

Nexusプラットフォームでのテレメトリの設定と確認

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[トポロジ](#)

[Nexusとレシーバの接続](#)

[背景説明](#)

[従来のポーリングとストリーミングテレメトリ](#)

[実際の使用例](#)

[テレメトリ処理の概要](#)

[テレメトリアーキテクチャと用語](#)

[データ収集](#)

[データ管理エンジン : DME](#)

[管理オブジェクトと管理情報ツリー](#)

[識別名\(Distinguished Name\):DN](#)

[センサバス](#)

[データのエンコード](#)

[データ転送](#)

[テレメトリ受信機](#)

[通知先グループ](#)

[センサグループ](#)

[定期購読](#)

[定期的およびイベントベースのテレメトリ](#)

[定期的なテレメトリ](#)

[イベントベースのテレメトリ](#)

[ラボで使用する収集動作](#)

[テレメトリレシーバの準備](#)

[この実習で使用するレシーバ](#)

[NX-OSでのストリーミングテレメトリの設定](#)

[テレメトリ機能の有効化](#)

[宛先グループの設定](#)

[センサグループ1の設定](#)

[サブスクリプション1の設定](#)

[センサグループ2の設定](#)

[サブスクリプション2の設定](#)

[ループバックインターフェイスの設定](#)

[センサグループ3の設定](#)

[サブスクリプション3の設定](#)

[初期テレメトリ設定の完了](#)

[定期的なテレメトリの確認](#)

[テレメトリトランスポートの確認](#)

[定期的なセンサーグループの確認](#)

[データ収集の検証](#)

[受信側での定期的なデータの確認](#)

[出力と動作情報の確認](#)

[定期的なテレメトリ検証の概要](#)

[イベントベースのテレメトリの確認](#)

[イベントベースのサブスクリプションの確認](#)

[イベントベースの更新の生成](#)

[初期スナップショットおよびイベントメッセージの確認](#)

[イベント収集の統計情報の確認](#)

[イベントコレクタエラーの確認](#)

[イベントベースのテレメトリ検証の概要](#)

[基本的なトラブルシューティング](#)

[現在のトランスポート状態の確認](#)

[データ収集カウンタの確認](#)

[コレクションがスキップされた理由の特定](#)

[転送エラーの検査](#)

[トランスポート回復の確認](#)

[設定およびイベントコレクタのエラーの確認](#)

[トラブルシューティングワークフロー](#)

[実用アプリケーションの例：システムリソースのモニタ](#)

[システムリソーステレメトリの設定](#)

[リソースパスラベルの確認](#)

[システムリソース遠隔測定の確認](#)

[NX-OSのシステムリソースとの比較](#)

[受信側でのリソースデータの確認](#)

[時間の経過に伴うリソース使用率の監視](#)

[結論](#)

概要

このドキュメントでは、Cisco NX-OSを実行しているCisco Nexus 9000スイッチでストリーミングテレメトリを実装および確認する方法について説明します。

前提条件

要件

次の項目に関する基本的な知識が推奨されます。

- Cisco NX-OSのCLI
- 基本的なレイヤ3 IP接続
- 基本的なネットワークモニタリングの概念
- ネットワークテレメトリの概念に関する基礎知識

使用するコンポーネント

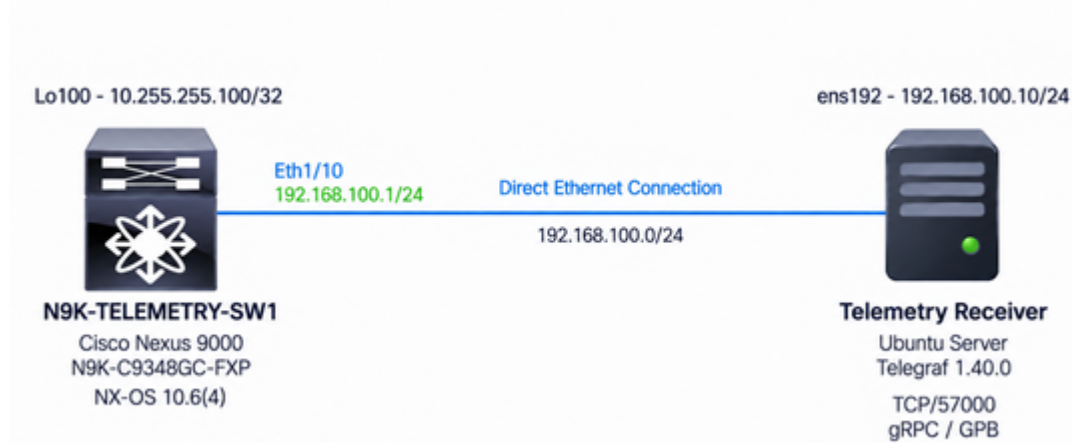
このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づくものです。

コンポーネント	プラットフォーム/ソフトウェア	バージョン/値	目的
N9KテレメトリSW1	N9K-C9348GC-FXP	10.6(4)	テレメトリソース
テレメトリ受信機	Ubuntuサーバ	22.04.5	外部テレメトリレシーバ
テレメトリアプリケーション	テレグラフ	1.40.0	Google Protocol Buffers(GPB)-over-gRPCテレメトリレシーバ
リスニングポート	TCP	57000	テレメトリ受信ポート
レシーバ出力フォーマット	JavaScriptオブジェクト表記(JSON)	—	人間が判読可能なテレメトリ出力

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

トポロジ

この実習では、外部テレメトリレシーバとして動作するUbuntuサーバに直接接続されたCisco Nexus 9000スイッチを使用します。



Nexusスイッチとテレメトリレシーバは、Ethernet1/10およびens192を介して直接接続されます。

Nexusとレシーバの接続

Ethernet1/10は、NexusスイッチとUbuntuレシーバ間のレイヤ3接続を提供します。

インターフェイスの設定は次のとおりです。

```
N9K-TELEMETRY-SW1# show running-config interface ethernet1/10 interface Ethernet1/10 description TELEMETRY-COLLECTOR ip address 192.168.100.1/24 no shutdown
```

Ubuntu受信機は次を使用します。インターフェイス： ens192 IPアドレス： 192.168.100.10/24

背景説明

ネットワークのモニタリングと可視性は、ネットワークの状態を理解し、異常な動作を特定し、ネットワークの問題をトラブルシューティングするための基盤となります。

Cisco NX-OSには、CLI、Simple Network Management Protocol(SNMP)、Syslogなど、Nexusスイッチから運用情報を取得またはエクスポートするための複数のメカニズムがあります。SNMPポーリングやCLIベースの収集などの従来のモニタリングワークフローは、一般にプルモデルに依存しています。プルモデルでは、外部モニタリングシステムがネットワークデバイスに情報を定期的に要求します。

ストリーミングテレメトリはプッシュモデルを導入します。プッシュモデルでは、ネットワークデバイスが、設定されたサブスクリプションに基づいて、選択された運用データを外部受信者に送信します。これにより、監視システムがデバイスを継続的にポーリングしなくても、ネットワーク情報を収集するための構造化されたメカニズムが提供されます。

従来のポーリングとストリーミングテレメトリ

従来のポーリングモデルでは、監視システムはネットワークデバイスから情報を取得するタイミングを決定します。

たとえば、ネットワーク管理システム(NMS)は60秒ごとにインターフェイス統計情報を要求できます。これにより、デバイスの状態のスナップショットが定期的に作成されますが、モニタリングアプリケーションで利用できる可視性は、設定されたポーリング間隔に直接関係します。

ストリーミングテレメトリでは、Nexusスイッチが、設定されたサブスクリプションに従って、選択された情報をテレメトリの受信者に送信します。

基本的な違いは次のとおりです。

従来のポーリング	ストリーミングテレメトリ
クライアントが要求を開始します。	ネットワークデバイスがデータを送信します。
プルモデル	プッシュモデル
データはポーリング間隔に従って取得されます。	データはテレメトリサブスクリプションに従って送信されます。
通常、定期的なスナップショットを提供します。	定期的なイベントベースの収集をサポートします。
モニタリングシステムは、デバイスに情報を要求します。	デバイスは、選択した情報を受信者にストリーミングします。

