

IOS XRでのテレメトリへのSNMP移行

内容

[概要](#)

[SNMP](#)

[SNMPのコンポーネント](#)

[SNMPマネージャ](#)

[SNMPエージェント](#)

[SNMP MIB](#)

[SNMPの動作](#)

[MIBとRFC](#)

[SNMPのバージョン](#)

[ヤンモデル](#)

[OpenConfigモデル](#)

[ネイティブモデル](#)

[テレメトリ](#)

[モデル駆動型テレメトリ](#)

[イベント駆動型テレメトリ](#)

[トランスポート](#)

[TCP](#)

[gRPC](#)

[gNMI/gNOI](#)

[符号化](#)

[JSON](#)

[GPB-KV](#)

[GPB](#)

[IOS XRでのMDTの設定](#)

[ダイヤルアウトモード](#)

[ダイヤルインモード](#)

[MDTへのSNMPの移行](#)

[XPATHへのMIBの移行](#)

[BGP4-MIB](#)

[CISCO-BGP4-MIB](#)

[CISCO-CLASS-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](#)

[SNMPトラップの移行](#)

[セキュリティに関する考慮事項](#)

概要

この記事では、Simple Network Management Protocol(SNMP)コンポーネントを紹介し、Model Driven Telemetry(MDT)アプローチへのSNMPモニタリングに基づく現在の実装の間の相関関係を説明します。

SNMP

SNMPは、SNMPマネージャとエージェントの間で通信を行うためのメッセージフォーマットを提供するアプリケーション層のプロトコルです。SNMPは、ネットワーク内のデバイスの監視と管理に使用される、標準化されたフレームワークと共通言語を提供します

SNMPのコンポーネント

SNMPフレームワークには、次のコンポーネントがあります。これについては、次のセクションで説明します。

- [SNMPマネージャ](#)
- [SNMPエージェント](#)
- [SNMP MIB](#)

SNMPマネージャ

SNMPマネージャは、SNMPを使用してネットワークホストのアクティビティを制御および監視するシステムです。最も一般的な管理システムは、ネットワーク管理システム(NMS)です。NMSという用語は、ネットワーク管理に使用される専用デバイスに適用することも、そのようなデバイスで使用されるアプリケーションに適用することもできます。

SNMPエージェント

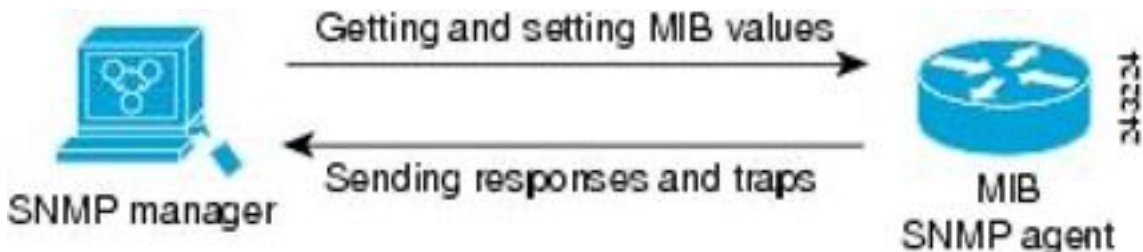
SNMPエージェントは、管理対象デバイス内のソフトウェアコンポーネントで、デバイスのデータを維持し、必要に応じてこのデータをシステム管理にレポートします。エージェントは、ルーティングデバイス(ルータ、アクセスサーバ、またはスイッチ)上に存在します。

SNMP MIB

SNMPエージェントにはMIB変数が含まれており、この変数の値は、「Get」または「Set」操作によってSNMPマネージャによって要求または変更できます。マネージャは、エージェントから

値を取得するか、そのエージェントに値を保存できます。エージェントは、デバイスパラメータとネットワークデータに関する情報を収集するためのリポジトリであるSNMP MIBからデータを収集します。エージェントは、データの get または set がマネージャから要求された際に応答することもできます。

次の図は、SNMPマネージャとエージェント間の通信を示しています。マネージャは、SNMP MIB値を取得および設定するエージェント要求を送信します。エージェントはこれらの要求に応答します。このインタラクションに関係なく、エージェントはマネージャに任意の通知 (トラップまたはインフォーム) を送信して、ネットワークの状態をマネージャに通知できます。



SNMPの動作

SNMPアプリケーションは、次の操作を実行して、データの取得、SNMPオブジェクト変数の変更、および通知の送信を行います。

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

SNMP Get

SNMP GET操作は、SNMPオブジェクト変数を取得するためにNMSによって実行されます。GET操作には3つのタイプがあります。

- GET:SNMPエージェントから正確なオブジェクトインスタンスを取得します。
- GETNEXT – 次のオブジェクト変数を取得します。この変数は、指定した変数の辞書編集上の後継です。
- GETBULK:GETNEXT操作を繰り返すことなく、大量のオブジェクト変数データを取得します。

SNMP SET

SNMP SET操作は、オブジェクト変数の値を変更するためにNMSによって実行されます。

SNMP通知

SNMPの重要な機能は、SNMPエージェントから非送信請求の通知を生成する機能です。

任意 (非同期) 通知は、トラップまたはインフォーム要求 (インフォーム) として生成できます。トラップは、Simple Network Management Protocol(SNMP)マネージャにネットワーク上の状態を通知するメッセージです。インフォームは、SNMPマネージャからの受信確認の要求を含むトラップです。通知は、不適切なユーザ認証、再起動、接続の終了、ネイバーデバイスへの接続の切断、その他の重要なイベントを示す可能性があります。

トラップの受信時に受信側から確認応答が送信されないため、トラップの信頼性はインフォームよりも低くなります。送信側は、トラップが受信されたかどうかを認識しません。インフォームを受信したSNMPマネージャは、SNMP応答プロトコルデータユニット(PDU)でメッセージに確認応答します。送信者が応答を受信しない場合は、インフォームを再送信できます。このため、インフォームは目的の宛先に到達できる可能性が高くなります。

トラップは、デバイスやネットワークのリソースを大量に消費するため、信頼性が低い場合でも推奨されます。送信されるとすぐに廃棄されるトラップとは異なり、インフォームは応答が受信されるか、要求がタイムアウトするまでメモリに保持される必要があります。また、トラップは1回だけ送信されますが、インフォームは複数回再送信される場合があります。再試行によってトラフィックが増加し、ネットワークのオーバーヘッドが増加します。トラップとインフォームを使用するには、信頼性とリソースのトレードオフが必要です。

MIBとRFC

Management Information Base(MIB)モジュールは通常、国際標準化機関であるInternet Engineering Task Force(IETF)に提出されたRequest for Comments(RFC)文書で定義されています。RFCは、一般に推奨されるインターネット標準を確立することを目的として、インターネット社会およびインターネットコミュニティ全体で検討するために個人またはグループによって作成されます。RFCのステータスが表示される前に、推奨事項はインターネットドラフト(I-D)ドキュメントとして公開されます。推奨される標準となったRFCは、標準ドキュメント(STD)とも呼ばれます。IETFの標準プロセスと活動については、インターネット学会のWebサイト(<http://www.isoc.org>)で学ぶことができます。シスコのドキュメントで参照されているすべてのRFC、I-D、およびSTDの全文は、IETFのWebサイト(<http://www.ietf.org>)で読むことができます。

シスコのSNMP実装では、RFC 1213で説明されているMIB II変数の定義と、RFC 1215で説明されているSNMPトラップの定義を使用します。

Ciscoでは、すべてのシステムに独自のMIB拡張機能を付加して提供しています。CiscoエンタープライズMIBは、ドキュメントに別段の記載がない限り、関連するRFCに記載されているガイドラインに準拠しています。MIBモジュール定義ファイルと、各シスコプラットフォームでサポートされているMIBのリストは、Cisco.comのCisco MIB Webサイトにあります。

SNMPのバージョン

現在、シスコデバイスは次のバージョンのSNMPをサポートしています。

- SNMPv1: Simple Network Management Protocol: RFC 1157で定義されている完全なインターネット標準 (RFC 1157は、RFC 1067およびRFC 1098として公開された以前のバージョンに代わるものです)。セキュリティはコミュニティストリングに基づいて実現されています。
- SNMPv2c: SNMPv2のコミュニティストリングベースの管理フレームワーク。SNMPv2c(「c」は「コミュニティ」用)は、RFC 1901、RFC 1905、およびRFC 1906で定義されている実験的なインターネットプロトコルです。SNMPv2cは、SNMPv2p(SNMPv2 Classic)のプロトコル動作とデータタイプの更新であり、SNMPv1のコミュニティベースのセキュリティモデルを使用します。
- SNMPv3: SNMPのバージョン3。SNMPv3は、RFC 3413 ~ 3415で定義されている、相互運用可能な標準ベースのプロトコルです。SNMPv3は、ネットワーク上のパケットを認証および暗号化することで、デバイスへのセキュアなアクセスを提供します。

SNMPv3で提供されるセキュリティ機能は次のとおりです。

- メッセージの整合性：転送中にパケットが改ざんされていないことを確認します。
- [認証(Authentication)]：メッセージが有効な送信元からのものであることを確認します。
- 暗号化：パケットの内容をスクランブルして、不正な送信元によってパケットが学習されないようにします。

SNMPv1 と SNMPv2c の両方では、コミュニティベースのセキュリティ方式が使用されています。SNMPマネージャのコミュニティは、エージェントMIBにアクセスできます。これは、コミュニティストリングによって定義されます。

SNMPv2cのサポートには、バルク取得メカニズムと、管理ステーションへの詳細なエラーメッセージの報告が含まれます。バルク検索メカニズムは、テーブルおよび大量の情報の検索をサポートし、必要なラウンドトリップの数を最小限に抑えます。SNMPv2cの改善されたエラー処理サポートには、さまざまなタイプのエラーを区別する拡張エラーコードが含まれています。これらの条件は、SNMPv1の単一のエラーコードによって報告されます。次の3種類の例外も報告されます。そのようなオブジェクトはなく、そのようなインスタンスはなく、MIBビューの終了です。

SNMPv3は、ユーザとユーザが存在するグループに対して認証戦略を設定するセキュリティモデルです。セキュリティレベルとは、セキュリティモデル内で許可されるセキュリティのレベルです。セキュリティモデルとセキュリティレベルの組み合わせによって、SNMPパケットを処理するときに使用されるセキュリティメカニズムが決まります。

次の3つのセキュリティモデルを使用できます。SNMPv1、SNMPv2c、およびSNMPv3。次の表に、セキュリティモデルとレベルの組み合わせとその意味を示します。

モデル	レベル	[Authentication]	暗号化	起こる
v1	noAuthNoPriv	コミュニティ文字列	No	コミュニティストリングの照合を使用して認証し
v2c	noAuthNoPriv	コミュニティ文字列	No	コミュニティストリングの照合を使用して認証し
v3	noAuthNoPriv	ユーザ名	No	認証にユーザ名の一致を使用する。
v3	authNoPriv	Message Digest 5(MD5)またはSecure Hash Algorithm(SHA)	No	HMAC-MD5 または HMAC-SHA アルゴリズムに基づいて認証を行う。
v3	authPriv	MD5またはSHA	データ暗号化規格(DES)	HMAC-MD5 または HMAC-SHA アルゴリズムに基づいて認証を行う。CBC-DES (DES-56) 標準に基づいて認証する以外に、DES 56ビット暗号化を行います。

管理ステーションでサポートされているSNMPのバージョンを使用するには、SNMPエージェントを実装する必要があります。エージェントは複数のマネージャと通信できます。

SNMPv3は、RFC 1901 ~ 1908、2104、2206、2213、2214、および2271 ~ 2275をサポートします。SNMPv3の詳細については、RFC 2570, Introduction to Version 3 of the Internet-standard Network Management Frameworkを参照してください (これは標準ドキュメントではありません)。

ヤン模型

Yangモデルは、システムの特定の機能またはハードウェア特性のツリー構造の抽象化を表します。ネットワーク要素では、Yangモデルはルーティングプロトコルの内部物理センサーアレイを表すことができます。YANGの言語と用語は[RFC 6020](#)で説明され、次に[RFC 7950](#)で更新されまし

た。Yangモデルでは、メイン構造を表すデータがサブモジュールとサブノードのリストに分類されます。次に、いくつかのノードタイプについて説明します。

リーフノードには、整数や文字列などの単純なデータが含まれます。特定のタイプの値が1つだけで、子ノードはありません。

```
leaf host-name {
    type string;
    description "Hostname for this system";
}
```

リーフリストは、リーフごとに特定のタイプの値を1つだけ持つリーフノードのシーケンスです。

```
leaf-list domain-search {
    type string;
    description "検索するドメイン名のリスト";
}
```

コンテナノードは、サブツリー内の関連ノードをグループ化するために使用されます。コンテナには子ノードのみがあり、値はありません。コンテナには、任意のタイプの任意の数の子ノード（リーフ、リスト、コンテナ、リーフリストを含む）を含めることができます。

```
コンテナシステム{
    コンテナログイン{
        リーフメッセージ{
            type string;
            説明
            「ログインセッションの開始時に表示されるメッセージ」;
        }
    }
}
```

リストは、リストのエントリのシーケンスを定義します。各エントリは構造やレコードインスタンスと似ており、キーリーフの値によって一意に識別されます。リストは複数のキーリーフを定義でき、任意のタイプの任意の数の子ノード（リーフ、リスト、コンテナなど）を含むことができます。

最後に、これらのノードタイプをすべてバインドするサンプルモデルを次の例に示します。

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

ただし、ヤンのモデルで使用されるヤン言語は、コンテナ/リスト/リーフへのデータの編成を示すものではありません。このため、ネットワーク要素の特定の機能を多様なYangモデルで表すことができます。この課題は、次のYang Modelsタイプで対処されています。

- [OpenConfigモデル](#)
- [ネイティブモデル](#)

