stats
icmpOutSrcQuenchs	1.3.6.1.2.1.5.19	Die Anzahl der gesendeten ICMP Source Quench-Nachrichten	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/s cs/traffic/icmp-stats
icmpOutRedirects	1.3.6.1.2.1.5.20	Die Anzahl der gesendeten ICMP-Umleitungsnachrichten. Für einen Host ist dieses Objekt immer 0, da Hosts keine Umleitungen senden.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/s cs/traffic/icmp-stats
ICMPOutEchos	1.3.6.1.2.1.5.21	Die Anzahl der gesendeten ICMP-Echo-Nachrichten (Anfrage).	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/s cs/traffic/icmp-stats
icmpOutEchoReps	1.3.6.1.2.1.5.22	Die Anzahl der gesendeten ICMP-Echo-Antwortnachrichten.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/s cs/traffic/icmp-stats
icmpOutTimestamps	1.3.6.1.2.1.5.23	Die Anzahl der gesendeten ICMP-Timestamp-Nachrichten (Anfrage).	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/s cs/traffic/icmp-stats
icmpOutAddrMasken	1.3.6.1.2.1.5.25	Die Anzahl der gesendeten ICMP-Adressenmaske-Anforderungsnachrichten.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/s cs/traffic/icmp-stats
icmpOutAddrMaskReps	1.3.6.1.2.1.5.26	Die Anzahl der gesendeten ICMP-Adressenmasken-Antwortnachrichten.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/s cs/traffic/icmp-stats
ipAdEntIfIndex	1.3.6.1.2.1.4.20.1.2	Der Indexwert, der die Schnittstelle, auf die dieser Eintrag anwendbar ist, eindeutig kennzeichnet. Die Schnittstelle, die durch einen bestimmten Wert dieses Index identifiziert wird, ist die gleiche Schnittstelle, die durch denselben Wert wie der ifIndex von RFC 1573 identifiziert wird.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/
ipAdEntAddr	1.3.6.1.2.1.4.20.1.1	Die IP-Adresse, auf die sich die Adressinformationen dieses Eintrags beziehen.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/interfaces/inte vrfs/vrf/detail/primary-address
ipAdEntNetMask	1.3.6.1.2.1.4.20.1.3	Die Subnetzmaske, die der IP-	Cisco-IOS-XR-ipv4-io-

ipAdEntBcastAddr	1.3.6.1.2.1.4.20.1.4	Adresse dieses Eintrags zugeordnet ist. Der Wert der Maske ist eine IP-Adresse, bei der alle Netzwerkbits auf 1 und alle Hosts auf 0 festgelegt sind. Der Wert des Bits mit der geringsten Bedeutung in der IP-Broadcast-Adresse, die zum Senden von Datagrammen an der (logischen) Schnittstelle verwendet wird, die der IP-Adresse dieses Eintrags zugeordnet ist. Wenn z. B. die standardmäßige All-One-Broadcast-Adresse des Internets verwendet wird, ist der Wert 1. Dieser Wert gilt sowohl für die Subnetz- als auch die Netzwerk-Broadcast-Adressen, die von der Entität auf dieser (logischen) Schnittstelle verwendet werden.	oper:ipv4-network/interfaces/inter vrfs/vrf/detail/prefix-len Cisco-IOS-XR-ipv4-io-oper:ipv4-network/interfaces/inter vrfs/vrf/detail/direct-broadcast
ipNetToMediaPhysAdresse	1.3.6.1.2.1.4.22.1.2	Die medienabhängige "physische" Adresse.	Cisco-IOS-XR-ipv4-arp oper:arp/knoten/knoten es/entry/hardware addr

IPMIB-COMMMON

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensensoren für IP-Statistiken eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
ipIfStatsHCOutTransmits	1.3.6.1.2.1.4.31.3.1.31	Die Gesamtzahl der IP-Datagramme, die diese Einheit den unteren Schichten zur Übertragung bereitstellte. Dieses Objekt zählt dieselben Datagramme wie ipIfStatsOutTransmits, ermöglicht jedoch größere Werte. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ipIfStatsDisContinuityTime angezeigten Zeiten auftreten.	Cisco-IOS-XR-IPv4-io oper:ipv4-network/knoten/node/s ics/traffic/ipv4-stats/Pa weitergeleitet
ipIfStatsInReceives	1.3.6.1.2.1.4.31.3.1.3	Die Gesamtzahl empfangener IP-Datagramme, einschließlich der irrtümlich empfangenen Datagramme. Diskontinuitäten	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/s ics/traffic/ipv4-stats/inp

