



## コレクタからルータへのモデル駆動型テレメトリセッションの確立

テレメトリのストリーミングは、ネットワークヘルスをモニタするための新しいパラダイムです。関心のある設定データおよび運用データを Cisco IOS XR ルータから効率的にストリーミングするメカニズムを提供します。このストリーミングされるデータは、モニタリングおよびトラブルシューティングのため、構造化された形式でリモート管理ステーションに送信されます。

テレメトリ データを使用して、データ レイクを作成します。このデータを分析することにより、ネットワークのプロアクティブなモニタリング、CPU およびメモリの使用率のモニタリング、パターンの特定、予測的な方法でのネットワークのトラブルシューティングを行い、自動化を使用した復元力のあるネットワークを作成するための戦略を考案します。

テレメトリは、関心のあるデータを **センサーパス** の形でサブスクライブする **サブスクリプション** モデルで機能します。センサーパスは、**OpenConfig データ モデル** またはネイティブのシスコデータ モデルを記述します。テレメトリのための **OpenConfig データ モデル** および **ネイティブ データ モデル** には、バージョン管理のためのホスティング サービスを提供するソフトウェア開発プラットフォームである **GitHub** からアクセスできます。ルータと受信者の間にテレメトリセッションを確立することによって、どのユーザがサブスクリプションを開始するかを選択します。セッションは、**ダイヤルアウト モード** または **ダイヤルイン モード** のいずれかを使用して確立されます。これについては、「**テレメトリを使用したネットワークモニタリング戦略の拡張**」の記事を参照してください。



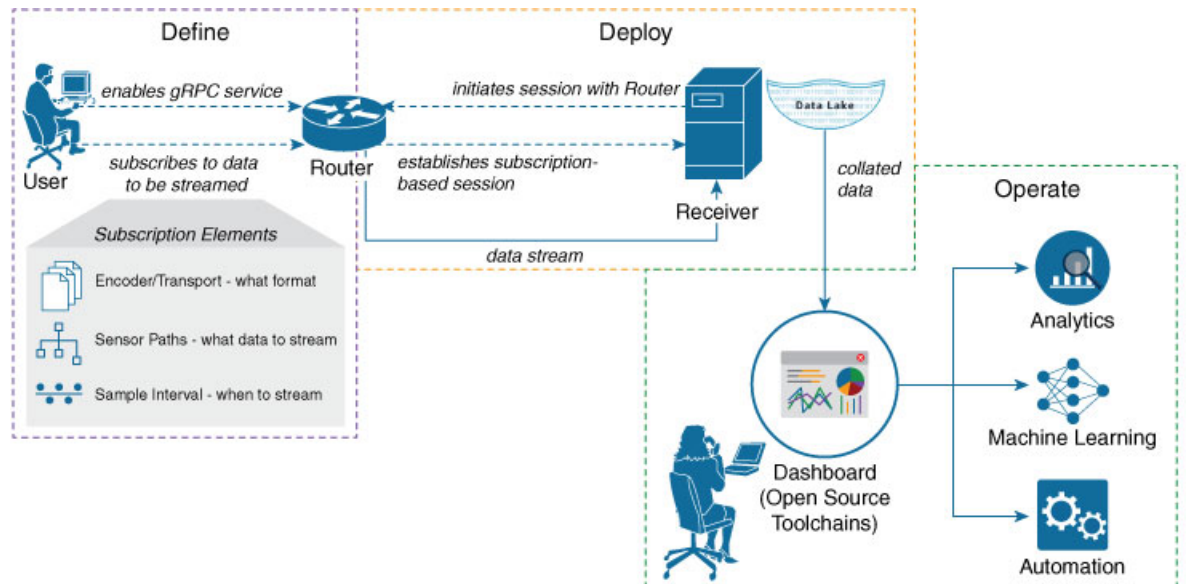
(注) モデル駆動型テレメトリを使用したリアルタイムでのネットワーク管理がもたらす力については、こちらの **ビデオ** をご覧ください。