OpenConfigモデル

OpenConfigモデルは、特定の機能を表すモデルに対して非依存ベンダー組織を使用して開発されました。このアプローチの利点は、NMSがこれらのモデルを使用して、マルチベンダーまたはマルチプラットフォーム環境のネットワーク要素と対話できることです。

名前が示すように、これらのモデルはオープンで、このリンクのgithubなどのリポジトリで検索するために公開されています。

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

例として、ボーダーゲートウェイプロトコル(BGP)のopenconfigモデル、Link Aggregation Control Protocol(LACP)のopenconfigモデル、およびISISの別のopenconfigモデルがあり、それぞれ異なる特定のモデルがあります。BGPの場合、BGPエラーのモデルや、BGPポリシーの別のモデルなどを見つけることができます。モデルは関連している可能性があり、一部のモデルは別のヤンパッケージを呼び出すことができます。たとえば、openconfig-bgp-neighbor.yangはopenconfig-bgp.yangに属します。

```
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;
```

要約すると、OpenConfigモデルは、IETFやRFCの標準機能など、すべてのプラットフォームに共通するプロトコルを対象としています。

ネイティブモデル

一方、ネイティブモデルは、特定のプラットフォームに固有の詳細な構造をカバーするベンダー指向モデルです。たとえば、電圧、温度、ASICカウンタ、ファブリックカウンタなど、ネットワーク要素内の物理値のセンサーをグループ化するモデルです。これらはプラットフォームに依存するため、NCS6K、ASR9K、またはCisco 8000に固有のモデルを見つけることが一般的です。

OpenConfig Modelsとして、ネイティブモデルはGithubリポジトリでも利用できます。

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

これらのモデルはOpenConfigモデルよりも具体的で完全である傾向があるため、特定のソフトウェアバージョンに関連付けられており、ソフトウェアリリース間で変更されることがあります。

ネイティブモデルには、次の2つの主なカテゴリがあります。

- 要素から情報を取得するために使用される「操作」モデル。

例：[Cisco-IOS-XR-eigrp-oper.yang](#)

- ネットワーク要素の設定に使用される「Cfg」モデル

[Cisco-IOS-XR-eigrp-cfg.yang](#)など

一般的に、Model Driven Telemetryは「oper」モデルを使用してインフラストラクチャからデータをストリーミングし、NSOなどのNMSは「cfg」モデルを使用してネットワーク要素の設定を変更します。

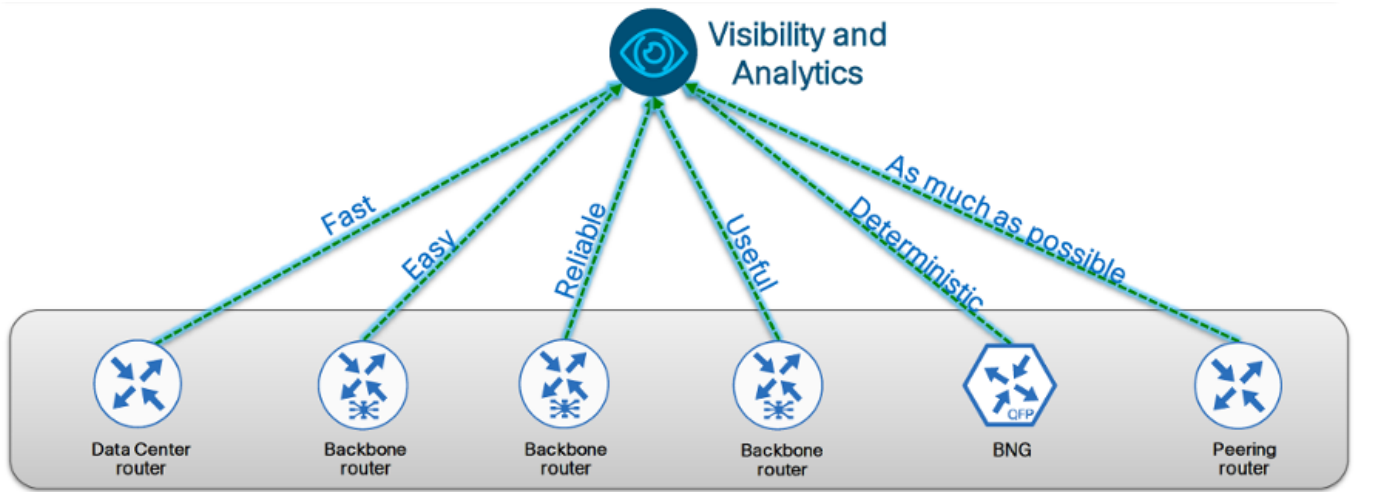
NativeおよびOpenConfig Yang Modelsは/pkg/yangフォルダのXRソフトウェアに存在し、プラットフォームでYang Modelが利用可能かどうかを確認するためにリストされます。次の例は、cXR 6.4.2を実行しているXRrv9k用です。

```
RP/0/RP0/CPU0:xrv9k1#run ls /pkg/yang | grep isis
9月22日 ( 火 ) 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#
```

テレメトリ

テレメトリは、異なるリモート要素から情報を一元的に収集し、可視性と分析レイヤを集約するプロセスです。

ネットワーク環境では、データはネットワーク内のすべての要素、ルータ、他の要素の間のスイッチによって生成され、情報は非常に大きな特定のプロトコル、パフォーマンスカウンタ、または物理センサーからのメジャーに関連する可能性があります。



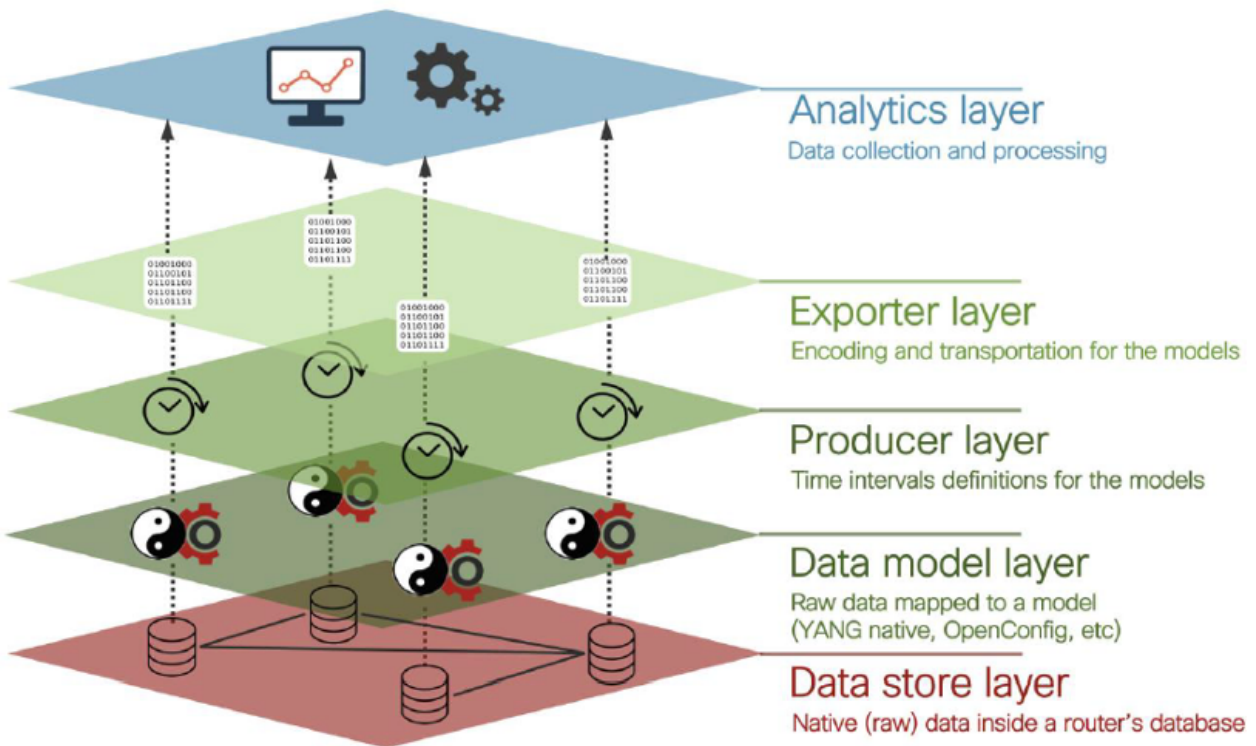
一般に、可視性と分析機能はネットワークの中心に位置し、テレメトリ情報のストリーミングはネットワークトランスポートメカニズムを使用するため、テレメトリ情報を可能な限り高速に拡張する必要があります。

テレメトリは、SNMPレガシーメカニズムとは異なり、プッシュパラダイムを使用します。プッシュパラダイムでは、定期的な間隔でポーリングされずにネットワーク自身のデータをストリーミングできます。これは、SNMPベースの監視の主な特性です。このプロビジョニングは、通常はサブスクリプションと呼ばれ、監視する変数のセット、データ収集のサンプリング間隔の通常の間隔、およびネットワークを介してこのデータを送信するリモートシステムに基づいています。

モデル駆動型テレメトリ

モデル駆動型テレメトリのMDT状態と、名前に示されているように、これはヤンモデルに基づいています。ネットワーク機器のすべての側面は、OSPFネイバーテーブル、RIBまたはモジュラシステムの各コンポーネントの温度センサーなど、ヤンのモデルで表すことができます。

MDTアーキテクチャについては、次のレイヤに分割できます。



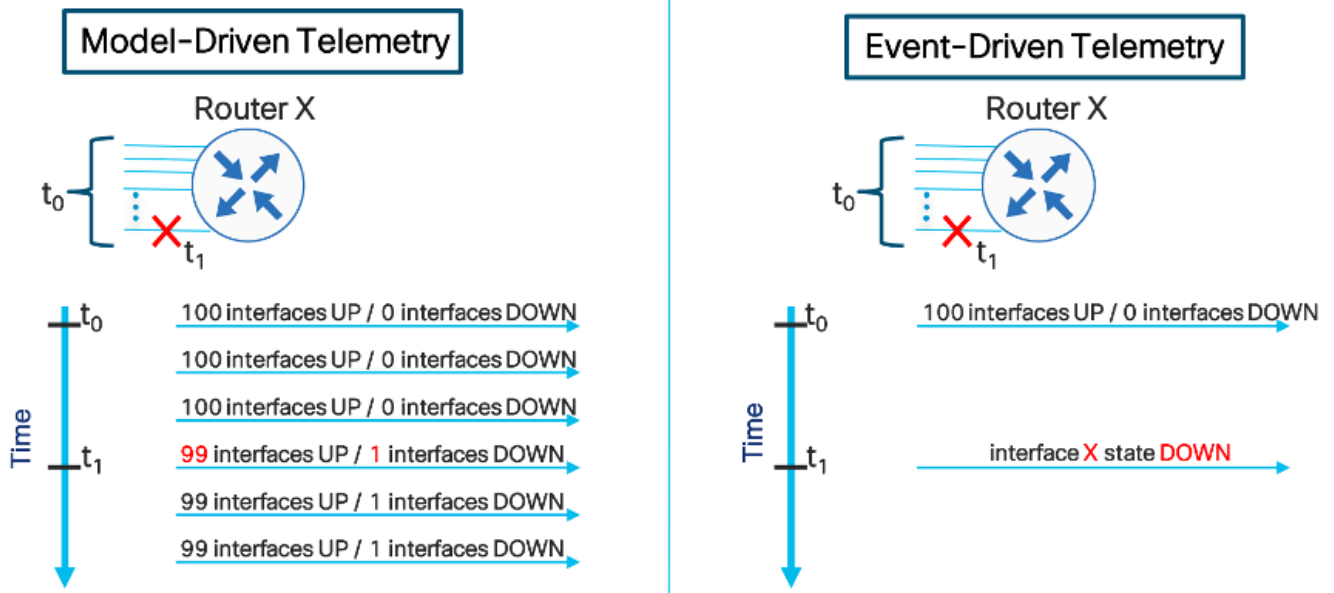
注：プロデューサー層に関しては、モデル駆動型テレメトリには、デバイスが内部データベースでrawデータを参照する頻度を制御するサンプリング間隔定義があり、このデータをデータモデル層に整理します。

テレメトリサブスクリプションでは、どのモデルとテナントパスを使用してデータを生成し、分析レイヤにストリーミングするかを定義します。この定義は、ビジネス目的に関連する情報に影響を与えます。このセンサーパスのMDT定義は、SNMPを介して取得するOIDを定義するアナログです。これは、両方のテクノロジーが定義されたサンプリングレートで構造化データを生成するためです。

イベント駆動型テレメトリ

EDTはEvent Driven Telemetryを意味し、構造のYangモデルにも基づいています。主な違いは、収集とデータストリームのトリガーが定期的な間隔ではなく、しきい値のクロス、リンクイベント、ハードウェア障害などの特定のイベントであることです。