		im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ipIfStatsDisContinuityTime angezeigten Zeiten auftreten.	packages
ipIfStatsHCInReceives	1.3.6.1.2.1.4.31.3.1.4	Die Gesamtzahl empfangener IP-Datagramme, einschließlich der irrtümlich empfangenen Datagramme. Dieses Objekt zählt dieselben Datagramme wie ipIfStatsInReceives, ermöglicht jedoch größere Werte. Diskontinuitäten im Wert dieses Zählers können bei der Neuinitialisierung des Managementsystems und zu anderen, durch den Wert von ipIfStatsDisContinuityTime angezeigten Zeiten auftreten.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/knoten/node/ics/traffic/ipv4-stats/inpackages

LLDP-MIB

Die nächste Tabelle stellt den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH dar, der auf modellgesteuerten Telemetriesensensoren-Gruppen im Zusammenhang mit LLDP-Betriebsdaten (Link Layer Discovery Protocol) auf dem überwachten Knoten eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
LLDPlocPortID	1.0.8802.1.1.2.1.3.7.1.3	Der Zeichenfolgenwert, der zum Identifizieren der Port-Komponente verwendet wird, die einem bestimmten Port im lokalen System zugeordnet ist.	Cisco-IOS-XR-etherne lldp-oper:lldp/Knoten/Knotchbarn/Geräte/Gerät/LNachbar/Port-ID-Deta
lldpLocPortIdSubtype	1.0.8802.1.1.2.1.3.7.1.2	Der Typ der im zugeordneten 'lldpLocPortId'-Objekt verwendeten Codierung für die Port-ID.	Cisco-IOS-XR-Etherne lldp-oper:lldp/Knoten/Knotchbarn/Geräte/Gerät/LNachbar/mib/port-id-stype
lldpLocChassisIdSubtype	1.0.8802.1.1.2.1.3.1	Der Kodierungstyp zum Identifizieren des Chassis, das dem lokalen System zugeordnet ist.	Cisco-IOS-XR-Etherne lldp-oper:lldp/Knoten/Knotchbarn/Geräte/Gerät/LNachbar/mib/Chassis-Subtyp
lldpLocSysName	1.0.8802.1.1.2.1.3.3	Der Zeichenfolgenwert, der zum Identifizieren des Systemnamens des lokalen Systems verwendet wird. Wenn der lokale Agent IETF	Cisco-IOS-XR-etherne lldp-oper:lldp/knoten/neighdevices/device/lldp-neighbor/detail/system

		RFC 3418 unterstützt, sollte das lldpLocSysName-Objekt denselben Wert wie das sysName-Objekt haben.	name
lldpRemSysName	1.0.8802.1.1.2.1.4.1.1.9	Der Zeichenfolgenwert, der zum Identifizieren des Systemnamens des Remotesystems verwendet wird.	Cisco-IOS-XR-etherne lldp-oper:lldp/knoten/neigh devices/device/lldp-neighbor/detail/system name
lldpRemChassisID	1.0.8802.1.1.2.1.4.1.1.5	Der Zeichenfolgenwert, der zum Identifizieren der Chassis-Komponente verwendet wird, die dem Remote-System zugeordnet ist.	Cisco-IOS-XR-etherne oper:lldp/knoten/node/ bors/devices/device/llc neighbor/chassis-id
lldpRemChassisIdSubtyp	1.0.8802.1.1.2.1.4.1.1.4	Der Kodierungstyp, der zum Identifizieren des Chassis verwendet wird, das dem Remote-System zugeordnet ist.	Cisco-IOS-XR-etherne oper:lldp/knoten/node/ bors/devices/device/llc neighbor
lldpRemPortIdSubtype	1.0.8802.1.1.2.1.4.1.1.6	Der Typ der im zugeordneten 'lldpRemPortId'-Objekt verwendeten Codierung für Port-Bezeichner.	Cisco-IOS-XR-etherne oper:lldp/knoten/node/ bors/devices/device/llc neighbor
LLDPremPortID	1.0.8802.1.1.2.1.4.1.1.7	Der Zeichenfolgenwert, der zum Identifizieren der Port-Komponente verwendet wird, die dem Remote-System zugeordnet ist.	Cisco-IOS-XR-etherne oper:lldp/knoten/node/ bors/devices/device/llc neighbor
LLDPlocChassisID	1.0.8802.1.1.2.1.3.2	Der Zeichenfolgenwert, der zum Identifizieren der Chassis-Komponente verwendet wird, die dem lokalen System zugeordnet ist.	Cisco-IOS-XR-ethern ldp-oper:lldp/knoten/neigh details/detail/lldp-neighbor/chassis-id