この記事では、受信者がルータにダイヤルインしてテレメトリセッションを確立するダイヤルインモードについて説明します。このモードでは、受信者がルータにダイヤルインし、サブスクリプションで指定された 1 つ以上のセンサーパスに動的にサブスクライブします。ルータは、受信者が確立したセッションと同じセッションを介してテレメトリデータをストリーミングします。サブスクリプションのダイヤルインモードは動的です。この動的サブスクリプション

ンは、受信者がサブスクリプションをキャンセルしたとき、またはセッションが終了したときに終了します。

次の図は、ダイヤルイン モードの概要を示しています。

図 1:ダイヤルイン モード



この記事では、ネットワーク内のさまざまなパラメータの同時モニタリングを示す使用例を使用して、ネットワークの可視性の向上およびネットワークを安定させるための情報を得たうえでの意思決定にテレメトリ データのストリーミングがどのように役立つかを説明します。

- [プロアクティブな分析のためのテレメトリ データを使用してネットワーク パラメータをモニタする \(2 ページ\)](#)

## プロアクティブな分析のためのテレメトリデータを使用してネットワーク パラメータをモニタする

この使用例では、[ダイヤルインモード](#)で、テレメトリ データを使用してネットワークに関するさまざまなパラメータをストリーミングする方法について説明します。このデータは、パターンをモニタして問題をプロアクティブにトラブルシューティングする予測分析に使用します。この使用例では、オープンソースの収集スタックにある、テレメトリデータの保存および分析に使用するツールについて説明します。



- (注) データモデル、オープンソースのコレクタ、エンコーディングを利用し、それらをモニタリングツールに統合するためにモデル駆動型テレメトリを設定する方法については、こちらの[ビデオ](#)をご覧ください。

テレメトリには、次のワークフローがあります。

- **定義**：ルータから受信者にデータをストリーミングするためのサブスクリプションを定義します。サブスクリプションを定義するには、センサーグループを作成します。
- **展開**：受信者は、ルータとのセッションを開始し、サブスクリプションベースのテレメトリセッションを確立します。ルータは、受信者にデータをストリーミングします。ルータでサブスクリプションの展開を確認します。
- **運用**：オープンソース ツールを使用してテレメトリ データを消費および分析し、分析に基づいて必要なアクションを実行します。

### 始める前に

次の依存関係を満たしていることを確認します。

- ルータと受信者の間に L3 接続があることを確認します。
- ルータの gRPC サーバが受信者からの着信接続を受け入れるようにします。

```
Router#configure
Router (config)#grpc
Router (config-grpc)#port <port-number>
Router (config-grpc)#commit
```

ポート番号の範囲は 57344 ~ 57999 です。ポート番号が使用できない場合は、エラーが表示されます。

次の例は、ルータで TLS が有効になっている場合の gRPC 設定の出力を示します。

```
Router#show grpc
Address family : ipv4
Port : 57300
VRF : global-vrf
TLS : enabled
TLS mutual : disabled
Trustpoint : none
Maximum requests : 128
Maximum requests per user : 10
Maximum streams : 32
Maximum streams per user : 32
TLS cipher suites
Default : none
Enable : none
Disable : none
Operational enable : ecdhe-rsa-chacha20-poly1305
: ecdhe-ecdsa-chacha20-poly1305
: ecdhe-rsa-aes128-gcm-sha256
: ecdhe-ecdsa-aes128-gcm-sha256
: ecdhe-rsa-aes128-sha
Operational disable : none
```

## ルータから受信者にデータをストリーミングするためのサブスクリプションを定義する

サブスクリプションを作成し、ルータから宛先にストリーミングする対象データを定義します。

### 手順

- ステップ1** センサーパスを使用して、ルータからストリーミングするデータのサブセットを指定します。**センサーパス**は、YANG データモデルの階層内のパスを表します。この例では、ネイティブデータモデル `Cisco-IOS-XR-um-telemetry-model-driven-cfg.yang` を使用しています。センサーパスを含むセンサーグループを作成します。