ストリーミングテレメトリは、必ずしも従来のモニタリングメカニズムに取って代わるわけではありません。CLI、SNMP、Syslog、およびテレメトリは共存でき、異なる運用目的で動作できます。

主な違いは、データ収集モデルです。

実際の使用例

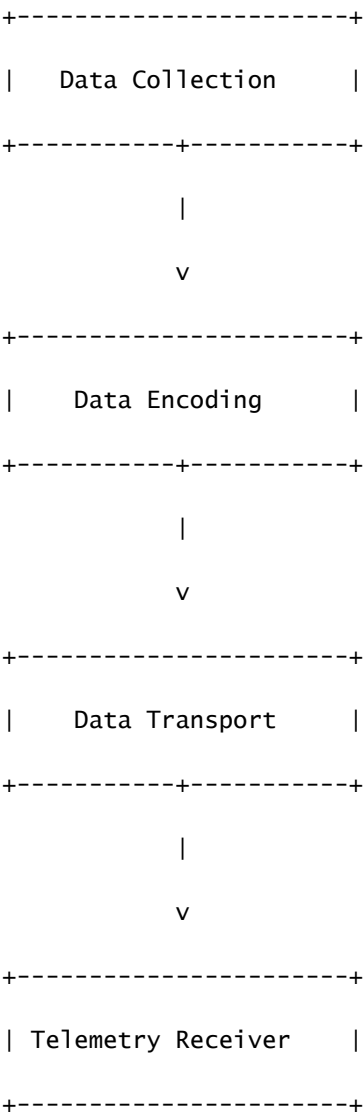
実稼働環境では、通常、ストリーミングテレメトリを使用して、ネットワークとシステムの状態を継続的に可視化します。

一般的なユースケースには、インターフェイスカウンタとステータス変更の監視、CPUやメモリの使用率などのシステムリソース、仮想拡張LAN(VXLAN)ピアやボーダーゲートウェイプロトコル(BGP)ピアの状態などのデータセンターファブリック情報が含まれます。また、イベントベースのテレメトリを使用して、変更が発生した際の運用状態の変化をレポートすることもできます。

エクスポートされたテレメトリデータは、ダッシュボード、アラート、履歴分析、およびトラブルシューティング用の外部モニタリングおよび監視プラットフォームで使用できます。

テレメトリ処理の概要

NX-OSでのストリーミングテレメトリは、大まかに次の4つの主要な段階と理解できます。



これらの段階では、次の4つの基本的な質問に答えます。

- データ収集：どのような情報が収集されますか。
- データのエンコード：情報はどのように表されますか。
- データ転送：テレメトリデータはどのように配信されますか。
- テレメトリレシーバ：テレメトリデータはどこに送信されますか。

テレメトリアーキテクチャと用語

データ収集

第1段階では、スイッチから収集する必要がある情報を決定します。

このドキュメントの例では、テレメトリデータはData Management Engine(DME)から収集されます。

データ管理エンジン : DME

Data Management Engine(DME)は、NX-OS内の設定および運用情報の構造化された表現を維持します。

デバイス情報をCLIテキストとしてのみ表す代わりに、DMEは情報を階層型パスを介してアクセス可能な管理オブジェクトとして編成します。

以下に、いくつかの例を示します。

```
sys/intf/phys-[eth1/10]
```

Ethernet1/10に関連付けられているDMEオブジェクトを示します。

この実習で使用する他のDMEパスは次のとおりです。

```
sys/intf/phys-[eth1/10]/dbgIfIn  
sys/intf/phys-[eth1/10]/dbgIfOut  
sys/intf/phys-[eth1/10]/phys  
sys/intf/lb-[1o100]
```

各パスは、DMEを介して利用可能な異なるオブジェクトまたは情報の一部を表します。これらは、ラボの設定全体で使用されているのと同じパスです。

管理オブジェクトと管理情報ツリー

DMEデータベースは、管理対象オブジェクト(MO)で構成されます。

管理オブジェクトは、NX-OS管理モデル内の次のようなエンティティを表します。

- 物理インターフェイス
- ループバックインターフェイス
- インターフェイス入力統計情報
- インターフェイス出力統計情報

管理オブジェクトは、管理情報ツリー(MIT)で階層的に編成されます。

この実習で使用するオブジェクトを簡略化して表すと、次のようになります。

```
sys
|
+-- intf
|
+-- phys-[eth1/10]
| |
| +-- dbgIfIn
| +-- dbgIfOut
| +-- phys
|
+-- lb-[1o100]
```

この階層は、テレメトリセンサーパスがDME内の特定の情報を識別する方法を理解するのに役立ちます。

識別名(Distinguished Name):DN

各マネージドオブジェクトは、識別名(DN)によって一意に識別できます。

DNは、DMEツリーのルートからターゲットオブジェクトまでの階層パスを表します。

For example: sys/intf/lb-[1o100]

Loopback100管理対象オブジェクトを識別します。

同様に、

```
sys/intf/phys-[eth1/10]/dbgIfIn
```

Ethernet1/10に関連付けられている入力統計オブジェクトを示します。

DNを理解する簡単な方法は、DNをDME階層内のオブジェクトの完全なアドレスと見なすことです。

センサーパス

センサーパスは、テレメトリサブスクリプションについてNX-OSが監視する必要がある情報を特定します。

最初のラボの例では、DME識別名(DN)がセンサーパスとして使用されます。後の実例で、定義済みリソースパスラベルを示します。

以下に、いくつかの例を示します。

```
sensor-group 2
path sys/intf/phys-[eth1/10]/dbgIfIn
path sys/intf/phys-[eth1/10]/dbgIfOut
path sys/intf/phys-[eth1/10]/phys
```

これらのパスは、Ethernet1/10の入力統計情報、出力統計情報、および動作情報を提供します。

データのエンコード

NX-OSが要求された情報を収集した後、送信する前にデータをエンコードする必要があります。

この実習で使用するエンコーディングは、Google Protocol Buffers(GPB)です。

宛先の設定では、次の項目を指定します。

```
ip address 192.168.100.10 port 57000 protocol gRPC encoding GPB
```

GPBは、収集されたテレメトリ情報がメッセージでどのように表されるかを定義します。

符号化と転送は別の機能です。GPBはデータ表現を定義し、転送メカニズムはメッセージの配信方法を決定します。

データ転送

この実習で使用するトランスポートプロトコルはgRPCです。

Nexusスイッチは、GPBエンコードされたテレメトリデータを次の宛先に送信します。

192.168.100.10:57000

grpcを使用します。

したがって

GPBは、テレメトリ情報のエンコード方法を定義します。

gRPCは、テレメトリメッセージを受信者に配信するために使用される転送メカニズムを提供します。

ストリーミングテレメトリで使用されるgRPCトランスポートは、gRPCネットワーク管理インターフェイス(gNMI)やgRPCネットワーク操作インターフェイス(gNOI)などのサービスで 사용되는NX-OS gRPCエージェントと混同しないでください。

テレメトリ受信機

テレメトリレシーバは、テレメトリストリームを受信して処理する外部システムまたはアプリケーションです。

この実習では、Telegrafを実行するUbuntuサーバをレシーバとして使用します。受信側はTCPポート57000でリッスンし、Nexusスイッチによって生成されたGPB-over-gRPCテレメトリストリームを受け入れます。

この実習で使用するレシーバの実装については、後述の「テレメトリレシーバの準備」の項で説明します。

通知先グループ

宛先グループは、テレメトリデータを送信する必要がある場所と、どのように転送する必要があるかを定義します。

このラボでは次を使用します。

```
destination-group 1
ip address 192.168.100.10 port 57000 protocol gRPC encoding GPB
use-vrf default
```