MPLS-TE-STD-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensensoren für MPLS-(Multiprotocol Label Switching)-Betriebssysteme für Datenverkehr auf dem verwalteten Gerät eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
mplsTunnelName	1.3.6.1.2.1.10.166.3.2.2.1.5	Der dem Tunnel zugewiesene kanonische Name. Mit diesem Namen kann auf den Tunnel am Konsolenport des LSR verwiesen werden. Wenn mplsTunnelIsIf auf true festgelegt ist, sollte der ifName der Schnittstelle, die	Cisco-IOS-XR-mpls-t oper:mpls-te/p2p-p2r tunnel/tunnel-head/tu head/tunnel-name

		diesem Tunnel entspricht, einen Wert aufweisen, der mplsTunnelName entspricht. Siehe auch die Beschreibung von ifName in RFC 2863.	
mplsTunnelDescr	1.3.6.1.2.1.10.166.3.2.2.1.6	Eine Textzeichenfolge, die Informationen über den Tunnel enthält. Wenn keine Beschreibung vorhanden ist, enthält dieses Objekt eine Zeichenfolge mit der Länge 0 (null). Dieses Objekt wird möglicherweise nicht von MPLS-Signalisierungsprotokollen signalisiert, sodass der Wert dieses Objekts bei der Übertragung und bei LSRs für Ausgangs-Datenverkehr automatisch generiert wird oder nicht.	openconfig-network-instance:network-instance/network-instance/mpls/lsps/co ned-path/tunnels/tunnel/s escription
mplsTunnelPerfHCPackets	1.3.6.1.2.1.10.166.3.2.9.1.2	Zähler mit hoher Kapazität für die Anzahl der durch den Tunnel weitergeleiteten Pakete	openconfig-network-instance:network-instance/network-instance/mpls/lsps/co ned-path/tunnels/tunnel/s ounter/pakete
mplsTunnelPerfHCBytes	1.3.6.1.2.1.10.166.3.2.9.1.5	Leistungsindikator für hohe Kapazität für die Anzahl der über den Tunnel weitergeleiteten Bytes.	openconfig-network-instance:network-instance/network-instance/mpls/lsps/co ned-path/tunnels/tunnel/s ounter/bytes
mplsTunnelHopIPAddr	1.3.6.1.2.1.10.166.3.2.4.1.5	Die Tunnel-Hop-Adresse für diesen Tunnel-Hop. Der Adresstyp wird durch den Wert des entsprechenden mplsTunnelHopAddrType bestimmt. Der Wert dieses Objekts kann nicht geändert werden, wenn der Wert des entsprechenden mplsTunnelHopRowStatus-Objekts 'active' ist.	Cisco-IOS-XR-mpls-t oper:mpls-te/p2p-p2r tunnel/tunnel-head/tu head/destination/des n-address

RFC2465-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensensoren für globale IPv6-Werte eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
ipv6AddrPfxLength	1.3.6.1.2.1.55.1.8.1.2	Die Länge des Präfixes (in Bits), das der IPv6-Adresse dieses Eintrags zugeordnet ist.	Cisco-IOS-XR-ipv6-ma-oper:ipv6-network/knoten/node/int e-data/vrfs/vrf/briefs/brief/ ss/prefix-length
ipv6AddrAnycastFlag	1.3.6.1.2.1.55.1.8.1.4	Dieses Objekt hat den Wert 'true(1)', wenn diese Adresse eine Anycast-Adresse ist, ansonsten den Wert 'false(2)'.	Cisco-IOS-XR-ipv6-ma-oper:ipv6-network/knoten/node/int e-data/vrfs/vrf/briefs/brief/ ss/is-anycast

SNMP-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensensensoren für den SNMP-Agenten selbst eingerichtet werden soll, sofern verfügbar.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
sysUpTime	1.3.6.1.2.1.1.3	Zeichenfolge, die die Systemverfügbarkeit darstellt	Cisco-IOS-XR-snmp-age oper:snmp/information/s -up/
sysObjectID	.1.3.6.1.2.1.1.2.0	Zeichenfolge, die die System-OID darstellt	Cisco-IOS-XR-snmp-age oper:snmp/information/s -oid/
sysDescr	1.3.6.1.2.1.1.1	Zeichenfolge, die die Systembeschreibung darstellt	Cisco-IOS-XR-snmp-age oper:snmp/information/s -abor