次に、モデル駆動型テレメトリとイベント駆動型テレメトリのイベントからの比較を示します。

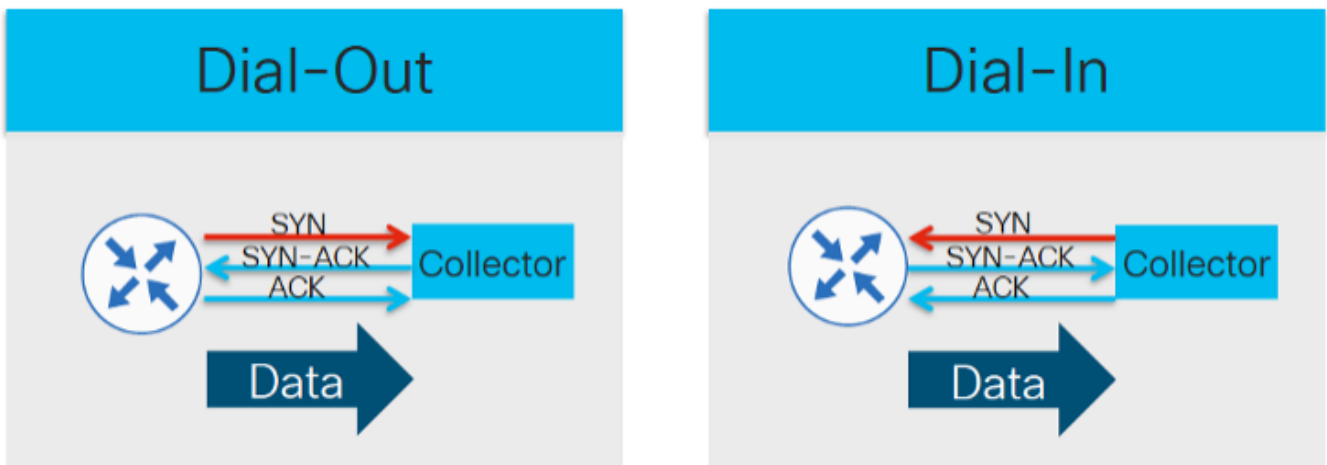


ヒント：この図は、MDTを使用した冗長メッセージを示していますが、EDTを使用した変更を表すメッセージのみを示しています。

トランスポート

テレメトリは可能な限り信頼性が高く、インフラストラクチャと分析層の間でセッション指向のソケットを使用するためにTransmission Control Protocol(TCP)ベースのトランスポートを使用することは理にかなっています。

テレメトリを使用する場合は2つの主要なアプローチがあり、3ウェイハンドシェイクの初期フローでは互いに異なります。

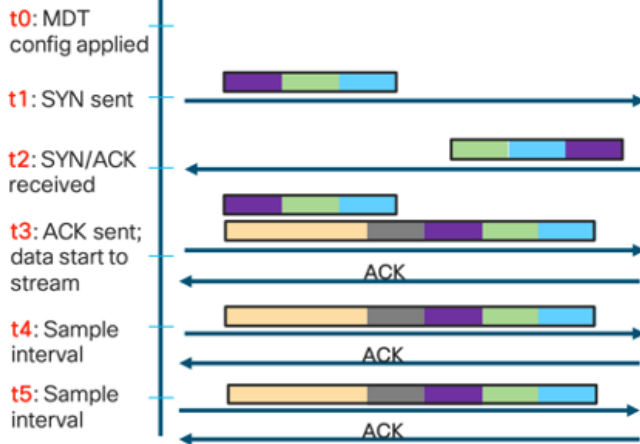


注：ダイヤルアウトモードでは、セッションのセットアップがインフラストラクチャ側で開始されます。これは、対象のセンサーをネットワーク要素に設定する必要があることを意味します。対照的に、ダイヤルインアプローチでは、コレクタがセットアップ段階で特定のセンサーパスを要求する必要があるため、ネットワーク要素の設定が簡単になります。

TCP

TCPは、ネットワーク要素とテレメトリコレクタの間にコネクション型セッションを確立する最も簡単な方法であり、信頼性の目的でACKをルータに送信したルータからコレクタへのデータストリームは次のように開始します。

Router



Collector

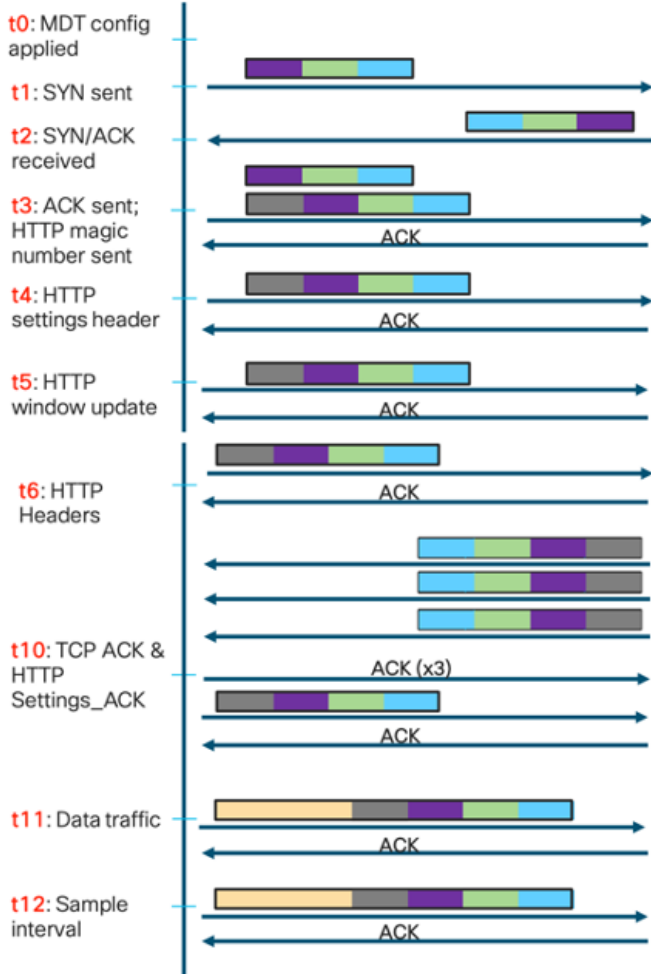


- Ethernet header (14B)
- IP header (20B)
- TCP header (28B)
- TCP MDT (12B)
- MDT data (N bytes)
- * Additional headers might exist (.1q, MPLS, etc)

gRPC

Google Protocol RPC (gRPC)はハイパーテキスト転送プロトコル/2 (HTTP/2)上で動作するため、セッション自体がセットアップ時に形成され、コレクタ側からネイティブに速度制御できます。

Router



Collector



- t7: HTTP Settings
- t8: HTTP Window_update
- t9: HTTP Settings_ACK

- Ethernet header (14B)
- IP header (20B)
- TCP header (32-40 bytes)
- HTTP/2 header (M bytes)
- MDT data (N bytes)
- * Additional headers might exist (.1q, MPLS, etc)

gNMI/gNOI

gRPCネットワーク管理インターフェイス(gNMI)は、Googleによって開発されたgRPCネットワーク管理プロトコルです。gNMIは、ネットワークデバイスの設定をインストール、操作、および削除するメカニズムを提供し、また動作データを表示するメカニズムも提供します。gNMIを通じて提供されるコンテンツは、YANGを使用してモデリングできます。

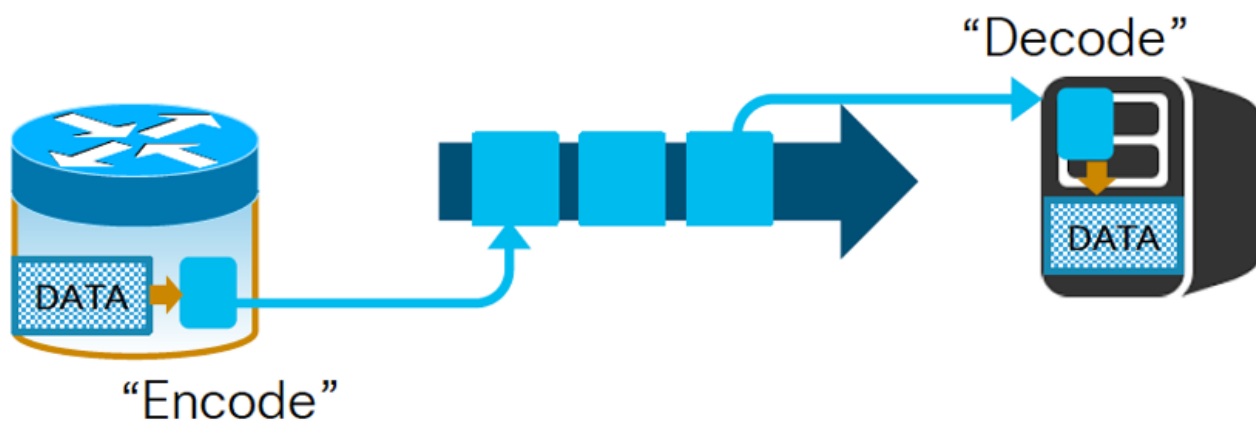
gNMIはgRPC-HTTP/2を使用して接続を確立し、ネットワーク要素とテレメトリコレクタとしても使用できるNMS間に双方向チャンネルを提供します。また、デバイスを管理するためのインターフェイスも提供します。

このプロトコルでサポートされている操作の間に、要求された情報、成功、またはエラーメッセージを返すgNMI Get、gNMI Setが見つかります。

gRPC Network Operations Interface(gNOI)は、gNMIと同じ通信チャンネルを使用するマイクロサービスの集まりです。ただし、ping、リポート、SSL証明書の変更、クリアなど、設定自体に関連しない一般的な操作が可能です。

符号化

Yangモデルは、データの構造、その階層、およびデータ上のすべてのリーフノードのタイプを定義します。ただし、モデリングでは、このデータのシリアル化の方法は示されません。このプロセスは、構造化データからTCP接続 (raw TCP、gRPC、gNMIなど) を介して送信されるバイトのストリームへの変換を制御します。



注：このプロセスは、データをエンコードするネットワーク要素の同等のメカニズムを使用して実装する必要があり、コレクタはこのデータをデコードする必要があります。

JSON

最初のエンコーディングメカニズムは、ネイティブのJavaScript Object Notation (JSON)形式です。これは周知の形式ですが、メッセージのサイズに関して非効率的な文字列として表されるすべてのキーが人間の指向です。JSONを使用する主な利点は、解析が容易で、次の例のようにテキストベースで読み取ることです。

```
{ "node_id_str": "test-IOSXR ", "subscription_id_str": " if_rate", "encoding_path": "Cisco-IOS-XR-  
infra-statsdoper:infra-statistics/interfaces/interface/latest/datarate", "collection_id": 49,  
"collection_start_time": 1510716302467, "msg_timestamp": 1510716302479, "data_json": [ {
```

```
"timestamp":1510716282334, "keys":{"interface-name":"Null0" }, "content":{"input-data-rate":0,
"input-packet-rate":0, "output-data-rate":0, "output-packet-rate":0, <> { "timestamp":
1510716282344, "keys":{"interface-name":"GigabitEthernet0/0/0/0" }, "content":{"input-data-
rate":8, "input-packet-rate":1, "output-data-rate":2, "output-packet-rate":0, <>
"collection_end_time":1510716302372 }
```

GPB-KV

Google Protocol Buffers-Key Value (GPB-KV)エンコーディング形式は、プロトコルバッファを使用してYangモデルの特定の要素をポイントするメッセージを使用するため、自己記述型GPBとも呼ばれます。これは、エンコード/デコードの目的に必要な.protoファイルが1つのみであり、データのキー自体が自己記述文字列であることを意味します。

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

GPB

最後に、コンパクトGPBとも呼ばれるGoogle Protocol Buffers (GPB)は、このアプローチをさらに一歩進め、.protoファイルを使って構造のすべてのキーをマッピングする必要があります。しかし、欠点は、インフラストラクチャ/コレクタがサポートするすべてのYangモデルに関連するすべての.protoファイルをコンパイルする必要があるということです。

```
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
```

IOS XRでのMDTの設定

モデル駆動型テレメトリデータのストリーミングで使用されるコアコンポーネントは次のとおりです。

- セッション
- センサーパス
- サブスクリプション

- トランスポートとエンコード

セッションオプションは、前述したように、ダイヤルインまたはダイヤルアウトです。IOS XRで設定を作成します。

ダイヤルアウトモード

ダイヤルアウトモードの場合、ルータはサブスクリプションに基づいて宛先へのセッションを開始します。プロセスには次の手順が含まれている必要があります。

- 通知先グループの作成
- センサーグループの作成
- サブスクリプションの作成
- ダイヤルアウト設定の検証

宛先グループを作成するには、コレクタのインターネットプロトコルバージョン4(IPv4)/インターネットプロトコルバージョン6(IPv6)アドレスと、このアプリケーションをサービスするポートを知っている必要があります。また、ネットワークデバイスとコレクタで合意すべきプロトコルと符号化を指定する必要があります。

最後に、コレクタネットワークアドレスとの通信に使用するVirtual Routing and Forwarding(VRF)を指定する必要があります。

次に、ダイヤルアウトの設定例を示します。

```
テレメトリモデル駆動型
destination-group DG1
vrf MGMT
address-family ipv4 192.168.122.20 port 5432
encoding self-describing-gpb
protocol tcp
!
```

次に、エンコーディングオプションを示します。

```
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#encoding ?
gpb GPB符号化
json JSONエンコード
自己記述型gpb自己記述型GPBエンコード←別名GPB-KV
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#encoding
```

プロトコルオプション :

```
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#protocol ?
grpc gRPC
TCP TCP
udp UDP
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#protocol grpc ?
gzip gRPC gzipメッセージ圧縮
no-tls TLS
tls-hostname TLS hostname
<cr>
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#protocol tcp ?
<cr>
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#protocol udp ?
packetize UDPパケットサイズ
<cr>
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#protocol udp
```

TCPプロトコルは単純で、IPv4/IPv6アドレスに接続されたポート設定だけが必要です。一方、User Datagram Protocol(UDP)はコネクションレス型であるため、宛先グループのステータスは常にアクティブになります。

gRPCでの圧縮は、オプションのgzipキーワードを使用して行うことができません。gRPCはデフォルトでTLSを使用するため、証明書をルータにローカルにインストールして使用する必要があります。この動作は、**no-tls**キーワードの設定によって**上書き**できます。最後に、**tls-hostname**キーワードを使用して、証明書の目的で別のホスト名を指定することができます。

次に、sensor-groupセクションを追加して、対象のsensor-pathをリストします。このセクションは簡単ですが、センサーパス自体でフィルタリングを行うことで、中央処理装置(CPU)や帯域幅