これは、テレメトリ配信に使用される受信者アドレス、宛先ポート、トランスポートプロトコル、エンコーディング、およびVirtual Routing and Forwarding(VRF)インスタンスを定義します。

センサーグループ

センサーグループは、監視する必要がある情報を定義します。

以下に、いくつかの例を示します。

```
sensor-group 2
path sys/intf/phys-[eth1/10]/dbgIfIn
path sys/intf/phys-[eth1/10]/dbgIfOut
path sys/intf/phys-[eth1/10]/phys
```

この例では、センサーグループ2はEthernet1/10の入力統計情報、出力統計情報、および動作インターフェイス情報を監視します。dbgIfInパスは入力インターフェイスの統計情報を提供し、dbgIfOutは出力インターフェイスの統計情報を提供し、physはインターフェイスの動作情報を提供します。

センサーグループには複数の関連するセンサーパスを含めることができるため、次の質問に回答します。

どのようなデータが収集されますか。

定期購読

サブスクリプションは、センサーグループを宛先グループに関連付け、収集動作を定義します。

以下に、いくつかの例を示します。

```
subscription 2
dst-grp 1
snsr-grp 2 sample-interval 60000
```

この例の設定は次のとおりです。

- センサーグループ2は、収集する情報を決定します。
- サンプル間隔によって、収集の動作が決まります。
- 宛先グループ1は、情報が送信される場所と方法を決定します。

したがって、サブスクリプションは次の主要なテレメトリ設定要素を接続します。

センサーグループ+収集動作+宛先グループ

定期的およびイベントベースのテレメトリ

Cisco NX-OSストリーミングテレメトリは、DMEベースのサブスクリプションの定期的およびイベントベースの収集をサポートします。

収集動作は、サブスクリプション内のセンサーグループに関連付けられたsample-intervalによって制御されます。

定期的なテレメトリ

定期的なテレメトリにより、NX-OSは設定された間隔で監視対象の情報を収集して送信します。

サンプリング間隔はミリ秒単位で指定します。

For example: snsr-grp 2 sample-interval 60000

60秒の収集間隔を設定します。

定期的なテレメトリは、継続的に変化し、通常は時間の経過とともに分析される次のような情報に役立ちます。

- インターフェイスパケットカウンタ
- バイトカウンタ
- エラーおよび破棄
- 運用統計情報
- リソース使用率

このラボ演習では、Subscription 1とSubscription 2は定期的な収集を使用します。

Subscription 1は10秒ごとにEthernet1/10インターフェイスオブジェクトを収集し、Subscription 2は60秒ごとにインターフェイス統計情報と動作情報を収集します。

イベントベースのテレメトリ

イベントベースのテレメトリでは、NX-OSは繰り返し収集タイマーを使用しません。

DMEベースのテレメトリの場合、イベントベースの動作は次のように設定されます。

```
sample-interval 0
```

監視オブジェクトが変更されると、テレメトリはその変更に関連付けられた更新を生成できます。

この収集方法は、次のような情報に役立ちます。

- 設定変更
- 管理状態の変更
- オブジェクトの作成または削除
- モニタ対象のDMEオブジェクトでの他の変更

このラボ演習では、Subscription 3はLoopback100 DMEオブジェクトをモニタします。

```
sys/intf/lb-[lo100]:
snsr-grp 3 sample-interval 0
```

したがって、監視されるLoopback100オブジェクトに対する変更は、このドキュメントの後半でイベントベースのテレメトリを示すために使用されます。

ラボで使用される収集動作

ラボの初期設定で使用された3つのサブスクリプションの概要を次に示します。

定期購読	監視情報	サンプル間隔	収集動作
1	Ethernet1/10インターフェイスオブジェクト	10000 ms	定期的
2	Ethernet1/10の統計情報と動作状態	60000 ms	定期的
3	Loopback100オブジェクト	0	イベントベース

主な違いは、テレメトリデータの生成に使用されるトリガーです。

定期的なテレメトリ	イベントベースのテレメトリ
設定されたタイマーを使用します。	定期的なタイマーを使用しない。
サンプル間隔> 0	サンプル間隔0
繰り返しサンプルを生成します。	監視対象オブジェクトの変更時に更新を生成します。
一般に、カウンタと統計情報に使用されます。	通常、設定や状態の変更に使用されます。

設定セクションと検証セクションでは、上で定義したサブスクリプションを使用した両方の収集動作を示します。

テレメトリレシーバの準備

ストリーミングテレメトリには、Nexusスイッチから送信されたテレメトリデータを受信して処理できる外部レシーバが必要です。

このドキュメントで使用するGPB-over-gRPCトランスポートに対して、受信側は次の機能を備えている必要があります。

- gRPC接続を受信しています
- 設定された宛先ポートでのリスニング
- GPBテレメトリメッセージのデコード
- 受信データの処理または保存

テレメトリレシーバの実装は、このドキュメントで説明されているNX-OSテレメトリ設定とは無関係です。



注：サードパーティ製テレメトリ受信機ソフトウェアのインストール、設定、操作、およびトラブルシューティングについては、この文書では説明しません。設定とサポート情報については、受信側のベンダーが提供するドキュメントを参照してください。

この実習で使用するレシーバ

このラボでは、デモンストレーションの目的で、テレメトリレシーバとしてTelegrafを実行する外部Ubuntuサーバを使用します。

受信パラメータは次のとおりです。

パラメータ	値
受信者アドレス	192.168.100.10
トランスポート	gRPC
リスニングポート	TCP/57000
符号化	ギガバイト

受信側ソフトウェアの設定は、このドキュメントでは説明しません。



注：このドキュメントに示すレシーバ出力の例は、各検証ステップに関連するフィールドを強調表示するようにフィルタリングされています。

Nexusスイッチでストリーミングテレメトリを設定する前に、次のことを確認します。

- Nexusスイッチとレシーバの間にIP接続が存在する。
- 受信側アプリケーションが実行中です。
- 設定された宛先ポートはリスニング中です。
- 受信側は、設定されたトランスポートプロトコルをサポートします。
- 受信側は、設定されたテレメトリエンコーディングをサポートします。

このラボでは、NexusスイッチのEthernet1/10は192.168.100.1/24を使用し、テレメトリレシーバは192.168.100.10/24を使用します。テレメトリ固有の動作をトラブルシューティングする前に、基本的なレイヤ3到達可能性を確認する必要があります。

受信側が到達可能で、TCPポート57000でGPB-over-gRPCテレメトリを受け入れる準備ができたら、NX-OSテレメトリ設定を適用できます。

NX-OSでのストリーミングテレメトリの設定

この設定では、次の3つの主要コンポーネントを使用します。

- 宛先グループ：テレメトリデータの送信先と転送方法を定義します。
- Sensor Group:収集する情報を定義します。
- サブスクリプション：センサーグループを宛先グループに関連付け、収集動作を定義します。

ラボでは、次のテレメトリの宛先を使用します。

パラメータ	値
宛先	192.168.100.10:57000
トランスポート	gRPC
符号化	ギガバイト
VRF	デフォルト

ラボの初期設定では、次の3つのサブスクリプションを使用します。

定期購読	センサー	コレクション型	サンプル間隔
1	Ethernet1/10インターフェイスオブジェクト	定期的	10000 ms
2	Ethernet1/10の統計情報と動作状態	定期的	60000 ms
3	Loopback100オブジェクト	イベントベース	0

テレメトリ機能の有効化

まず、ストリーミングテレメトリをグローバルに有効にする必要があります。

```
N9K-TELEMETRY-SW1# configure terminal
N9K-TELEMETRY-SW1(config)# feature telemetry
```

テレメトリコンフィギュレーションモードを開始します。

```
N9K-TELEMETRY-SW1(config)# telemetry
N9K-TELEMETRY-SW1(config-telemetry)#
```