TCP-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensensoren für TCP-spezifische Zähler eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
tcpInErs	1.3.6.1.2.1.6.14	Die Gesamtzahl der fehlerhaften Segmente (z. B. fehlerhafte TCP-Prüfsummen).	Cisco-IOS-XR-ip-tcp-oper:tcp/knoten/node/sta s/ipv4-traffic/tcp-checksu error-pakete
tcpInSegs	1.3.6.1.2.1.6.10	Die Gesamtzahl empfangener Segmente, einschließlich fehlerhafter Segmente. Diese Anzahl umfasst Segmente, die auf aktuell eingerichteten Verbindungen empfangen werden.	Cisco IOS-XR-ip-tcp-oper:tcp/knoten/node/sta s/ipv4-traffic/tcp-input-packages
tcpOutSegs	1.3.6.1.2.1.6.11	Die Gesamtzahl der gesendeten Segmente, einschließlich der Segmente mit aktuellen	Cisco IOS-XR-ip-tcp-oper:tcp/knoten/node/sta s/ipv4-traffic/tcp-output-

		Verbindungen, jedoch mit Ausnahme derjenigen, die nur neu übertragene Oktette enthalten.	pakete

UDP-MIB

Die nächste Tabelle enthält den OID-Namen und die OID-Nummer sowie den entsprechenden XPATH, der auf modellgesteuerten Telemetriesensensoren für UDP-spezifische Zähler eingerichtet werden soll.

OID-Name	OID-Nummer	OID-Beschreibung	XPFAD
udpOutDatagrams	1.3.6.1.2.1.7.4	Die Gesamtzahl der von dieser Entität gesendeten UDP-Datagramme.	Cisco-IOS-XR-ip-udp-oper:/udp/knoten/node/cs/ipv4-traffic/udp-outputpakete Cisco-IOS-XR-ip-udp-oper:/udp/knoten/node/cs/ipv6-traffic/udp-outputpakete
udpNoPorts	1.3.6.1.2.1.7.2	Die Gesamtzahl empfangener UDP-Datagramme, für die es am Zielport keine Anwendung gab.	Cisco-IOS-XR-ip-udp-oper:/udp/knoten/node/cs/ipv4-traffic/udp-no-popakete Cisco-IOS-XR-ip-udp-oper:/udp/knoten/node/cs/ipv6-traffic/udp-no-popakete
udpInErrors	1.3.6.1.2.1.7.3	Die Anzahl empfangener UDP-Datagramme, die aus anderen Gründen als dem Fehlen einer Anwendung am Zielport nicht bereitgestellt werden konnten.	Cisco-IOS-XR-ip-udp-oper:/udp/knoten/node/cs/ipv4-traffic/udp-checkerror-pakete Cisco-IOS-XR-ip-udp-oper:/udp/knoten/node/cs/ipv6-traffic/udp-checkerror-pakete
udpInDatagrams	1.3.6.1.2.1.7.1	Die Gesamtzahl der UDP-Datagramme, die UDP-Benutzern bereitgestellt werden.	Cisco-IOS-XR-ip-udp-oper:/udp/knoten/node/cs/ipv4-traffic/udp-input-pakete Cisco-IOS-XR-ip-udp-oper:/udp/knoten/node/cs/ipv6-traffic/udp-input-pakete

Migration von SNMP-Traps

SNMP-Traps sind Meldungen, die durch dynamische Ereignisse auf dem verwalteten Gerät ausgelöst werden. Daher verhalten sich diese Botschaften analog zum Konzept der EDT, das wir zuvor behandelt haben.

Auf der Konfigurationsseite ermöglicht MDT die gleiche Struktur für EDT, die von der

Implementierung des Telemetrie-Collectors im Hinblick auf Dial-In- oder Dial-Out-Optionen oder -Funktionen abhängt.

Sicherheitsüberlegungen

SNMPv2 verwendet nur Community als Authentifizierungs-/Autorisierungsmechanismus. SNMPv3, wie im Abschnitt "SNMP" bereits beschrieben, könnte Anmeldeinformationen für die Authentifizierung und das AES-Verschlüsselungsmodell zum Schutz der Informationen verwenden.

Im Telemetrie-Ansatz ermöglicht IOS XR die Verwendung von gRPC/TLS-Techniken, die auf Zertifikaten basieren, um die Authentifizierung durchzuführen. Diese Zertifikate können mit einem zentralen Vertrauenspunkt (z. B. einem CA-Server) verwendet werden. Nach dem Aufbau einer vertrauenswürdigen Beziehung werden alle Telemetriedaten in einer gRPC-Sitzung verschlüsselt, die mit TLS verschlüsselt wird und dieselben Vorteile wie SNMPv3 bietet.