などの複数のリソースを最適化できることを知っておくことが重要です。

```
テレメトリモデル駆動型
sensor-group SG1
sensor-path Cisco-IO-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
sensor-path Cisco-IO-XR-infra-statsd-oper:infra-statistics/interfaces/interface[interface-name='Mgmt*']/data-rate
!
```

注：センサーパスに必要な形式は<model-name>:<container-path>です

このドキュメントでは、SNMPベースのモニタリングから、この従来のアプローチではリーフを表すOIDを使用したYANGモデルへのマッピングを示します。これは、同じ「リーフ」に一致するXPathで表されます。

最後の設定段階では、サブスクリプションを設定する必要があります。サブスクリプションは、センサーグループとテレメトリストリーミング用のキャデンスを宛先グループに結び付けます。

```
テレメトリモデル駆動型
サブスクリプションSU1
sensor-group-id SG1 sample-interval 5000
destination-id DG1
!
```

この例では、前のコレクションの最後を基準にした5000ミリ秒（5秒）のサンプリング間隔を使用します。この動作を変更するには、*strict-timer*オプションを使用して*sample-interval*キーワードを変更します。

確認のために、サブスクリプションのステータスをカバーする次のコマンドを使用できます。この方法では、sensor-groupおよびdestination-group情報もカバーできます。

```
RP/0/RP0/CPU0:C8000-1#sh telemetry model-driven subscription SU1
Wed Nov 18 15:38:01.397 UTC
サブスクリプション： SU1
```

```
State          : アクティブ
センサーグループ：
id:SG1
サンプル間隔： 5000 ms
ハートビート間隔： 適用外
センサーパス： Cisco-IO-XR-infra-statsd-oper:infra-statistics/interfaces/interface[interface-name='Mgmt*']/data-rate
センサーパスの状態： Resolved
センサーパス： Cisco-IO-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
センサーパスの状態： Resolved
通知先グループ：
グループID： DG1
宛先IP： 192.168.122.10
宛先ポート： 5432
宛先Vrf： MGMT(0x60000001)
エンコード： 自己記述型gpb
トランスポート： tcp
State          : アクティブ
TLS            : False
送信された総バイト数： 636284346
送信パケットの総数： 4189
最終送信時刻： 2020-11-18 15:37:58.1700077650 +0000
コレクショングループ：
```

```
id:9 ミリ秒
サンプル間隔： 5000 ms
ハートビート間隔： 適用外
ハートビートは常に： False
エンコード： 自己記述型gpb
コレクション数： 1407
収集時間： 最小値： 最大4ミリ秒： 13 ms
合計時間： 最小値： 平均8 ms： 最大10ミリ秒： 20 ms
遅延合計： 0
送信エラーの合計： 0
送信ドロップの合計： 0
その他のエラーの合計： 0
```



```
データインスタンスなし : 1407
Last Collection Start:2020-11-18 15:37:57.1699545994 +0000
最後のコレクション終了 : 2020-11-18 15:37:57.1699555589 +0000
センサーパス : Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/data-rate
id:10
サンプル間隔 : 5000 ms
ハートビート間隔 : 適用外
ハートビートは常に : False
エンコード : 自己記述型gpb
コレクション数 : 1391
収集時間 : 最小値 : 最大178 ms: 473 ms
合計時間 : 最小値 : 平均247 ms: 最大283 ms: 559 ms
遅延合計 : 0
送信エラーの合計 : 0
送信ドロップの合計 : 0
その他のエラーの合計 : 0
データインスタンスなし : 0
Last Collection Start:2020-11-18 15:37:58.1699805906 +0000
最後のコレクション終了 : 2020-11-18 15:37:58.1700078415 +0000
センサーパス : Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
RP/0/RP0/CPU0:C8000-1#
```

ダイヤルインモード

ダイヤルインモードでは、コレクタがネットワーク要素への接続を開始します。次に、コレクタはサブスクリプションの構築に関心があることを示す必要があります。

設定には次の手順があります。

- gRPCサービスの有効化
- センサーグループの設定
- 確認

gRPCサービスを有効にするには、次に設定が表示されます。

```
!  
grpc  
vrf MGMT  
port 57400  
no-tls  
アドレスファミリデュアル  
!
```

オプションは、VRFやTCPポートなど、簡単です。デフォルトでは、gRPCはTLSを使用しますが、**no-tls**キーワードを使用して無効にできます。最後に、**address-family dual**オプションを使用すると、IPv4とIPv6を使用した接続が可能です。

次に、ダイヤルインではセンサーグループの定義がローカルで必要になります。これは、後でコレクタがサブスクリプションを定義するために使用します。

```
テレメトリモデル駆動型  
sensor-group SG3  
sensor-path Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization  
sensor-path Cisco-IOS-XR-fib-common-oper:fib-statistics/nodes/node/drops  
!  
!
```

これで、ダイヤルインモードの設定が完了し、コレクタ自身がgRPCを使用してルータにサブスクリプションを作成できます。確認のために、ダイヤルアウトモードと同じアプローチを行うことができます。

```
RP/0/RP0/CPU0:C8000-1#sh telemetry model-driven subscription anx-1605878175837  
11月20 13:58:37.894 UTC(GMT)  
サブスクリプション : ANX-1605878175837  
—  
State : アクティブ  
センサーグループ :  
id:SG3  
サンプル間隔 : 15000 ms
```

```
ハートビート間隔： 適用外
センサーパス： Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
センサーパスの状態： Resolved
センサーパス： Cisco-IOS-XR-fib-common-oper:fib-statistics/nodes/node/drops
センサーパスの状態： Resolved
通知先グループ：
グループ ID： DialIn_1003
宛先 IP： 192.168.122.10
宛先ポート： 46974
圧縮： gzip
エンコード： json
トランスポート： ダイアルイン
State： アクティブ
TLS： False
送信された総バイト数： 71000035
送信パケットの総数： 509
最終送信時刻： 2020-11-20 13:58:32.1030932699 +0000
コレクショングループ：
—
```

```
id:5
サンプル間隔： 15000 ms
ハートビート間隔： 適用外
ハートビートは常に： False
エンコード： json
コレクション数： 170
収集時間： 最小値： 最大273 ms: 640 ms
合計時間： 最小値： 平均276 ms: 最大390ミリ秒： 643 ms
遅延合計： 0
送信エラーの合計： 0
送信ドロップの合計： 0
その他のエラーの合計： 0
データインスタンスなし： 0
Last Collection Start:2020-11-20 13:58:32.1030283276 +0000
最後のコレクション終了： 2020-11-20 13:58:32.1030910008 +0000
センサーパス： Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
id:6
サンプル間隔： 15000 ms
ハートビート間隔： 適用外
ハートビートは常に： False
エンコード： json
コレクション数： 169
収集時間： 最小値： 最大15ミリ秒： 33 ms
合計時間： 最小値： 平均17 ms: 最大22ミリ秒： 33 ms
遅延合計： 0
送信エラーの合計： 0
送信ドロップの合計： 0
その他のエラーの合計： 0
データインスタンスなし： 0
Last Collection Start:2020-11-20 13:58:32.1030910330 +0000
最後のコレクション終了： 2020-11-20 13:58:32.1030932787 +0000
センサーパス： Cisco-IOS-XR-fib-common-oper:fib-statistics/nodes/node/drops
RP/0/RP0/CPU0:C8000-1#
```

ヒント：ダイアルインモードのルータでは、cadence、encoding、collector IPまたはtransportがハードコードされていないことに注意してください。

MDTへのSNMPの移行

従来のSNMPからテレメトリモデルへの移行を実現するには、次の点を取り上げる必要があります。

- XPATHへのMIBの移行
- テレメトリへのトラップ移行
- セキュリティに関する考慮事項

XPATHへのMIBの移行

このため、MIBを特定の機能に（少なくとも上位レベルで）マッピングできる独自の階層を使用して分類できます。

BGP4-MIB

次の表は、OID名と番号、およびBGPピアリングセッションに関連するモデル駆動型テレメトリセンサーグループで設定される対応するXPathを示しています。

OID名	OID番号	OIDの説明	XPATH
bgpPeerLastError	1.3.6.1.2.1.15.3.1.14	この接続でこのピアによって最後に表示されたエラーコードとサブコード。エラーが発生していない場合、このフィールドは0です。それ以外の場合、この2バイトのOCTET STRINGの最初のバイトにはエラーコードが含まれ、2番目のバイトにはサブコードが含まれます。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/neighbor-missing-error-table/neighbor/last-notify-error-code
bgpPeerOutUpdates	1.3.6.1.2.1.15.3.1.11	この接続で送信されたBGP UPDATEメッセージの数。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/update-messages-out
bgpPeerInUpdates	1.3.6.1.2.1.15.3.1.10	この接続で受信したBGP UPDATEメッセージの数。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/update-messages-in
bgpPeerNegotiatedVersion	1.3.6.1.2.1.15.3.1.4	2つのピア間で実行されるBGPのネゴシエートされたバージョン。 bgpPeerStateがopenconfirmまたはestablished状態でない限り、このエントリはゼロ(0)である必要があります。このオブジェクトの有効な値は0 ~ 255です。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/negotiated-protocol-version
bgpPeerState	1.3.6.1.2.1.15.3.1.2	BGPピアの接続状態。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connection-state
bgpPeerRemoteAddr	1.3.6.1.2.1.15.3.1.7	このエントリのBGPピアのリモートIPアドレス。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connection-remote-address
bgpPeerLocalAddr	1.3.6.1.2.1.15.3.1.5	このエントリのBGP接続のローカルIPアドレス。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connection-local-address

			stance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connection-local-address
bgpPeerFsmEstablishedTime	1.3.6.1.2.1.15.3.1.16	このタイマーは、このピアが確立状態であった時間（秒）または確立状態で最後にピアになった後の期間を示します。これは、新しいピアが設定されたとき、またはルータがブートされたときに、ゼロに設定されます。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connection-established-time
bgpPeerAdminStatus	1.3.6.1.2.1.15.3.1.3	BGP接続の望ましい状態。「stop」から「start」への移行により、BGP手動開始イベントが生成されます。「start」から「stop」への移行により、BGP手動停止イベントが生成されます。このパラメータは、BGPピア接続を再起動するために使用できません。適切な認証を行わずに、このオブジェクトへの書き込みアクセスを提供する際には注意が必要です。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connection-admin-status

CISCO-BGP4-MIB

次の表は、OIDの名前と番号、およびBGPセッションの状態とプレフィックスが変更されたモデル駆動型テレメトリセンサーグループで設定される対応するXPathを示しています。

OID名	OID番号	OIDの説明	XPATH
cbgpPeer2RemoteAs	1.3.6.1.4.1.9.9.187.1.2.5.1.11	BGP OPENメッセージで受信したリモート自律システム番号。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/sessions/session/remote-as
cbgpPeer2PrevState	1.3.6.1.4.1.9.9.187.1.2.5.1.29	BGPピア接続の以前の状態。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/previous-connection-state
cbgpPeer2State	1.3.6.1.4.1.9.9.187.1.2.5.1.30	BGPピアの接続状態。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connection-state
cbgpPeer2LocalAddr	1.3.6.1.4.1.9.9.187.1.2.5.1.6	このエントリのBGP接続のローカル	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connection-local-address

			IPアドレス。	instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connectio local-address
cbgpPeer2AdvertisedPrefixes	1.3.6.1.4.1.9.9.187.1.2.8.1.6		このカウンタは、アドレスファミリに属するルートプレフィックスがこの接続でアドバタイズされると増加します。接続がハードリセットされると、この値は0に初期化されます。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/af-data/prefixes-advertiseo
cbgpPeer2AcceptedPrefixes	1.3.6.1.4.1.9.9.187.1.2.8.1.1		アドレスファミリに属するこの接続で受け入れられたルートプレフィックスの数。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/af-data/prefixes-acted
cbgpPeerPrefixLimit	1.3.6.1.4.1.9.9.187.1.2.1.1.3		この接続で受け入れられるルートプレフィックスの最大数	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/af-data/n prefix-limit
cbgpPeer2PrefixThreshold	1.3.6.1.4.1.9.9.187.1.2.8.1.4		この接続のアドレスファミリのプレフィックスしきい値(%)で、プレフィックス数がしきい値を超えたことを示す警告メッセージまたは対応するSNMP通知が生成されます。	Cisco-IOS-XR-ipv4-bgp-oper:bgp/config-instances/config-instance/default-vrf/entity-configurations/entity-configuration/af-depend config/max-prefix-warn- threshold

CISCO-CLASS-BASED-QOS-MIB

次の表は、Quality of Service(QoS)クラス/ポリシーの統計情報に関連するモデル駆動型テレメトリセンサーグループに設定されるOID名と番号、および対応するXPathを示しています。