残りのテレメトリ設定は、このモードで実行されます。

宛先グループの設定

宛先グループは、外部テレメトリ受信者を識別し、テレメトリデータの送信に使用されるトランスポートとエンコーディングを定義します。

設定例:

```
N9K-TELEMETRY-SW1(config-telemetry)# destination-group 1
N9K-TELEMETRY-SW1(conf-tm-dest)# ip address 192.168.100.10 port 57000 protocol gRPC encoding GPB
N9K-TELEMETRY-SW1(conf-tm-dest)# use-vrf default
```

この設定では次の項目を定義します。

- 宛先アドレス : 192.168.100.10
- 宛先ポート : 57000
- トランスポート : gRPC
- エンコード : GPB
- VRF : デフォルト

デフォルトのVRFでは、テレメトリレシーバがEthernet1/10を介して到達可能なため、デフォルトのVRFが使用されます。

テレメトリトランスポート用に選択されたVRFは、設定されたレシーバへのIP到達可能性を提供する必要があります。

センサーグループ1の設定

センサーグループ1は、Ethernet1/10 DMEインターフェイスオブジェクトをモニタします。

```
N9K-TELEMETRY-SW1(config-telemetry)# sensor-group 1
N9K-TELEMETRY-SW1(conf-tm-sensor)# path sys/intf/phys-[eth1/10]
```

センサーパスは、DME階層内のEthernet1/10管理対象オブジェクトを識別し、そのオブジェクトに関連する一般的なインターフェイス情報を提供します。

サブスクリプション1の設定

センサーグループ1を宛先グループ1に関連付けます。

```
N9K-TELEMETRY-SW1(config-telemetry)# subscription 1
N9K-TELEMETRY-SW1(conf-tm-sub)# dst-grp 1
N9K-TELEMETRY-SW1(conf-tm-sub)# snsr-grp 1 sample-interval 10000
```

サンプリング間隔はミリ秒単位で指定します。

10000 ms = 10秒

したがって、Subscription 1は10秒ごとにEthernet1/10 DMEオブジェクトを定期的に収集し、テレメトリデータを宛先グループ1に送信します。

センサーグループ2の設定

センサーグループ2は、Ethernet1/10の統計情報と動作情報を収集します。

設定例:

```
N9K-TELEMETRY-SW1(config-telemetry)# sensor-group 2
N9K-TELEMETRY-SW1(conf-tm-sensor)# path sys/intf/phys-[eth1/10]/dbgIfIn
N9K-TELEMETRY-SW1(conf-tm-sensor)# path sys/intf/phys-[eth1/10]/dbgIfOut
N9K-TELEMETRY-SW1(conf-tm-sensor)# path sys/intf/phys-[eth1/10]/phys
```

センサーパスは次の機能を提供します。

センサーパス	情報
sys/intf/phys-[eth1/10]/dbgIfIn	入力インターフェイスの統計情報
sys/intf/phys-[eth1/10]/dbgIfOut	出力インターフェイスの統計情報
sys/intf/phys-[eth1/10]/phys	運用インターフェイス情報

センサーグループには複数の関連するセンサーパスを含めることができ、サブスクリプションは同じ監視対象インターフェイスから複数のカテゴリの情報を収集できます。

サブスクリプション2の設定

センサーグループ2を宛先グループ1に関連付けます。

```
N9K-TELEMETRY-SW1(config-telemetry)# subscription 2
N9K-TELEMETRY-SW1(conf-tm-sub)# dst-grp 1
N9K-TELEMETRY-SW1(conf-tm-sub)# snsr-grp 2 sample-interval 60000
```

設定されているサンプリング間隔は次のとおりです。

60000 ms = 60秒

したがって、Subscription 2は60秒ごとにEthernet1/10の統計情報と動作情報を収集します。

このサブスクリプションは、定期的なテレメトリ動作を確認するために、このドキュメントの後半で使用します。

ループバックインターフェイスの設定

ループバックインターフェイスは、追加の物理接続を必要とせずに、イベントベースのテレメトリを実証するために使用されます。

設定例:

```
N9K-TELEMETRY-SW1(config)# interface loopback100
N9K-TELEMETRY-SW1(config-if)# ip address 10.255.255.100/32
```

対応するDMEオブジェクトは次のとおりです。

```
sys/intf/lb-[1o100]
```

ループバックインターフェイスは、イベントベースのテレメトリ検証中に、制御された設定およ

び管理状態の変更を生成する簡単な方法を提供します。

センサーグループ3の設定

Loopback100 DMEセンサーパスを設定します。

```
N9K-TELEMETRY-SW1(config-telemetry)# sensor-group 3
N9K-TELEMETRY-SW1(conf-tm-sensor)# path sys/intf/lb-[l0100]
```

センサーグループ3はLoopback100管理対象オブジェクトを監視します。

サブスクリプション3の設定

センサーグループ3を宛先グループ1に関連付け、イベントベースの収集を設定します。

```
N9K-TELEMETRY-SW1(config-telemetry)# subscription 3
N9K-TELEMETRY-SW1(conf-tm-sub)# dst-grp 1
N9K-TELEMETRY-SW1(conf-tm-sub)# snsr-grp 3 sample-interval 0
```

DMEベースのサブスクリプションの場合、サンプル間隔を0に設定すると、イベントベースの動作が設定されます。

したがって、監視対象のLoopback100オブジェクトの下で行われた変更は、繰り返し収集タイマーを使用せずにテレメトリ通知を生成できます。

このサブスクリプションは、後で説明および管理状態の変更を示すために使用されます。

初期テレメトリ設定の完了

設定が完了したら、次のコマンドを使用して設定を確認します。

```
N9K-TELEMETRY-SW1# show running-config telemetry
feature telemetry
telemetry
  destination-group 1
```

```

ip address 192.168.100.10 port 57000 protocol gRPC encoding GPB
use-vrf default
sensor-group 1
  path sys/intf/phys-[eth1/10]
sensor-group 2
  path sys/intf/phys-[eth1/10]/dbgIfIn
  path sys/intf/phys-[eth1/10]/dbgIfOut
  path sys/intf/phys-[eth1/10]/phys
sensor-group 3
  path sys/intf/lb-[l0100]
subscription 1
  dst-grp 1
  snsr-grp 1 sample-interval 10000
subscription 2
  dst-grp 1
  snsr-grp 2 sample-interval 60000
subscription 3
  dst-grp 1
  snsr-grp 3 sample-interval 0

```

この時点で、スイッチには、設定済みのレシーバへのテレメトリデータの送信を開始するために必要な宛先、センサーグループ、およびサブスクリプションがあります。

次のセクションでは、Ethernet1/10サブスクリプションによって生成されたトランスポートセッションと定期的なテレメトリを確認します。

定期的なテレメトリの確認

テレメトリ設定の適用後、トランスポートセッションが確立され、定期的なセンサーグループがアクティブであり、テレメトリデータが収集されて受信者に配信されていることを確認します。

テレメトリトランスポートの確認

次のコマンドを実行します。

N9K-TELEMETRY-SW1# show telemetry transport

Session Id	Dst Grp	IP Address	Port	Encoding	Transport	Status
0	1	192.168.100.10	57000	GPB	gRPC	Connected

Retry buffer Size:

10485760

Event Retry Messages (Bytes):

0

Timer Retry Messages (Bytes):

0

Total Retries sent:

0

Total Retries Dropped:

0

N9K-TELEMETRY-SW1#

Connected状態は、設定済みのテレメトリレシーバへのgRPC転送セッションが確立されていることを示します。

関連するトランスポートパラメータも、設定された宛先グループに一致します。

定期的なセンサーグループの確認

使用目的:

N9K-TELEMETRY-SW1# show telemetry control database

Subscription Database size = 3

Subscription ID	Data Collector Type	Reachability	Configuration Method
3	DME	Reachable	CONFIG_DME-ADMIN
2	DME	Reachable	CONFIG_DME-ADMIN
1	DME	Reachable	CONFIG_DME-ADMIN

Sensor Group Database size = 3

Row ID	Sensor Group ID	Sensor Group type	Sampling interval(ms)	Linked subscriptions	SubID
1	3	Event /DME	0/No Timer	1	3
Collection Time in ms (Cur/Min/Max): 0/0/0					
Encoding Time in ms (Cur/Min/Max): 0/0/0					
Transport Time in ms (Cur/Min/Max): 2/1/414					
Streaming Time in ms (Cur/Min/Max): 2/1/414					

Collection Statistics:
collection_id_dropped = 0
last_collection_id_dropped = 0
drop_count = 0

Configuration method: CONFIG_DME-ADMIN

2	2	Timer /DME	60000/Running	1	2
Collection Time in ms (Cur/Min/Max): 0/0/1					
Encoding Time in ms (Cur/Min/Max): 0/0/1					
Transport Time in ms (Cur/Min/Max): 2/1/415					
Streaming Time in ms (Cur/Min/Max): 3/2/416					

Collection Statistics:
collection_id_dropped = 0
last_collection_id_dropped = 0
drop_count = 0

Configuration method: CONFIG_DME-ADMIN

3	1	Timer /DME	10000/Running	1	1
Collection Time in ms (Cur/Min/Max): 0/0/1					
Encoding Time in ms (Cur/Min/Max): 0/0/1					
Transport Time in ms (Cur/Min/Max): 2/1/415					
Streaming Time in ms (Cur/Min/Max): 3/2/417					

Collection Statistics:
collection_id_dropped = 0
last_collection_id_dropped = 0
drop_count = 0

Configuration method: CONFIG_DME-ADMIN

Row ID	Subscribed	Linked Groups	Sec Groups	Retrieve level	Path(GroupId) : Query : Filter
1	No	1	0	Self	sys/intf/phys-[eth1/10](1) : NA : NA
<snip>					
2	No	1	0	Self	sys/intf/phys-[eth1/10]/dbgIfIn(2) : NA : NA
<snip>					
4	No	1	0	Self	sys/intf/phys-[eth1/10]/dbgIfOut(2) : NA : NA

センサーグループデータベースは、定期的なセンサーグループがアクティブであることを確認します。

センサーグループ	Type	サンプリング間隔	定期購読
1	タイマー/DME	10000ミリ秒/実行中	1
2	タイマー/DME	60000ミリ秒/実行中	2

センサーグループ1は、10秒ごとにEthernet1/10インターフェイスオブジェクトを収集します。

センサーグループ2は、60秒ごとにEthernet1/10の統計情報と動作情報を収集します。

同じコマンドで、設定されているセンサーパスも表示されます。

```
sys/intf/phys-[eth1/10]
sys/intf/phys-[eth1/10]/dbgIfIn
sys/intf/phys-[eth1/10]/dbgIfOut
sys/intf/phys-[eth1/10]/phys
```

この実習で使用したセンサーパスでは、収集時間とエンコード時間は通常0 ~ 1ミリ秒の間で、テレメトリメッセージのドロップは確認されませんでした。

データ収集の検証

使用目的:

```
N9K-TELEMETRY-SW1# show telemetry data collector brief
```

Row ID	Collector Type	Successful	Payloads	Failed	Skipped
1	YANG	0	0	0	0
2	DME	513	513	0	44
3	NX-API	0	0	0	0

```
N9K-TELEMETRY-SW1#
```

ラボでは次のように報告されました。

DME成功コレクション： 513ペイロード： 513失敗： 0スキップ： 44ドロップ： 0

成功カウンタとペイロードカウンタは、DMEテレメトリデータが収集され、テレメトリペイロードが生成されていることを確認します。

44個のスキップされた収集は、テレメトリの宛先がラボ中に一時的に使用できなかった間に観察された履歴カウンタです。これらのカウンタについては、「基本的なトラブルシューティング」のセクションで後ほど説明します。

その他のセンサーごとのパス情報については、次を使用してください。

```
N9K-TELEMETRY-SW1# show telemetry data collector details
```

このコマンドでは、収集の成功、失敗、スキップ、またはドロップの原因となった、設定済みのセンサーパスを特定できます。

受信側での定期的なデータの確認

Subscription 2は、60秒ごとにEthernet1/10の統計情報を収集します。

テレメトリの受信者は、21:44:45 Coordinated Universal Time (UTC ; 世界標準時) に次の入力統計情報を報告しました。

```
timestamp: 2026-09-17T21:44:45Z
source: N9K-TELEMETRY-SW1
subscription: 2
path: sys/intf/phys-[eth1/10]/dbgIfIn
```

```
broadcastPkts: 5074
discards: 0
```

errors: 0
multicastPkts: 143403
octets: 12870535
ucastPkts: 31256

1分後、別のサンプルが受信されました。

timestamp: 2026-09-17T21:45:45Z
source: N9K-TELEMETRY-SW1
subscription: 2
path: sys/intf/phys-[eth1/10]/dbgIfIn

broadcastPkts: 5075
discards: 0
errors: 0
multicastPkts: 143430
octets: 12875428
ucastPkts: 31292

両方のサンプル間でのカウンタの変更を次にまとめます。

カウンタ	21:44:45	21:45:45
ユニキャストパケット	31,256	31,292
マルチキャスト パケット	143,403	143,430
ブロードキャストパケット	5,074	5,075
オクテット	12,870,535	12,875,428
Errors	0	0
Discards	0	0

タイムスタンプは約60秒間隔で、サブスクリプション2に設定されたサンプリング間隔と一致します。

パケットカウンタとバイトカウンタの増加は、更新されたインターフェイス統計情報が収集され、受信側に配信されていることの確認にもなります。

出力と動作情報の確認

dbgIfOutセンサーパスは、Ethernet1/10の出力統計情報を提供します。

21:45:45 UTCに収集されたサンプルの報告：

broadcastPkts: 8
discards: 0
errors: 0
multicastPkts: 4712
octets: 2444721
ucastPkts: 4092

物理センサーパスは、同じインターフェイスの動作属性を提供します。

受信側は次のように報告しました。

adminSt: up
operSt: up
operSpeed: 1G
operDuplex: full
operMtu: 1500
operDescr: TELEMETRY-COLLECTOR

これらのサンプルは、Sensor Group 2が3つの設定済みDMEセンサーパスを通じて、インターフェイス統計情報と動作情報の両方を提供していることを確認しています。

定期的なテレメトリ検証の概要

定期的なテレメトリ検証によって次のことが確認されます。

確認	結果
gRPC転送セッション	接続中
GPB符号化	確認済
センサーグループ1	10000ミリ秒で実行
センサーグループ2	60000ミリ秒で実行
DMEコレクション	成功
失敗したコレクション	0
ペイロードのドロップ	0
定期的なレシーバサンプル	Received
Interface Counters	サンプル間の更新

イベントベースのテレメトリの確認

定期的なテレメトリを確認した後、Subscription 3を使用して、Loopback100管理対象オブジェクトに対して制御された変更を生成することにより、イベントベースのテレメトリを検証します。

イベントベースのサブスクリプションの確認

使用目的:

N9K-TELEMETRY-SW1# show telemetry control database

Subscription Database size = 3

Subscription ID	Data Collector Type	Reachability	Configuration Method
3	DME	Reachable	CONFIG_DME-ADMIN
2	DME	Reachable	CONFIG_DME-ADMIN
1	DME	Reachable	CONFIG_DME-ADMIN

<snip>

Sensor Group Database size = 3

Row ID	Sensor Group ID	Sensor Group type	Sampling interval(ms)	Linked subscriptions	SubID
1	3	Event /DME	0/No Timer	1	3