OID名	OID番号	OIDの説明	XPATH
cbQosCMDropBitRate	1.3.6.1.4.1.9.9.166.1.15.1.1.18	ドロップを生成できるすべての機能(たとえば、ポリシング、ランダム検出など)の結果として生じる、クラスごとのドロップのビットレート。	Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/input/service-policy-names/service-po instance/statistics/class- stats/general-stats/total- rate Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/output/se policy-names/service-po instance/statistics/class-

cbQosCMDropPkt64	1.3.6.1.4.1.9.9.166.1.15.1.1.14	ドロップを発生させるすべての機能（たとえば、警察、ランダム検出など）の結果として、クラスごとにドロップされたパケットの64ビットカウンタが表示されます。	stats/general-stats/total-rate Cisco-IOS-XR-qos-map- oper:qos/interface- table/interface/input/serv- policy-names/service-po- instance/statistics/class- stats/general-stats/total- packets Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/output/se- policy-names/service-po- instance/statistics/class- stats/general-stats/total- packets Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/input/serv- policy-names/service-po- instance/statistics/class- stats/general-stats/pre-p- matched-packets Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/output/se- policy-names/service-po- instance/statistics/class- stats/general-stats/pre-p- matched-packets Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/input/serv- policy-names/service-po- instance/statistics/class- stats/class-name Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/input/serv- policy-names/service-po- instance/statistics/class- stats/child-policy/class- stats/general-stats/trans- bytes Cisco-IOS-XR-qos-ma- oper:qos/interface- table/interface/output/se- policy-names/service-po- instance/statistics/class- stats/child-policy/class- stats/general-stats/trans- bytes
cbQosCMPrePolicyPkt64	1.3.6.1.4.1.9.9.166.1.15.1.1.13	QoSポリシーを実行する前の着信パケットの64ビット数。	
cbQosCMName	1.3.6.1.4.1.9.9.166.1.7.1.1.1	クラスマップの名前。	
cbQosCMPostPolicyByte64	1.3.6.1.4.1.9.9.166.1.15.1.1.10	QoSポリシー実行後の発信オクテットの64ビット数。	

cbQosIfIndex	1.3.6.1.4.1.9.9.166.1.1.1.1.4	このサービスが接続されているインターフェイスのifIndex。このフィールドは、論理インターフェイスにsnmp ifIndexがある場合にのみ意味を持ちます。たとえば、cbQosIfTypeがcontrolPlaneの場合、このフィールドの値は意味がありません。	Cisco-IOS-XR-infra-policymgr-oper:policy-manager/global/policy-map/policy-map-types/policy-map-type/policy-maps
cbQosConfigIndex	1.3.6.1.4.1.9.9.166.1.5.1.1.2	各Objectの任意の（システム割り当て）構成（インスタンス独立）インデックス。同じ構成を持つ各オブジェクトは、同じ構成インデックスを共有します。	Cisco-IOS-XR-infra-policymgr-oper:policy-manager/global/policy-map/policy-map-types/policy-map-type/policy-maps
cbQosCMPrePolicyByte64	1.3.6.1.4.1.9.9.166.1.15.1.1.6	QoSポリシーを実行する前の着信オクテットの64ビット数。	Cisco-IOS-XR-qos-manager:qos/interface-table/interface/input/service-policy-names/service-policy-instance/statistics/class-stats/child-policy/class-stats/general-stats/pre-policy-matched-bytes
			Cisco-IOS-XR-qos-manager:qos/interface-table/interface/output/service-policy-names/service-policy-instance/statistics/class-stats/child-policy/class-stats/general-stats/pre-policy-matched-bytes

CISCO-ENHANCED-MEMPOOL-MIB

次の表は、メモリ使用量に関連するモデル駆動型テレメトリセンサーグループに設定されるOID名と番号、および対応するXPathを示しています。

OID名	OID番号	OIDの説明	XPATH
cempMemPoolUsed	1.3.6.1.4.1.9.9.221.1.1.1.1.7	物理エンティティ上のアプリケーションによって現在使用されているメモリプールからのバイト数を示します。	Cisco-IOS-XR-not-misc-oper:memory-summary/nodes/node/sur
cempMemPoolHCUsed	1.3.6.1.4.1.9.9.221.1.1.1.1.18	物理エンティティ上のアプリケーションによって現在使用されてい	Cisco-IOS-XR-not-misc-oper:memory-summary/nodes/node/det

cempMemPoolHCFree 1.3.6.1.4.1.9.9.221.1.1.1.20

るメモリプールからの
バイト数を示します。
このオブジェクトは、 al-used
cempMemPoolUsedの
64ビット版です。
物理エンティティで現
在使用されていないメ
モリプールのバイト数
を示します。このオブ
ジェクトは、
cempMemPoolFreeの
64ビット版です。

Cisco-IOS-XR-not-misc-
oper:memory-
summary/nodes/node/det
e-physical-memory

CISCO-ENTITY-FRU-CONTROL-MIB

次の表は、監視対象システムの現場交換可能ユニットに関連するモデル駆動型テレメトリセンサーグループに設定されるOID名と番号、および対応するXPATHを示しています。

OID名	OID番号	OIDの説明	XPATH
cefcFRUPowerOperStatus	1.3.6.1.4.1.9.9.117.1.1.2.1.2	動作可能FRUの電力状態。	Cisco-IOS-XR-invmgr- oper:inventory/entities/ attributes/fru-info/powe operational-state
cefcFRUPowerAdminStatus	1.3.6.1.4.1.9.9.117.1.1.2.1.1	管理目的のFRU電力状態。	Cisco-IOS-XR-invmgr- oper:inventory/entities/ attributes/fru-info/powe administrative-state
cefcModuleStatusLastChangeTime	1.3.6.1.4.1.9.9.117.1.2.1.1.4	cefcModuleOperStatusが変更されたときのsysUpTimeの値。 このオブジェクトは、モジュールが最後に再初期化されてからのアップタイムを提供します。このオブジェクトは永続的ではありません。モジュールがリセット、再起動、電源オフになると、アップタイムはゼロから始まります。	Cisco-IOS-XR-invmgr- oper:inventory/entities/ attributes/fru-info/last- operational-state-chang
cefcModuleUpTime	1.3.6.1.4.1.9.9.117.1.2.1.1.8	このオブジェクトは永続的ではありません。モジュールがリセット、再起動、電源オフになると、アップタイムはゼロから始まります。	Cisco-IOS-XR-invmgr- oper:inventory/entities/ attributes/fru-info/card- time
cefcModuleResetReason	1.3.6.1.4.1.9.9.117.1.2.1.1.3	このオブジェクトは、モジュールで最後にリセットされた理由を示します。	Cisco-IOS-XR-invmgr- oper:inventory/entities/ attributes/fru-info/card- reason
cefcModuleOperStatus	1.3.6.1.4.1.9.9.117.1.2.1.1.2	このオブジェクトは、モジュールの動作状態を示します。	Cisco-IOS-XR-invmgr- oper:inventory/entities/ attributes/fru-info/card- operational-state
cefcModuleAdminStatus	1.3.6.1.4.1.9.9.117.1.2.1.1.1	このオブジェクトは、モジュールの管理	Cisco-IOS-XR-invmgr- oper:inventory/entities/

CISCO-ENTITY-SENSOR-MIB

次の表は、ノード上のセンサーエンティティに関連するモデル駆動型テレメトリセンサーグループに設定されるOID名と番号、および対応するXPathを表しています。

OID名	OID番号	OIDの説明	XPATH
entSensorValue	1.3.6.1.4.1.9.9.91.1.1.1.1.4	この変数は、センサーによって表示される最新の測定値を報告します。この変数の値を正しく表示または解釈するには、entSensorType、entSensorScale、およびentSensorPrecisionも理解している必要があります。ただし、entSensorValueとentSensorThresholdTableで指定されたしきい値を、意味的な知識がなくても比較できます。 この変数は、しきい値の最新の評価の結果を示します。しきい値条件がtrueの場合、	Cisco-IOS-XR-invmgoper:inventory/entity/attributes/env-seninfo/value
entSensorThresholdEvaluation	1.3.6.1.4.1.9.9.91.1.2.1.1.5	entSensorThresholdEvaluationはtrue(1)です。しきい値条件がfalseの場合、entSensorThresholdEvaluationはfalse(2)です。しきい値は、entSensorValueUpdateRateで示されるレートで評価されます。	Cisco-IOS-XR-invmgoper:inventory/entity/attributes/thresho

CISCO-FLASH-MIB

次の表は、システムのフラッシュストレージに関連するモデル駆動型テレメトリセンサーグループに設定されるOID名と番号、および対応するXPathを示しています。

OID名	OID番号	OIDの説明	XPATH
ciscoFlashPartitionName	1.3.6.1.4.1.9.9.10.1.1.4.1.1.10	システムによってパーティションを参照するために使用されるフラッシュパーティション名。AAAAAAAnn形式の任意の英数字の文字列を指定できます。ここで、Aはオプションの英数字を表し、nは数字を表します。数字は、常に文字列の末尾の	Cisco-IOS-XR-shellutil-filesystoper:file-system/node/file-system/type

部分を形成する必要があります。システムはアルファベット文字を削除し、数値部分を使用してパーティションインデックスにマッピングします。フラッシュ操作は、この名前に基づいてデバイスパーティションに転送されます。システムにはデフォルトパーティションの概念があります。これはデバイスの最初のパーティションになります。パーティション名が指定されていない場合、システムは操作をデフォルトパーティションに転送します。したがって、デフォルトパーティションで操作が行われている場合、またはデバイスに1つのパーティションしか存在しない場合(パーティション化されていない)を除き、パーティション名は必須です。

フラッシュパーティションサイズ。

ciscoFlashDeviceMinPartitionSizeの整数倍である必要があります。1つのパーティションがある場合、このサイズはciscoFlashDeviceSizeと同じになります。このオブジェクトは、64ビット版のciscoFlashPartitionSizeです。フラッシュパーティション内の空き領域。フラッシュ内のファイルの実際のサイズには、ファイルシステムのファイルヘッダーを表す小さなオーバーヘッドが含まれていることに注意してください。特定のファイルシステムには、空き領域を計算する際に考慮するパーティションまたはデバイスヘッダーのオーバーヘッドが存在する場合があります。空き領域は、既存のすべてのファイル(有効/無効/削除されたファイルと各ファイルのファイルヘッダーを含む)の合計パーティション

ciscoFlashPartitionSizeExtended 1.3.6.1.4.1.9.9.10.1.1.4.1.1.13

Cisco-IOS-XR-shellutil-filessystem/node/filessystem/size

ciscoFlashPartitionFreeSpaceExtended 1.3.6.1.4.1.9.9.10.1.1.4.1.1.14

Cisco-IOS-XR-shellutil-filessystem/node/filessystem/free

ンサイズより小さいサイズ、パーティションヘッダーより小さいサイズ、コピーされる次のファイルのヘッダーサイズより小さいサイズとして計算されます。つまり、このオブジェクトは、コピー可能な最大ファイルのサイズを与えます。管理エンティティは、ファイルやパーティション・ヘッダーの長さなどのオーバーヘッドを認識したり使用したりすることは想定されていません。このようなオーバーヘッドは、ファイル・システムによって異なる場合があります。フラッシュ内の削除されたファイルに空き領域がありません。ファイルが占める領域を再利用するには、パーティションを消去する必要がある場合があります。このオブジェクトは、64ビット版の ciscoFlashPartitionFreeSpace です

CISCO-PROCESS-MIB

次の表は、モデル駆動型テレメトリセンサーグループで設定されるOID名と番号、および対応するXPATHを表しています。関連するCPU使用率とプロセスのリソース割り当てです。

OID名	OID番号	OIDの説明	XPATH
cpmCPUTotal1minRev	1.3.6.1.4.1.9.9.109.1.1.1.1.7	直近の 1 分間の CPU 全体のビジー パーセンテージ .これは、廃止されたオブジェクト cpmCPUTotal1min に代わる新しいオブジェクトで、値の範囲が 0 ~ 100 に拡大されています。	Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization/total-cpu-minute
cpmCPUTotal5minRev	1.3.6.1.4.1.9.9.109.1.1.1.1.8	直近の 5 分間の CPU 全体のビジー パーセンテージ .これは、廃止されたオブジェクト cpmCPUTotal5min に代わる新しいオブジェクトで、値の範囲が 0 ~ 100 に拡大されています。	Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization/total-cpu-minute
cpmCPUTotal15minRev	1.3.6.1.4.1.9.9.109.1.1.1.1.31	直近の 15 分間の CPU 全体のビジー パーセンテージ ジ.このオブジェクトは	Cisco-IOS-XR-wdsysmon-fd-oper:system-

		cpmCPUTotal15minオブジェクトを無効にし、値の範囲を(0..100)を増やします	monitoring/cpu-utilization/total-cpu-minute
cpmProcessName	1.3.6.1.4.1.9.9.109.1.2.1.1.2	このプロセスに関連付けられている名前。名前が32文字を超える場合は、最初の31文字に切り捨てられ、最後の文字として「*」が追加され、これが切り捨てられたプロセス名であることを示します。	Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization/process-cpu/process-name
cpmProcessTextSegmentSize	1.3.6.1.4.1.9.9.109.1.2.3.1.15	これは、プロセスとその共有オブジェクトのテキストメモリを示します。	Cisco-IOS-XR-oper:processes-memory/nodes/node-ids/process-id-seg-size
cpmProcessDynamicMemorySize	1.3.6.1.4.1.9.9.109.1.2.3.1.18	これは、プロセスで使用されているダイナミックメモリの量を示します。	Cisco-IOS-XR-oper:processes-memory/nodes/node-ids/process-id-limit
cpmProcessDataSegmentSize	1.3.6.1.4.1.9.9.109.1.2.3.1.16	これは、プロセスとその共有オブジェクトのデータセグメントを示します。	Cisco-IOS-XR-oper:processes-memory/nodes/node-ids/process-id-seg-size
cpmProcExtMemAllocatedRev	1.3.6.1.4.1.9.9.109.1.2.3.1.1	このプロセスがシステムから受信したすべての動的に割り当てられたメモリの合計。これには、返された可能性のあるメモリが含まれます。解放されたメモリの合計は、cpmProcExtMemFreedRevによって提供されます。このオブジェクトはcpmProcExtMemAllocatedを無効にします。	Cisco-IOS-XR-oper:processes-memory/nodes/node-ids/process-id
cpmProcExtMemFreedRev	1.3.6.1.4.1.9.9.109.1.2.3.1.2	このプロセスがシステムに返したすべてのメモリの合計。このオブジェクトはcpmProcExtMemFreedを無効にします。	Cisco-IOS-XR-oper:processes-memory/nodes/node-ids/process-id

ENTITY-MIB

次の表は、システム上のモデル駆動型テレメトリセンサーグループ関連の物理エンティティで設定されるOID名と番号、および対応するXPATHを表しています。

OID名	OID番号	OIDの説明	XPATH
entPhysicalName	1.3.6.1.2.1.47.1.1.1.1.7	物理エンティティのテキスト名。このオブジェクトの値は	Cisco-IOS-XR-snmp-entitymib-oper:entity-