<snip>

Sensor Path Database size = 5

Row ID	Subscribed	Linked Groups	Sec Groups	Retrieve level	Path(GroupId) : Query : Filter
4	Yes	1	0	Self	sys/intf/1b-[1o100](3) : NA : NA

<snip>

Subscription Id: 3

Snapshot Stats:

Sent = 1
Error = 0
Drops = 0

The Sensor Group Database reports:

Sensor Group ID: 3
Sensor Group Type: Event / DME
Sampling Interval: 0 / No Timer
Subscription ID: 3

The Event / DME type and 0 / No Timer state confirm that Sensor Group 3 is configured for event-based c
The associated sensor path is:
sys/intf/1b-[1o100]

同じテレメトリデータベースで、センサーパスがSubscription 3を介してサブスクライブされていることが確認されます。

イベントベースの更新の生成

イベントの生成を確認するには、Loopback100の説明を次のように変更します。

```
N9K-TELEMETRY-SW1(config)# interface loopback100
N9K-TELEMETRY-SW1(config-if)# description TELEMETRY-EVENT-DEMO
```

テレメトリの受信者は次を報告しました：

```
{
"timestamp": "2026-09-17T21:41:12Z",
"source": "N9K-TELEMETRY-SW1",
"subscription": "3",
"event": {
"descr": "TELEMETRY-EVENT-DEMO",
"dn": "sys/intf/lb-[lo100]",
"status": "modified"
}
}
```

dnは監視対象のDMEオブジェクトを示し、descr属性は変更されたプロパティを示します。

次に、インターフェイスを管理上ディセーブルにして復元します。

```
N9K-TELEMETRY-SW1(config)# interface loopback100
N9K-TELEMETRY-SW1(config-if)# shutdown
```

次に実行するコマンド

```
N9K-TELEMETRY-SW1(config-if)# no shutdown
```

レシーバが両方の状態変化を検出した。

Timestamp	変更された属性	値
21:41:12	説明	テレメトリ – イベント – デモ
21:41:19	管理セット	ダウン
21:41:24	管理セット	up

各更新で同じ監視対象オブジェクトが参照されました：

```
sys/intf/1b-[1o100]
```

オブジェクトの状態が変更されたことを報告しました。

これらの結果から、監視対象の管理対象オブジェクトのさまざまな属性に対する変更によって、個々のテレメトリの更新が発生する可能性があることが確認されました。

初期スナップショットおよびイベントメッセージの確認

イベントベースのサブスクリプションが確立されると、NX-OSは監視対象オブジェクトの初期スナップショットを生成します。

レポートされたセンサーパスデータベース：

スナップショットの状態：

送信= 1エラー= 0ドロップ= 0

制御された3つの変更後、同じセンサーパスが次のように報告されました。

メッセージ統計：

送信= 3エラー= 0ドロップ= 0

したがって、結果は次のように要約できます。

収集	[Count]
初期スナップショット	1
説明の変更	1
管理状態ダウン	1
管理状態アップ	1
合計	4

最初のスナップショットは、サブスクリプションがアクティブになったときに監視対象オブジェクトの状態を表し、後続のメッセージはオブジェクトの変更に対応します。

イベント収集の統計情報の確認

```
N9K-TELEMETRY-SW1# show telemetry event collector stats
-----
Row ID      Collection Count  Latest Collection Time  Sensor Path(GroupId)
-----
1           4           Thu Sep 17 21:41:24.045 UTC sys/intf/lb-[l0100](3)
N9K-TELEMETRY-SW1#
```

ラボでは次のように報告されました。

```
Collection Count: 4
Latest Collection Time: Thu Sep 17 21:41:24.045 UTC
Sensor Path: sys/intf/lb-[l0100]
```

収集カウントは、1つの最初のスナップショットと、テスト中に生成された3つの変更に一致します。

イベントコレクタエラーの確認

```
N9K-TELEMETRY-SW1# show telemetry event collector errors
-----
Error Description                                Error Count
-----
Dme Event Subscription Init Failures              - 0
Event Data Enqueue Failures                      - 0
Event Subscription Failures                      - 0
Pending Subscription List Create Failures         - 0
Subscription Hash Table Create Failures          - 0
Subscription Hash Table Destroy Failures         - 0
Subscription Hash Table Insert Failures          - 0
Subscription Hash Table Remove Failures          - 0
N9K-TELEMETRY-SW1#
```

テスト中にイベントコレクタエラーは観察されませんでした。

イベントベースのテレメトリ検証の概要

イベントベースのテレメトリ検証によって、次のことが確認されます。

確認	結果
センサーグループ	3
センサーパス	sys/intf/lb-[lo100]
コレクション型	イベント/DME
サンプリング間隔	0 / タイマーなし
初期スナップショット	Sent
説明の変更	検出
adminStダウン	検出
adminStアップ	検出
コレクションの合計	4
イベントコレクタエラー	0

制御されたLoopback100の変更が正常に検出されたことで、サブスクリプション3が期待どおりに動作していることが確認されます。

次のセクションでは、ストリーミングテレメトリの動作を評価するために使用する、主な検証コマンドとトラブルシューティングコマンドについて検証します。

基本的なトラブルシューティング

テレメトリデータが期待どおりに受信されない場合は、問題がトランスポートセッション、データ収集、センサー設定、または外部レシーバのいずれに関連しているのかを特定することによってトラブルシューティングを開始します。

このワークフローでは、NX-OSが44個のスキップされた収集と1個の履歴gRPC送信エラーを報告した、このラボ演習で観察された問題を使用します。

現在のトランスポート状態の確認

```
N9K-TELEMETRY-SW1# show telemetry transport
Session Id  Dst Grp  IP Address      Port      Encoding      Transport      Status
-----
0           1         192.168.100.10  57000     GPB           gRPC           Connected
```

```
-----
Retry buffer Size:                10485760
Event Retry Messages (Bytes):    0
Timer Retry Messages (Bytes):    0
Total Retries sent:              0
Total Retries Dropped:          0
```

N9K-TELEMETRY-SW1#

最終確認中、テレメトリセッションは次のように報告しました。

```
Session ID: 0
Destination Group: 1
IP Address: 192.168.100.10
Port: 57000
Encoding: GPB
Transport: gRPC
Status: Connected
```

Connected状態は、トランスポートセッションが現在確立されていることを示します。

ただし、現在のセッション状態は、接続問題が以前に発生したかどうかを必ずしも示しているわけではありません。したがって、履歴の収集カウンタとトランスポートカウンタも確認します。

データ収集カウンタの確認

N9K-TELEMETRY-SW1# show telemetry data collector brief

Row ID	Collector Type	Successful	Payloads	Failed	Skipped
1	YANG	0	0	0	0
2	DME	513	513	0	44
3	NX-API	0	0	0	0

N9K-TELEMETRY-SW1#

ラボでは次のように報告されました。

```
DME Successful Collections: 513
Payloads: 513
Failed: 0
Skipped: 44
Dropped: 0
```

SuccessfulカウンタとPayloadカウンタは、DMEテレメトリ収集とペイロード生成が行われたことを確認します。

ただし、Skippedカウンタは、スケジュールされた44個のコレクションが実行されなかったことを示します。

影響を受けたセンサーパスを特定するには、次のコマンドを使用します。

```
N9K-TELEMETRY-SW1# show telemetry data collector details
```

Row ID	Successful	Payloads	Failed	Skipped	Dropped	Sensor Path(G
1	414	414	0	29	0	sys/intf/phys
2	54	54	0	5	0	sys/intf/phys
3	54	54	0	5	0	sys/intf/phys
4	54	54	0	5	0	sys/intf/phys

```
N9K-TELEMETRY-SW1#
```

報告された詳細な出力：

センサーパス	スキップ
sys/intf/phys-[eth1/10]	29
sys/intf/phys-[eth1/10]/dbgIfIn	5
sys/intf/phys-[eth1/10]/dbgIfOut	5
sys/intf/phys-[eth1/10]/phys	5
合計	44

スキップされた収集は定期的なセンサーパスに分散されており、問題が単一のDMEオブジェクトに分離されなかったことを示しています。



注：コレクションカウンタは累積されます。ゼロ以外の履歴カウンタは、必ずしも同じ条件が現在存在することを示すわけではありません。

コレクションがスキップされた理由の特定

```
N9K-TELEMETRY-SW1# show telemetry control stats
```

Error Description	Error Count
-------------------	-------------

```
<snip>
Collections skipped due to destination unreachable      44
<snip>
N9K-TELEMETRY-SW1#
```

次の出力は、スキップされたコレクションの理由を直接示しています。

宛先到達不能= 44

この値は、DMEデータコレクタによって報告された44個のスキップされたコレクションと一致します。

これにより、調査をセンサーパス自体からテレメトリの宛先およびトランスポートパスに向けて移動できます。

転送エラーの検査

show telemetry transportで報告されるセッションIDを使用します。

```
N9K-TELEMETRY-SW1# show telemetry transport 0 errors
Session Id:                0
Connection Errors
  Connection Error Count:   0
Transmission Errors
  Tx Error Count:          1
  Last Tx Error:            Thu Sep 17 21:22:16.008 UTC
  Last Tx Return Code:      UNAVAILABLE
N9K-TELEMETRY-SW1#
```

ラボでは次のように報告されました。

```
Connection Error Count: 0
Tx Error Count: 1
Last Tx Error: Thu Sep 17 21:22:16.008 UTC
Last Tx Return Code: UNAVAILABLE
```

UNAVAILABLEリターンコードは、テレメトリの宛先に関連付けられたgRPC送信エラーを記録します。

このラボ演習では、外部レシーバを意図的に停止し、設定の変更中に再起動しました。その間、

NX-OSはテレメトリの宛先に到達できませんでした。これは、上記のdestination-unreachable収集カウンタに対応しています。

重要な相関関係は次のとおりです。

観察	結果
スキップされたコレクション	44
宛先到達不能	44
トランスポートTxエラー	1
最後のTxリターンコード	使用不可
現在のトランスポートの状態	接続中

一致するskippedカウンタとdestination-unreachableカウンタにより、コレクションがスキップされた理由を直接確認できます。

転送エラー履歴には、同じラボ演習で観察された転送障害に関する追加情報が含まれています。

トランスポート回復の確認

レシーバへの接続が回復したら、次のコマンドを使用します。

```
N9K-TELEMETRY-SW1# show telemetry transport 0 stats
Session Id:                0
Connection Stats
  Connection Count          3
  Last Connected:           Thu Sep 17 21:27:16.010 UTC
  Disconnect Count          0
  Last Disconnected:        Never
Transmission Stats
  Compression:              disabled
  Source Interface:         not set()
  Transmit Count:           603
  Last TX time:             Thu Sep 17 21:51:36.008 UTC
  Min Tx Time:              1 ms
  Max Tx Time:              414 ms
  Avg Tx Time:              6 ms
  Cur Tx Time:              1 ms

Flow Stats
  Allowed Queued Msgs Size (bytes): 78643200
  Current Queued Msgs Size (bytes): 0
  Utilization (percent):         0
  Max Queued Msgs Size (bytes): 3849
  Total Queued Msgs Size (bytes): 865313
  Current Msgs Held (# Msgs):    0
```

<snip>

```
Connection Count: 3
Disconnect Count: 0
Transmit Count: 603
Average Transmission Time: 6 ms
Current Transmission Time: 1 ms
Current Queued Messages Size: 0
Transport Utilization: 0%
Messages Dropped: 0
Flow Control Last Applied: Never
```

最も重要な現況指標は以下の通りです。

指標	結果
転送ステータス	接続中
現在のキュー	0
ドロップされたメッセージ	0
フロー制御	適用なし

設定およびイベントコレクタのエラーの確認

```
N9K-TELEMETRY-SW1# show telemetry config errors
```

Row	ID	Path	Sensor Group	Error
-----	----	------	--------------	-------

Transport Errors

Destination group ID	Source Interface	Configured VRF	Correct VRF
N9K-TELEMETRY-SW1#			

イベントベースのテレメトリには、次を使用します。

N9K-TELEMETRY-SW1# show telemetry event collector errors

Error Description	Error Count
Dme Event Subscription Init Failures	- 0
Event Data Enqueue Failures	- 0
Event Subscription Failures	- 0
Pending Subscription List Create Failures	- 0
Subscription Hash Table Create Failures	- 0
Subscription Hash Table Destroy Failures	- 0
Subscription Hash Table Insert Failures	- 0
Subscription Hash Table Remove Failures	- 0
N9K-TELEMETRY-SW1#	

すべてのイベントコレクタエラーカウンタもゼロでした。

これらの結果は、スキップされた定期的な収集を調査する際に、構成およびイベントコレクタの障害を除外するのに役立ちます。

トラブルシューティングワークフロー

この調査で使用したコマンドの概要を次に示します。

コマンド	目的
テレメトリトランスポートの表示	現在のトランスポートセッションの状態を確認します。
テレメトリデータコレクタの概要を表示	成功、失敗、スキップ、またはドロップされたコレクションを特定します。
テレメトリデータコレクタの詳細の表示	影響を受けるセンサーパスを特定します。
テレメトリ制御統計情報の表示	コレクションがスキップされた理由を確認します。
show telemetry transport <session-id>エラー	トランスポート障害を検査します。
show telemetry transport <session-id>統計情報	トランスポートの回復、キュー、およびドロップを確認します。
show telemetry config errors (テレメトリ設定エラーの表示)	テレメトリ設定エラーを特定します。
テレメトリイベントコレクタエラーの表示	イベントコレクタエラーを特定します。

このラボ演習では、トラブルシューティングシーケンスによって、DMEセンサーパスや設定の障害ではなく、一時的なテレメトリ宛先の到達可能性状態が特定されました。

受信側が再度使用可能になると、テレメトリトランスポートが接続状態に戻り、収集が再開され、現在のキューまたはメッセージドロップ状態は確認されませんでした。

実用アプリケーションの例：システムリソースのモニタ

システムリソースのモニタリングは、ストリーミングテレメトリの一般的な使用例です。CPUとメモリの情報をNexusスイッチから外部モニタリングプラットフォームに定期的にエクスポートして、履歴分析、ダッシュボード、キャパシティモニタリング、アラートを実行できます。

Cisco NX-OSには、一般的に監視される情報に対して事前定義されたテレメトリパスラベルが用意されています。この例では、システムCPUとメモリの情報を収集するために、リソースパスラベルが使用されます。

システムリソーステレメトリの設定

リソースパスラベルを使用して、新しいセンサーグループを作成します。

```
N9K-TELEMETRY-SW1(config)# telemetry
N9K-TELEMETRY-SW1(config-telemetry)# sensor-group 4
N9K-TELEMETRY-SW1(conf-tm-sensor)# path resources
```

サブスクリプション4を作成し、センサーグループ4を宛先グループ1に関連付けます。

```
N9K-TELEMETRY-SW1(config-telemetry)# subscription 4
N9K-TELEMETRY-SW1(conf-tm-sub)# dst-grp 1
N9K-TELEMETRY-SW1(conf-tm-sub)# snsr-grp 4 sample-interval 10000
```

Sensor Group 4は、定義済みのリソースパスラベルを使用します。Subscription 4は、センサーグループを既存のテレメトリ宛先に関連付け、ゼロ以外のサンプリング間隔で定期的な収集を設定します。

関連する設定は次のとおりです。

```
telemetry
  destination-group 1
    ip address 192.168.100.10 port 57000 protocol gRPC encoding GPB
    use-vrf default

  sensor-group 4
    path resources

  subscription 4
    dst-grp 1
    snsr-grp 4 sample-interval 10000
```