、ローカルデバイスによって割り当てられたコンポーネントの名前である必要があり、デバイスの「コンソール」に入力するコマンドで使用するのに適しています。デバイスの物理コンポーネントの命名構文に応じて、「console」などのテキスト名や、単純なコンポーネント番号（ポートやモジュール番号など）「1」などのテキスト名を使用できます。ローカル名が存在しない場合、またはこのオブジェクトが該当しない場合、このオブジェクトには長さ0の文字列が含まれます。2つの物理エンティティの entPhysicalName の値は、コンソールインターフェイスがスロット1とスロット1のカードを区別しない場合でも同じであることを注意してください。

physical-index

論理エンティティのテキストの説明。このオブジェクトには、論理エンティティの製造元の名前を識別する文字列を含める必要があり、論理エンティティの各バージョンに対して個別の値を設定する必要があります。

Cisco-IOS-XR-snmp-agent-oper:snmp/information-name/

物理エンティティのテキストの説明。このオブジェクトには、物理エンティティの製造元の名前を示す文字列を含める必要があり、物理エンティティのバージョンまたはモデルごとに異なる値を設定する必要があります。

Cisco-IOS-XR-snmp-agent-oper:snmp/Cisco-IOS-XR-snmp-entity-mib/entity-physical-indexes/

この物理エンティティを'含む'物理エンティティの

entPhysicalIndex の値。値0は、この物理エンティティが他の物理エンティティに含まれていないことを示します。「封じ込め」関係のセットは厳密な階層を定義することに注意してください。つまり、再帰は許可されません。物理エンティティが複数の物理エンティティ（倍幅モジュールなど）に含まれる場合、このオ

Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/inv-basic-bag/unique-id

entLogicalDescr

1.3.6.1.2.1.47.1.2.1.1.2

entPhysicalDescr

1.3.6.1.2.1.47.1.1.1.1.2

entPhysicalContainedIn

1.3.6.1.2.1.47.1.1.1.1.4

プロジェクトは

entPhysicalIndexの最小値を持つ含むエンティティを識別する必要があります。

物理エンティティの一般的なハードウェアタイプを示します。このオブジェクトは、物理エンティティの一般クラスを最も正確に示す標準の列挙値に設定する必要があります。複数ある場合は主クラスを指定します。この物理エンティティに対して適切な標準登録識別子が存在しない場合、値「other(1)」が返されます。

このエージェントによって値が不明な場合、値「unknown(2)」が返されます。

物理エンティティのベンダー固有のハードウェアリビジョン文字列。推奨値は、コンポーネント自体に実際に印刷されるハードウェアリビジョンIDです（存在する場合）。リビジョン情報が内部的に印刷不可能（バイナリなど）な形式で保存されている場合、エージェントは実装固有の方法で、このような情報を印刷可能な形式に変換する必要があります。特定のハードウェアリビジョン文字列が物理コンポーネントに関連付けられていない場合、またはこの情報がエージェントに不明な場合、このオブジェクトには長さ0の文字列が含まれます。

物理エンティティのベンダー固有のファームウェアリビジョン文字列。リビジョン情報が内部的に印刷不可能（バイナリなど）な形式で保存されている場合、エージェントは実装固有の方法で、このような情報を印刷可能な形式に変換する必要があります。特定のファームウェアプログラムが物理コンポーネントに関連付けられていない場合、またはこの情報がエージェントに不明な場合、このオブジェクトには長さ0の文字列が含まれます。

entPhysicalClass

1.3.6.1.2.1.47.1.1.1.1.5

Cisco-IOS-XR-invmgr
oper:inventory/entities

entPhysicalHardwareRev

1.3.6.1.2.1.47.1.1.1.1.8

Cisco-IOS-XR-invmgr
oper:inventory/entities
y/attributes/inv-basic-
bag/hardware-revision

entPhysicalFirmwareRev

1.3.6.1.2.1.47.1.1.1.1.9

Cisco-IOS-XR-invmgr
oper:inventory/entities
y/attributes/inv-basic-
bag/firmware-revision

entPhysicalSoftwareRev	1.3.6.1.2.1.47.1.1.1.1.10	<p>物理エンティティのベンダー固有のソフトウェアリビジョン文字列。リビジョン情報が内部的に印刷不可能（バイナリなど）な形式で保存されている場合、エージェントは実装固有の方法で、このような情報を印刷可能な形式に変換する必要があります。特定のソフトウェアプログラムが物理コンポーネントに関連付けられていない場合、またはこの情報がエージェントに不明な場合、このオブジェクトには長さ0の文字列が含まれます。</p>	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/inv-basic-bag/software-revision
entPhysicalSerialNum	1.3.6.1.2.1.47.1.1.1.1.11	<p>物理エンティティのベンダー固有のシリアル番号文字列。推奨値は、コンポーネント自体に実際に印刷されたシリアル番号の文字列（存在する場合）です。物理エンティティの最初のインスタンス化では、エージェントがこの情報を使用できる場合、そのエンティティに関連付けられたentPhysicalSerialNumの値は、ベンダーが割り当てた正しいシリアル番号に設定されます。シリアル番号が不明または存在しない場合、entPhysicalSerialNumは長さ0の文字列に設定されます。</p> <p>インストールされているすべての物理エンティティのシリアル番号を正しく識別できる実装では、entPhysicalSerialNumオブジェクトへの書き込みアクセスを提供する必要はありません。entPhysicalSerialNum文字列に不揮発性ストレージを提供できないエージェントは、このオブジェクトに対する書き込みアクセスを実装する必要はありません。すべての物理コンポーネントがシリアル番号を持つわけではなく、シリアル番号を必要とするのでもありません。</p> <p>entPhysicalIsFRUオブジェクトの関連付けられた値が「false(2)」(リピータモジュール</p>	Cisco-IOS-XR-invmgr-oper:inventory/entities/attributes/inv-basic-bag/serial-number

ル内のリピータポートなど)である物理エンティティには、固有のシリアル番号は必要ありません。エージェントは、このようなエンティティに対して書き込みアクセスを提供する必要はなく、長さが0の文字列を返す場合があります。

entPhysicalSerialNumのインスタンスに対して書き込みアクセスが実装され、値がインスタンスに書き込まれる場合、エージェントは、そのエンティティがインスタンス化されている限り、同じ物理エンティティに関連付けられたentPhysicalSerialNumインスタンス内のものに値を保持します。これには、物理エンティティのentPhysicalIndex値が変更される原因を含め、ネットワーク管理システムのすべての再初期化/リブートにおけるインスタンス化が含まれます。

この物理コンポーネントの製造元の名前。推奨値は、実際にコンポーネント自体に印刷されたメーカー名の文字列(存在する場合)です。

entPhysicalModelName、entPhysicalFirmwareRev、entPhysicalSoftwareRev、およびentPhysicalSerialNumオブジェクトのインスタンス間の比較は、entPhysicalMfgNameの同じ値を持つentPhysicalEntriesでのみ意味があることに注意してください。物理コンポーネントに関連付けられている製造元の名前文字列がエージェントに認識されない場合、このオブジェクトには長さ0の文字列が含まれます。

この物理コンポーネントに関連付けられたベンダー固有のモデル名識別子の文字列。推奨値は、お客様が表示できる部品番号です。この部品番号は、コンポーネント自体に印刷できます。物理コンポーネ

entPhysicalMfgName 1.3.6.1.2.1.47.1.1.1.1.12

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

entPhysicalModelName 1.3.6.1.2.1.47.1.1.1.1.13

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

ントに関連付けられたモデル名文字列がエージェントに不明な場合、このオブジェクトには長さ0の文字列が含まれます。

IF-MIB

次の表は、インターフェイスの特性とカウンタに関連するモデル駆動型テレメトリセンサーグループで設定されるOID名と番号、および対応するXPATHを示しています。

OID名	OID番号	OIDの説明	XPATH
ifMtu	1.3.6.1.2.1.2.2.1.4	インターフェイスで送受信できる最大パケットのサイズ（オクテット単位）。ネットワークデータグラムの送信に使用されるインターフェイスの場合、これはインターフェイスで送信できる最大のネットワークデータグラムのサイズです。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/mtu
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	プロトコルサブレイヤのインターフェイスのアドレス。たとえば、802.xインターフェイスの場合、このオブジェクトには通常、MACアドレスが含まれます。インターフェイスのメディア固有のMIBは、ビットとバイトの順序およびこのオブジェクトの値の形式を定義する必要があります。そのようなアドレス（シリアル回線など）を持たないインターフェイスの場合、このオブジェクトには長さがゼロのオクテット文字列が含まれている必要があります。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface-ty information/bundle- information/member/m address
ifType	1.3.6.1.2.1.2.2.1.3	インターフェイスのタイプ。ifTypeの追加の値は、IANAifTypeテキスト表記法の構文を更新することによって、インターネット割り当て番号局(IANA)によって割り当てられます。	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface-ty
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	上位レベルのプロトコルが送信を要求し、このサブレイヤでマルチキャストまたはブロードキャストアドレスにアドレス指定されなかった（廃棄または送信されなかったパケットを含む）パケットの合計数。このカウンタの値の不連続性は、管理システムの再初	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/packets-sent

ifHCOUcastPkts	1.3.6.1.2.1.31.1.1.1.11	<p>期化時、および ifCounterDiscontinuityTimeの 値で示される他の時に発生す ることがあります。 上位レベルのプロトコルが送 信を要求し、このサブレイヤ でマルチキャストまたはブロード キャストアドレスにアド レス指定されなかった (廃棄 または送信されなかったパケ ットを含む) パケットの合計 数。このオブジェクトは、 ifOutUcastPktsの64ビット版で す。このカウンタの値の不連 続性は、管理システムの再初 期化時、および</p>	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/packets-sent
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	<p>ifCounterDiscontinuityTimeの 値で示される他の時に発生す ることがあります。 このサブレイヤによって上位 (サブ) レイヤに配信され、 このサブレイヤのマルチキャ ストアドレスまたはブロード キャストアドレスにアドレス 指定されなかったパケットの 数。このカウンタの値の不連 続性は、管理システムの再初 期化時、および</p>	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/packets-received
ifHCInUcastPkts	1.3.6.1.2.1.31.1.1.1.7	<p>ifCounterDiscontinuityTimeの 値で示される他の時に発生す ることがあります。 このサブレイヤによって上位 (サブ) レイヤに配信され、 このサブレイヤのマルチキャ ストアドレスまたはブロード キャストアドレスにアドレス 指定されなかったパケットの 数。このオブジェクトは、 ifInUcastPktsの64ビット版で す。このカウンタの値の不連 続性は、管理システムの再初 期化時、および</p>	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/packets-received
ifOutErrors	1.3.6.1.2.1.2.2.1.20	<p>ifCounterDiscontinuityTimeの 値で示される他の時に発生す ることがあります。 パケット指向インターフェイ スの場合、エラーが原因で送 信できなかった発信パケット の数。文字指向インターフェ イスまたは固定長インターフ ェイスの場合、エラーが原因 で送信できなかった発信送信 ユニットの数。このカウンタ</p>	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	<p>の値の不連続性は、管理システムの再初期化時、および ifCounterDiscontinuityTime の値で示される他の時に発生することがあります。</p> <p>送信を妨げるエラーが検出されなかったにもかかわらず、廃棄が選択された発信パケットの数。このようなパケットが廃棄される理由の 1 つとして、バッファ領域の解放が考えられます。このカウンタの値の不連続性は、管理システムの再初期化時、および ifCounterDiscontinuityTime の値で示される他の時に発生することがあります。</p>	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/output-drops
ifOutMulticastPkts	1.3.6.1.2.1.31.1.1.1.4	<p>上位レベルのプロトコルが送信を要求し、このサブレイヤのマルチキャストアドレスにアドレス指定されたパケットの総数 (廃棄または送信されなかったパケットを含む)。</p> <p>MAC 層プロトコルの場合は、グループアドレスと機能アドレスの両方が含まれます。このカウンタの値の不連続性は、管理システムの再初期化時、および</p>	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/multicast-packets
ifHCOutMulticastPkts	1.3.6.1.2.1.31.1.1.1.12	<p>ifCounterDiscontinuityTime の値で示される他の時に発生することがあります。</p> <p>上位レベルのプロトコルが送信を要求し、このサブレイヤのマルチキャストアドレスにアドレス指定されたパケットの総数 (廃棄または送信されなかったパケットを含む)。</p> <p>MAC 層プロトコルの場合は、グループアドレスと機能アドレスの両方が含まれます。このオブジェクトは、ifOutMulticastPkts の 64 ビット版です。このカウンタの値の不連続性は、管理システムの再初期化時、および</p>	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface- stats/multicast-packets
ifInMulticastPkts	1.3.6.1.2.1.31.1.1.1.2	<p>ifCounterDiscontinuityTime の値で示される他の時に発生することがあります。</p> <p>このサブレイヤから上位 (サブ) レイヤに配信され、このサブレイヤのマルチキャストアドレスにアドレス指定され</p>	Cisco-IOS-XR-pfi-im-c oper:interfaces/interfac xr/interface/interface- statistics/full-interface-

ifHCInMulticastPkts	1.3.6.1.2.1.31.1.1.1.8	<p>たパケットの数。MAC層プロトコルの場合は、グループアドレスと機能アドレスの両方が含まれます。このカウンタの値の不連続性は、管理システムの再初期化時、およびifCounterDiscontinuityTimeの値で示される他の時に発生することがあります。</p> <p>このサブレイヤから上位(サブ)レイヤに配信され、このサブレイヤのマルチキャストアドレスにアドレス指定されたパケットの数。MAC層プロトコルの場合は、グループアドレスと機能アドレスの両方が含まれます。このオブジェクトは、ifInMulticastPktsの64ビット版です。このカウンタの値の不連続性は、管理システムの再初期化時、およびifCounterDiscontinuityTimeの値で示される他の時に発生することがあります。</p>	stats/multicast-packets-received	Cisco-IOS-XR-pfi-im-oper:interfaces/interface-xr/interface/interface-statistics/full-interface-stats/multicast-packets-received
ifInErrors	1.3.6.1.2.1.2.2.1.14	<p>パケット指向のインターフェイスの場合、エラーが含まれ、上位層プロトコルへの配信を妨げた着信パケットの数。文字指向インターフェイスまたは固定長インターフェイスの場合、上位層プロトコルへの配信を妨げるエラーを含んだ着信転送ユニットの数。このカウンタの値の不連続性は、管理システムの再初期化時、およびifCounterDiscontinuityTimeの値で示される他の時に発生することがあります。</p>	Cisco-IOS-XR-pfi-im-oper:interfaces/interface-xr/interface/interface-statistics/full-interface-stats/input-errors	Cisco-IOS-XR-pfi-im-oper:interfaces/interface-xr/interface/interface-statistics/full-interface-stats/input-drops
ifInDiscards	1.3.6.1.2.1.2.2.1.13	<p>上位レイヤプロトコルへの配信を妨げるエラーが検出されなかったにもかかわらず、廃棄が選択された着信パケットの数。このようなパケットが廃棄される理由の1つとして、バッファ領域の解放が考えられます。このカウンタの値の不連続性は、管理システムの再初期化時、およびifCounterDiscontinuityTimeの値で示される他の時に発生することがあります。</p>	Cisco-IOS-XR-pfi-im-oper:interfaces/interface-xr/interface/interface-statistics/full-interface-stats/input-drops	Cisco-IOS-XR-pfi-im-c
ifOutOctets	1.3.6.1.2.1.2.2.1.16	<p>フレーミング文字を含む、イ</p>	Cisco-IOS-XR-pfi-im-c	