リソースパスラベルの確認

次のコマンドを実行して、リソースパスラベルで表されるDMEパスを調べます。

```
N9K-TELEMETRY-SW1# show telemetry usability resources
```

```
1) label_name      : resources
   path_name       : sys/proc
   query_type      : poll
   <snip>

2) label_name      : resources
   path_name       : sys/procsys
   query_type      : poll
   <snip>

3) label_name      : resources
   path_name       : sys/procsys/systemem
   query_type      : event
   query_condition : query-target-filter=and(updated(procSysMem.memstatus),ne(procSysMem.memstatus))
```

出力は、resourcesラベルが複数の基盤DMEパスを表していることを示しています。

sys/procおよびsys/procsysパスはポーリングクエリを使用し、プロセスおよびシステムリソース情報を提供します。sys/procsys/systememパスは、監視対象のメモリ状態が更新されて「OK」でなくなった場合に変更を報告できるイベントクエリを使用します。

これは、個々のDME識別名(DN)を直接指定することと、事前定義されたテレメトリパスラベルを使用することの違いを示しています。

以下に、いくつかの例を示します。

```
Individual DME path:
path sys/intf/phys-[eth1/10]
```

Predefined path label:
path resources

システムリソース遠隔測定の確認

show telemetry control databaseを使用して、センサーグループ4とサブスクリプション4の状態を確認します。

Sensor Group Databaseは次のように報告します。

Sensor Group ID	Sensor Group type	Sampling interval(ms)	Linked subscriptions	SubID
4	Timer /DME	10000/Running	1	4

タイマー/DMEタイプと10000/Runningサンプリング間隔により、センサーグループ4が定期的なDMEテレメトリソースとして動作していることが確認されます。

Sensor Path Databaseには、リソースラベルに関連付けられた基盤となるパスも表示されます。

以下に、いくつかの例を示します。

```
resources:sys/procsys(4)
GPB Encoded Data size in bytes (Cur/Min/Max): 20221/20221/20229
Subscription Id: 4
Message Stats:
Sent = 14
Error = 0
Drops = 0
```

プロセス関連のパスでは、アクティブなテレメトリ収集も報告されます。

```
resources:sys/proc(4)

GPB Encoded Data size in bytes (Cur/Min/Max): 82284/81262/85098
Subscription Id: 4

Message Stats:

Sent      = 14
Error     = 0
Drops     = 0
```

これらのカウンタは、リソース情報が収集され、GPBとしてエンコードされ、メッセージエラーやドロップなしで送信されていることを確認します。

NX-OSのシステムリソースとの比較

従来のNX-OS CLIを使用して、スイッチの現在のCPUとメモリの状態を表示できます。

```
N9K-TELEMETRY-SW1# show system resources
Load average:  1 minute: 0.57   5 minutes: 0.58   15 minutes: 0.63
Processes   :   854 total, 2 running
CPU states  :   14.64% user,    3.50% kernel,   81.85% idle
<snip>
Memory usage: 24530808K total,   9315248K used,   15215560K free
Kernel buffers:  22104K Used
Kernel cached :  6255520K Used
Current memory status: OK
```

CLIは、システムリソースの即時ビューを提供します。

ストリーミングテレメトリを使用すると、同じタイプのCPUおよびメモリ情報を外部受信者にエクスポートできるため、複数のサンプルを時間をかけて保存および分析できます。

CPU使用率は急速に変化する可能性があります。したがって、サンプルが収集される時間が異なると、CLIに表示されるCPU値とテレメトリを通じて受信される値が異なる場合があります。

受信側でのリソースデータの確認

テレメトリの受信者は、サブスクリプション4に関連付けられたシステムリソース情報を正常にデコードしました。

次の例は、1つのテレメトリサンプルからのCPUとメモリの情報を示しています。

```
{
  "timestamp": "2026-09-18T23:15:08Z",
  "source": "N9K-TELEMETRY-SW1",
  "cpu": {
    "user": 11,
    "kernel": 2,
    "idle": 85,
    "averageLast60Seconds": 7.099999904632568
  },
}
```

```

"memory": {
  "status": "OK",
  "utilization": 38.18336486816406,
  "usedKB": 9366688,
  "freeKB": 15164120,
  "totalKB": 24530808
}
}

```

受信側の出力は、サブスクリプション4のCPUおよびメモリ情報が正常にデコードされ、外部モニタリングに使用できることを示しています。

比較のため、NX-OS CLIでは約24.5 GBの総メモリのうち約9.3 GBの使用済みメモリが報告され、現在のメモリステータスはOKになっています。テレメトリサンプルは、同じ合計メモリ値、約38 %のメモリ使用率、および同じOKメモリ状態を報告します。

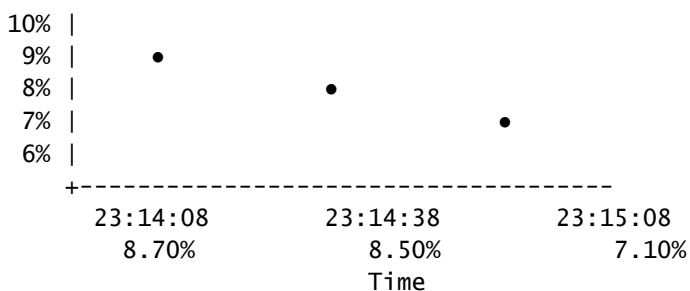
CPU使用率は動的に変化し、測定は必ずしも同じ瞬間に収集されないため、CLIとテレメトリサンプルの間でCPU値が変化する可能性があります。

時間の経過に伴うリソース使用率の監視

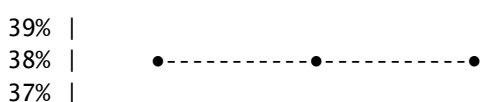
複数のテレメトリサンプルを使用して、時間の経過とともに変化するシステムリソースの使用状況を確認できます。

次のサンプルは、ラボ演習の間にサブスクリプション4から受け取りました。

CPU平均使用率 – 過去60秒



メモリ使用率



+-----+-----+-----+		
23:14:08	23:14:38	23:15:08
37.82%	38.06%	38.18%
Time		

これらのサンプルは、定期的なテレメトリが、1回の瞬間的な測定ではなく、時間ベースのシステム動作のビューを提供できることを示しています。実稼働環境では、監視プラットフォームまたは監視プラットフォームによって大量のサンプルを保存し、そのサンプルを使用して傾向の特定、アラートの生成、履歴ダッシュボードの構築を行うことができます。

結論

Cisco NX-OSストリーミングテレメトリは、Cisco Nexus 9000スイッチから外部テレメトリ受信機に動作情報をエクスポートするための、構造化されたメカニズムを提供します。

このドキュメントでは、DMEセンサーパス、GPBエンコーディング、gRPCトランスポート、宛先グループ、センサーグループ、サブスクリプションなど、ストリーミングテレメトリの基本的なコンポーネントについて説明しました。

ラボ環境を使用して、定期的なテレメトリとイベントベースのテレメトリの両方を設定および検証しました。定期的なサブスクリプションを使用してEthernet1/10の統計情報と運用情報を収集し、イベントベースのサブスクリプションを使用してLoopback100管理対象オブジェクトに対する制御された変更を検出しました。

また、実際のシステムリソースのモニタリング例を使用して、事前定義されたリソースパスラベルを使用して、外部テレメトリ受信機にCPUおよびメモリ情報をエクスポートする方法を示しました。

検証およびトラブルシューティングの例では、NX-OSテレメトリコマンドを使用して、トランスポート接続、データ収集、イベント処理、および履歴収集の失敗を検証する方法を示しました。

これらの概念は、Cisco Nexus 9000 NX-OS上の基本的なストリーミングテレメトリ展開を理解し、実装し、検証し、トラブルシューティングするための基盤となります。

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。