		<p>ンターフェイスから送信されたオクテットの合計数。このカウンタの値の不連続性は、管理システムの再初期化時、および</p> <p>ifCounterDiscontinuityTimeの値で示される他の時に発生することがあります。</p>	<p>oper:interfaces/interface-xr/interface/interface-statistics/full-interface-stats/bytes-sent</p>
ifHCOctets	1.3.6.1.2.1.31.1.1.1.10	<p>オブジェクトは、ifOutOctetsの64ビット版です。このカウンタの値の不連続性は、管理システムの再初期化時、および</p> <p>ifCounterDiscontinuityTimeの値で示される他の時に発生することがあります。</p>	<p>Cisco-IOS-XR-pfi-im-c oper:interfaces/interface-xr/interface/interface-statistics/full-interface-stats/bytes-sent</p>
ifInOctets	1.3.6.1.2.1.2.2.1.10	<p>インターフェイスで受信したオクテットの総数 (フレーミング文字を含む)。このカウンタの値の不連続性は、管理システムの再初期化時、および</p> <p>ifCounterDiscontinuityTimeの値で示される他の時に発生することがあります。</p>	<p>Cisco-IOS-XR-pfi-im-c oper:interfaces/interface-xr/interface/interface-statistics/full-interface-stats/bytes-received</p>
ifHCInOctets	1.3.6.1.2.1.31.1.1.1.6	<p>オブジェクトは、ifInOctetsの64ビット版です。このカウンタの値の不連続性は、管理システムの再初期化時、および</p> <p>ifCounterDiscontinuityTimeの値で示される他の時に発生することがあります。</p>	<p>Cisco-IOS-XR-pfi-im-c oper:interfaces/interface-xr/interface/interface-statistics/full-interface-stats/bytes-received</p>
ifOutBroadcastPkts	1.3.6.1.2.1.31.1.1.1.5	<p>上位レベルのプロトコルが送信を要求し、このサブレイヤのブロードキャストアドレスにアドレス指定されたパケット (廃棄されたか送信されなかったパケットを含む) の合計数。このカウンタの値の不連続性は、管理システムの再初期化時、および</p> <p>ifCounterDiscontinuityTimeの値で示される他の時に発生することがあります。</p>	<p>Cisco-IOS-XR-pfi-im-c oper:interfaces/interface-xr/interface/interface-statistics/full-interface-stats/broadcast-packets</p>
ifHCOutBroadcastPkts	1.3.6.1.2.1.31.1.1.1.13	<p>上位レベルのプロトコルが送信を要求し、このサブレイヤのブロードキャストアドレス</p>	<p>Cisco-IOS-XR-pfi-im-c oper:interfaces/interface-xr/interface/interface-</p>

		にアドレス指定されたパケット (廃棄されたか送信されなかったパケットを含む) の合計数。このオブジェクトは、ifOutBroadcastPktsの64ビット版です。このカウンタの値の不連続性は、管理システムの再初期化時、およびifCounterDiscontinuityTimeの値で示される他の時に発生することがあります。	statistics/full-interface-stats/broadcast-packets sent
ifInBroadcastPkts	1.3.6.1.2.1.31.1.1.1.3	このサブレイヤから上位 (サブ) レイヤに配信され、このサブレイヤのブロードキャストアドレスにアドレス指定されたパケットの数。このカウンタの値の不連続性は、管理システムの再初期化時、およびifCounterDiscontinuityTimeの値で示される他の時に発生することがあります。	Cisco-IOS-XR-pfi-im-oper:interfaces/interface-xr/interface/interface-statistics/full-interface-stats/broadcast-packets received
ifHCInBroadcastPkts	1.3.6.1.2.1.31.1.1.1.9	このサブレイヤから上位 (サブ) レイヤに配信され、このサブレイヤのブロードキャストアドレスにアドレス指定されたパケットの数。このオブジェクトは、ifInBroadcastPktsの64ビット版です。このカウンタの値の不連続性は、管理システムの再初期化時、およびifCounterDiscontinuityTimeの値で示される他の時に発生することがあります。	Cisco-IOS-XR-pfi-im-oper:interfaces/interface-xr/interface/interface-statistics/full-interface-stats/broadcast-packets received
ifIndex	1.3.6.1.2.1.2.2.1.1	各インターフェースの 0 以上の固有値。値は1から連続して割り当てることを推奨します。各インターフェイスサブレイヤの値は、少なくとも1つのエンティティのネットワーク管理システムの再初期化から次の再初期化まで、一定である必要があります。	Cisco-IOS-XR-pfi-im-oper:interfaces/interface-xr/interface/if-index
ifDescr	1.3.6.1.2.1.2.2.1.2	インターフェイスに関する情報を含むテキスト文字列。この文字列には、製造元の名前、製品名、インターフェイスハードウェア/ソフトウェアのバージョンが含まれている必要があります。	Cisco-IOS-XR-pfi-im-oper:interfaces/interface-xr/interface/description
ifSpeed	1.3.6.1.2.1.2.2.1.5	インターフェイスの現在の帯域幅 (ビット/秒) の推定値。	Cisco-IOS-XR-pfi-im-oper:interfaces/interface

帯域幅が変化しないインターフェイスや、正確な推定が行われないインターフェイスでは、このオブジェクトには公称帯域幅が含まれている必要があります。インターフェイスの帯域幅がこのオブジェクトでレポート可能な最大値より大きい場合、このオブジェクトは最大値

xr/interface/bandwidth

(4,294,967,295)をレポートし、ifHighSpeedを使用してインターフェイスの速度をレポートする必要があります。帯域幅の概念がないサブレイヤの場合、このオブジェクトはゼロである必要があります。

インターフェイスの現在の動作状態。testing(3)状態は、動作パケットが通過できないことを示します。

ifAdminStatusがdown(2)の場合、ifOperStatusはdown(2)になります。ifAdminStatusが

up(1)に変更された場合、インターフェイスがネットワークトラフィックを送受信できる状態であれば、ifOperStatusはup(1)に変更されます。インターフェイスが外部アクション

Cisco-IOS-XR-pfi-im-c
oper:interfaces/interfac
non-dynamics/interfac
non-dynamic/oper-stat

ifOperStatus

1.3.6.1.2.1.2.2.1.8

(たとえば、着信接続を待つシリアル回線)を待っている場合は、dormant(5)に変更されるはずですが、アップ(1)状態に移行するのを妨げる障害がある場合にのみ、ダウン(2)状態のままにしておく必要があります。インターフェイスに(通常はハードウェア)コンポーネントがない場合、notPresent(6)状態のままにしておく必要があります。

インターフェイスの望ましい状態。testing(3)状態は、動作パケットが通過できないことを示します。管理対象システムが初期化されると、すべてのインターフェイスが

Cisco-IOS-XR-pfi-im-c
oper:interfaces/interfac
non-dynamics/interfac
non-dynamic/admin-st

ifAdminStatus

1.3.6.1.2.1.2.2.1.7

down(2)状態のifAdminStatusで開始されます。明示的な管理アクションまたは管理対象システムによって保持された構成情報ごとの結果として、

ifAdminStatusがup(1)またはtesting(3)状態(またはdown(2)状態に変わります)に変わります。

インターフェイスのテキスト名。このオブジェクトの値は、ローカルデバイスによって割り当てられたインターフェイスの名前である必要があり、デバイスの「コンソール」に入力するコマンドで使用するのに適しています。これは、デバイスのインターフェイスの名前付け構文に応じて、「le0」などのテキスト名または「1」などの単純なポート番号です。ifTable内の複数のエントリがデバイスによって指定された1つのインターフェイスを表す場合、それぞれがifNameと同じ値を持ちます。他の(プロキシされた)デバイスのインターフェイスに関するSNMPクエリに回答するエージェントの場合、このようなインターフェイスのifNameの値は、プロキシされたデバイスのローカル名です。ローカル名が存在しない場合、またはこのオブジェクトが該当しない場合、このオブジェクトには長さ0の文字列が含まれます。

インターフェイスの現在の帯域幅の単位は1,000,000ビット/秒です。このオブジェクトが「n」の値を報告した場合、インターフェイスの速度は「n-500,000」~「n+499,999」の範囲内になります。帯域幅が変化しないインターフェイスや、正確な推定が行われないインターフェイスでは、このオブジェクトには公称帯域幅が含まれている必要があります。帯域幅の概念がないサブレイヤの場合、このオブジェクトはゼロである必要があります。

Cisco-IOS-XR-pfi-im-c
oper:interfaces/interfac
brief/interface-
brief/interface-name

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

ifName

1.3.6.1.2.1.31.1.1.1.1

ifHighSpeed

1.3.6.1.2.1.31.1.1.1.15

IP-MIB

次の表は、OID名と番号、および対応するXPathを、インターネットプロトコル(IP)の統計情報

と動作値に関連するモデル駆動型テレメトリセンサーグループで設定することを示しています。

OID名	OID番号	OIDの説明	XPATH
icmpInDestUnreachs	1.3.6.1.2.1.5.3	受信したICMP宛先到達不能メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpInParmProbs	1.3.6.1.2.1.5.5	受信したICMPパラメータ問題メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpInSrcQuenchs	1.3.6.1.2.1.5.6	受信したICMP Source Quenchメッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpInEchos	1.3.6.1.2.1.5.8	受信したICMPエコー (要求) メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpInEchoReps	1.3.6.1.2.1.5.9	受信したICMPエコー応答メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpInTimestamps	1.3.6.1.2.1.5.10	受信したICMPタイムスタンプ (要求) メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpInAddrMasks	1.3.6.1.2.1.5.12	受信したICMPアドレスマスク要求メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpInAddrMaskReps	1.3.6.1.2.1.5.13	受信したICMPアドレスマスク応答メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutMsgs	1.3.6.1.2.1.5.14	このエンティティが送信しようとしたICMPメッセージの合計数です。このカウンタには、icmpOutErrorsによってカウントされたすべてのカウンタが含まれることに注意してください。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutDestUnreachs	1.3.6.1.2.1.5.16	送信されたICMP宛先到達不能メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutTimeExcds	1.3.6.1.2.1.5.17	送信されたICMP Time Exceededメッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutParmProbs	1.3.6.1.2.1.5.18	送信されたICMPパラメータ問題メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st

icmpOutSrcQuenchs	1.3.6.1.2.1.5.19	送信されたICMP Source Quenchメッセージの数。	s/traffic/icmp-stats Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutRedirects	1.3.6.1.2.1.5.20	送信されたICMPリダイレクトメッセージの数。ホストはリダイレクトを送信しないため、ホストの場合、このオブジェクトは常に0になります。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutEchos	1.3.6.1.2.1.5.21	送信されたICMPエコー（要求）メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutEchoReps	1.3.6.1.2.1.5.22	送信されたICMPエコー応答メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutTimestamps	1.3.6.1.2.1.5.23	送信されたICMPタイムスタンプ（要求）メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutAddrMasks	1.3.6.1.2.1.5.25	送信されたICMPアドレスマスク要求メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
icmpOutAddrMaskReps	1.3.6.1.2.1.5.26	送信されたICMPアドレスマスク応答メッセージの数。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/st s/traffic/icmp-stats
ipAdEntIfIndex	1.3.6.1.2.1.4.20.1.2	このエントリが適用されるインターフェイスを一意に識別するインデックス値。このインデックスの特定の値で識別されるインターフェイスは、RFC 1573のifIndexの同じ値で識別されるインターフェイスと同じです。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/nodes/node/
ipAdEntAddr	1.3.6.1.2.1.4.20.1.1	このエントリのアドレス情報が関係するIPアドレス。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/interfaces/inte vrf/detail/primary-addre
ipAdEntNetMask	1.3.6.1.2.1.4.20.1.3	このエントリのIPアドレスに関連付けられているサブネットマスク。マスクの値は、すべてのネットワークビットが1に、すべてのホストビットが0に設定されたIPアドレスです。	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/interfaces/inte vrf/detail/prefix-length
ipAdEntBcastAddr	1.3.6.1.2.1.4.20.1.4	このエントリのIPアドレスに関連付けられた（論理）インターフェイスでデータグラムを送信するために使用される、IPブロードキャストアドレスの	Cisco-IOS-XR-ipv4-io- oper:ipv4- network/interfaces/inte vrf/detail/direct-broadca

最下位ビットの値。たとえば、インターネット標準オールワンプロードキャストアドレスを使用する場合、値は1になります。この値は、この（論理）インターフェイス上のエンティティによって使用されるサブネットとネットワークのブロードキャストアドレスの両方に適用されます。

ipNetToMediaPhysAddress 1.3.6.1.2.1.4.22.1.2

メディア依存の「物理」アドレス。

Cisco-IOS-XR-ipv4-arp-
oper:arp/nodes/node/e
/entry/hardware-address

IPMIB-COMMMON

次の表は、OIDの名前と番号、およびIP統計情報に関連するモデル駆動型テレメトリセンサーグループで設定される対応するXPathを示しています。

OID名	OID番号	OIDの説明	XPATH
ipLfStatsHCOutTransmits	1.3.6.1.2.1.4.31.3.1.31	このエンティティが送信のために下位層に提供したIPデータグラムの合計数。このオブジェクトは、 ipLfStatsOutTransmitsと同じデータグラムをカウントしますが、より大きな値を許可します。このカウンタの値の不連続性は、管理システムの再初期化時、および ipLfStatsDiscontinuityTimeの値で示される他の時に発生することがあります。	Cisco-IOS-XR-ipv4- oper:ipv4- network/nodes/node/s cs/traffic/ipv4-stats/pa forwarded
ipLfStatsInReceives	1.3.6.1.2.1.4.31.3.1.3	受信した入力IPデータグラムの総数（エラーで受信したデータグラムを含む）。このカウンタの値の不連続性は、管理システムの再初期化時、および ipLfStatsDiscontinuityTimeの値で示される他の時に発生することがあります。	Cisco-IOS-XR-ipv4- oper:ipv4- network/nodes/node/s cs/traffic/ipv4-stats/inp packets
ipLfStatsHCInReceives	1.3.6.1.2.1.4.31.3.1.4	受信した入力IPデータグラムの総数（エラーで受信したデータグラムを含む）。このオブジェクトは、 ipLfStatsInReceivesと同じデータグラムをカウントしますが、より大きな値を許可します。このカウンタの値の不連続性は、管理システムの再初期化時、および ipLfStatsDiscontinuityTimeの値	Cisco-IOS-XR-ipv4- oper:ipv4- network/nodes/node/s cs/traffic/ipv4-stats/inp packets

で示される他の時に発生することがあります。

LLDP-MIB

次の表は、モニタ対象ノードのLink Layer Discovery Protocol(LLDP)動作データに関連するモデル駆動型テレメトリセンサーグループで設定されるOID名と番号、および対応するXPathを示しています。

OID名	OID番号	OIDの説明	XPATH
lldpLocPortId	1.0.8802.1.1.2.1.3.7.1.3	ローカルシステムの特定のポートに関連付けられたポートコンポーネントを識別するために使用される文字列値。	Cisco-IOS-XR-etherne lldp- oper:lldp/nodes/node/ bors/devices/device/lldp- neighbor/port-id-detail Cisco-IOS-XR-etherne
lldpLocPortIdSubtype	1.0.8802.1.1.2.1.3.7.1.2	関連付けられた 'lldpLocPortId' オブジェクトで使用されるポートIDエンコードの種類です。	lldp- oper:lldp/nodes/node/ bors/devices/device/lldp- neighbor/mib/port-id-s type Cisco-IOS-XR-etherne
lldpLocChassisIdSubtype	1.0.8802.1.1.2.1.3.1	ローカルシステムに関連付けられたシャーシを識別するために使用されるエンコーディングのタイプ。	lldp- oper:lldp/nodes/node/ bors/devices/device/lldp- neighbor/mib/chassis- sub-type
lldpLocSysName	1.0.8802.1.1.2.1.3.3	ローカルシステムのシステム名を識別するために使用される文字列値。ローカルエージェントがIETF RFC 3418をサポートしている場合、lldpLocSysNameオブジェクトの値はsysNameオブジェクトと同じである必要があります。	Cisco-IOS-XR-etherne lldp- oper:lldp/nodes/node/ bors/devices/device/lldp- neighbor/detail/system name
lldpRemSysName	1.0.8802.1.1.2.1.4.1.1.9	リモートシステムのシステム名を識別するために使用する文字列値。	Cisco-IOS-XR-etherne lldp- oper:lldp/nodes/node/ bors/devices/device/lldp- neighbor/detail/system name
lldpRemChassisId	1.0.8802.1.1.2.1.4.1.1.5	リモートシステムに関連付けられているシャーシコンポーネントを識別するために使用する文字列値。	Cisco-IOS-XR-etherne lldp- oper:lldp/nodes/node/ bors/devices/device/lldp- neighbor/chassis-id
lldpRemChassisIdSubtype	1.0.8802.1.1.2.1.4.1.1.4	リモートシステムに関連付けられているシャーシを識別するために使用されるエンコーディングのタイプ。	Cisco-IOS-XR-etherne lldp- oper:lldp/nodes/node/ bors/devices/device/lldp- neighbor
lldpRemPortIdSubtype	1.0.8802.1.1.2.1.4.1.1.6	関連付けられた	Cisco-IOS-XR-etherne

		'lldpRemPortId'オブジェクトで使用されるポートIDエンコードの種類です。	lldp-oper:lldp/nodes/node/bors/devices/device/lldp-neighbor Cisco-IOS-XR-ethernet-lldp-oper:lldp/nodes/node/bors/devices/device/lldp-neighbor Cisco-IOS-XR-ethernet-lldp-oper:lldp/nodes/node/bors/details/lldp-neighbor/chassis-id
lldpRemPortId	1.0.8802.1.1.2.1.4.1.1.7	リモートシステムに関連付けられているポートコンポーネントを識別するために使用する文字列値。	
lldpLocChassisId	1.0.8802.1.1.2.1.3.2	ローカルシステムに関連付けられたシャーシコンポーネントを識別するために使用される文字列値。	

MPLS-TE-STD-MIB

次の表は、管理対象デバイスのマルチプロトコルラベルスイッチング(MPLS)トラフィックエンジニアリングの動作値に関連するモデル駆動型テレメトリセンサーグループで設定されるOID名と番号、および対応するXPathを示しています。

OID名	OID番号	OIDの説明	XPATH
mplsTunnelName	1.3.6.1.2.1.10.166.3.2.2.1.5	トンネルに割り当てられた標準名。この名前は、LSRのコンソールポートのトンネルを参照するために使用できます。mplsTunnelsIfがtrueに設定されている場合、このトンネルに対応するインターフェイスのifNameの値はmplsTunnelNameと同じになります。RFC 2863のifNameの説明も参照してください。	Cisco-IOS-XR-mpls-te-oper:mpls-te/p2p-p2p-tunnel/tunnel-heads/tunnel-head/tunnel-name
mplsTunnelDescr	1.3.6.1.2.1.10.166.3.2.2.1.6	トンネルに関する情報を含むテキスト文字列。説明がない場合、このオブジェクトには長さ0の文字列が含まれます。このオブジェクトはMPLSシグナリングプロトコルによってシグナリングされない可能性があります。したがって、トランジットおよび出力LSRでこのオブジェクトの値が自動的に生成または存在しない場合があります。	openconfig-network-instance:network-instances/network-instance/mpls/lsp/connected-path/tunnels/tunnel-description
mplsTunnelPerfHCpackets	1.3.6.1.2.1.10.166.3.2.9.1.2	トンネルによって転送されたパケット数の大容量カウンタ。	openconfig-network-instance:network-instances/network-instance/mpls/lsp/connected-path/tunnels/tunnel-statistics/counters/packets

mplsTunnelPerfHCBytes	1.3.6.1.2.1.10.166.3.2.9.1.5	トンネルによって転送されたバイト数の大容量カウンタ。	openconfig-network-instance/network-instances/network-instance/mpls/lsp/connected-path/tunnels/tunnel/s- counters/bytes
mplsTunnelHopIpAddr	1.3.6.1.2.1.10.166.3.2.4.1.5	このトンネルホップのトンネルホップアドレス。このアドレスのタイプは、対応するmplsTunnelHopAddrTypeの値によって決まります。対応するmplsTunnelHopRowStatusオブジェクトの値が「active」の場合、このオブジェクトの値は変更できません。	Cisco-IOS-XR-mpls- oper:mpls-te/p2p-p2r tunnel/tunnel- heads/tunnel- head/destination/des- n-address

RFC2465-MIB

次の表は、OIDの名前と番号、およびIPv6グローバル値に関連するモデル駆動型テレメトリセンサーグループで設定される対応するXPathを示しています。

OID名	OID番号	OIDの説明	XPATH
ipv6AddrPfxLength	1.3.6.1.2.1.55.1.8.1.2	このエントリのIPv6アドレスに関連付けられているプレフィックスの長さ（ビット単位）。	Cisco-IOS-XR-ipv6- oper:ipv6- network/nodes/node/int- - data/vrf/brief/brief/addre- efix-length
ipv6AddrAnycastFlag	1.3.6.1.2.1.55.1.8.1.4	このオブジェクトの値は 'true(1)' で、このアドレスがエニキャストアドレスの場合は 'false(2)' になります。	Cisco-IOS-XR-ipv6- oper:ipv6- network/nodes/node/int- - data/vrf/brief/brief/addre- anycast

SNMP-MIB

次の表は、OIDの名前と番号、およびSNMPエージェント自体に関連するモデル駆動型テレメトリセンサーグループで設定される対応するXPathを示しています（可能な場合）。

OID名	OID番号	OIDの説明	XPATH
sysUpTime	1.3.6.1.2.1.1.3	システムの稼働時間を表す文字列	Cisco-IOS-XR-snmp- oper:snmp/information/s- -up-time/ Cisco-IOS-XR-snmp- oper:snmp/information/s- -oid/ Cisco-IOS-XR-snmp- oper:snmp/information/s- -descr
sysObjectID	1.3.6.1.2.1.1.2.0	システムOIDを表す文字列	
sysDescr	1.3.6.1.2.1.1.1	システムの説明を表す文字列	

TCP-MIB

次の表は、OIDの名前と番号、およびTCP固有のカウンタに関連するモデル駆動型テレメトリセンサーグループで設定される対応するXPathを示しています。

OID名	OID番号	OIDの説明	XPATH
tcpInErrs	1.3.6.1.2.1.6.14	エラーで受信したセグメントの総数（不正なTCPチェックサムなど）。	Cisco-IOS-XR-ip-tcp-oper:tcp/nodes/node/status/ipv4-traffic/tcp-checksum-error-packets
tcpInSegs	1.3.6.1.2.1.6.10	受信したセグメントの総数（エラーで受信したセグメントを含む）。この数には、現在確立されている接続で受信されたセグメントが含まれます。	Cisco-IOS-XR-ip-tcp-oper:tcp/nodes/node/status/ipv4-traffic/tcp-input-packets
tcpOutSegs	1.3.6.1.2.1.6.11	現在の接続のセグメントを含め、再送信されたオクテットのみを含むセグメントを除く、送信されたセグメントの合計数。	Cisco-IOS-XR-ip-tcp-oper:tcp/nodes/node/status/ipv4-traffic/tcp-output-packets

UDP-MIB

次の表は、OIDの名前と番号、およびUDP固有のカウンタに関連するモデル駆動型テレメトリセンサーグループで設定される対応するXPathを示しています。

OID名	OID番号	OIDの説明	XPATH
udpOutDatagrams	1.3.6.1.2.1.7.4	このエンティティから送信されたUDPデータグラムの合計数。	Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv4-traffic/udp-output-packets Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv6-traffic/udp-output-packets
udpNoPorts	1.3.6.1.2.1.7.2	宛先ポートにアプリケーションがなかった、受信したUDPデータグラムの合計数。	Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv4-traffic/udp-no-port-packets Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv6-traffic/udp-no-port-packets
udpInErrors	1.3.6.1.2.1.7.3	宛先ポートにアプリケーションが存在しない以外の理由で配信できなかった受信UDPデータグラムの数。	Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv4-traffic/udp-checksum-error-packets Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv6-traffic/udp-checksum-error-packets
udpInDatagrams	1.3.6.1.2.1.7.1	UDPユーザに配信されたUDPデータグラムの合計数。	Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv4-traffic/udp-input-packets Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/status/ipv6-traffic/udp-input-packets

```
s/ipv4-traffic/udp-input-  
packets  
Cisco-IOS-XR-ip-udp-  
oper:/udp/nodes/node/st  
s/ipv6-traffic/udp-input-  
packets
```

SNMPトラップの移行

SNMPトラップは、管理対象デバイスの動的イベントによってトリガーされるメッセージです。したがって、これらのメッセージは、前に説明したEDTの概念に類似した動作をします。

設定側では、MDTはEDTと同じ構造を使用できます。これは、ダイヤルインまたはダイヤルアウトの選択または機能に関するテレメトリコレクタの実装によって異なります。

セキュリティに関する考慮事項

SNMPv2では、認証/認可メカニズムとしてコミュニティだけが使用されます。ただし、SNMPv3については前に説明したように、情報を保護するために、認証にクレデンシャルを使用し、AES暗号化モデルを使用できます。

テレメトリ方式では、IOS XRは証明書に基づくgRPC/TLS技術を使用して認証を実行できます。これらの証明書は、中央の信頼ポイント（CAサーバなど）とともに使用できます。信頼関係を構築するプロセスの後、すべてのテレメトリメッセージがgRPCセッション内で送信されます。このセッションはTLSで暗号化され、SNMPv3と同じ利点を実現します。