例：

```

sensor-group health
  sensor-path Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
  sensor-path Cisco-IOS-XR-nto-misc-oper:memory-summary/nodes/node/summary
  sensor-path Cisco-IOS-XR-shellutil-oper:system-time/uptime
  !
sensor-group interfaces
  sensor-path
Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters

  sensor-path Cisco-IOS-XR-pfi-im-cmd-oper:interfaces/interface-summary
  !
sensor-group optics
  sensor-path
Cisco-IOS-XR-controller-optics-oper:optics-oper/optics-ports/optics-port/optics-info
  !
sensor-group routing
  sensor-path
Cisco-IOS-XR-clns-isis-oper:isis/instances/instance/levels/level/adjacencies/adjacency
  sensor-path Cisco-IOS-XR-clns-isis-oper:isis/instances/instance/statistics-global
  sensor-path
Cisco-IOS-XR-ip-rib-ipv4-oper:rib/vrfs/vrf/afs/af/safs/saf/ip-rib-route-table-riates/ip-rib-route-table-riate/protocol/isis/as/information

  sensor-path
Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/instance-active/default-vrf/process-info

  !
sensor-group mpls-te
  sensor-path Cisco-IOS-XR-mpls-te-oper:mpls-te/tunnels/summary
  sensor-path Cisco-IOS-XR-ip-rsvp-oper:rsvp/interface-briefs/interface-brief
  sensor-path Cisco-IOS-XR-ip-rsvp-oper:rsvp/counters/interface-messages/interface-message

  !

```

- ステップ2** ルータからストリーミングされるテレメトリ データをサブスクライブします。**サブスクリプション**は、センサーグループをバインドし、ストリーミング方式を設定します。ストリーミング方式は、**パターン駆動型**または**イベント駆動型**にすることができます。センサーパスを異なるサブスクリプションに分割すると、ルータの効率が向上し、運用データを大規模に取得できるようになります。

例：

(注) イベント駆動型テレメトリの設定は、サンプル間隔が異なる点を除き、パターン駆動型テレメトリに似ています。サンプル間隔の値を 0 (ゼロ) に設定すると、イベント駆動型テレメトリのサブスクリプションが設定され、間隔をゼロ以外の値に設定すると、パターン駆動型テレメトリのサブスクリプションが設定されます。

```
subscription health
  sensor-group-id health strict-timer
  sensor-group-id health sample-interval 30000
!
subscription interfaces
  sensor-group-id interfaces strict-timer
  sensor-group-id interfaces sample-interval 30000
!
subscription optics
  sensor-group-id optics strict-timer
  sensor-group-id optics sample-interval 30000
!
subscription routing
  sensor-group-id routing strict-timer
  sensor-group-id routing sample-interval 30000
!
subscription mpls-te
  sensor-group-id mpls-te strict-timer
  sensor-group-id mpls-te sample-interval 30000
!
```

---

## サブスクリプションの展開を確認する

受信者は、受信者にダイヤルインし、サブスクリプションに基づいて動的なセッションを確立します。セッションが確立されると、ルータはデータを受信者にストリーミングしてデータレイクを作成します。

サブスクリプションの展開は、ルータ上で確認できます。

手順

---

サブスクリプションの状態を確認します。Active状態は、ルータがサブスクリプションに基づいて受信者にデータをストリーミングする準備が整っていることを示します。

例：

```
Router#show telemetry model-driven subscription
```

ルータは、サブスクリプションベースのテレメトリセッションを使用して受信者にデータをストリーミングし、受信側にデータレイクを作成します。

---

## ネットワークの詳細な分析のためにテレメトリ データを操作する

データ レイクからのテレメトリ データの消費と分析を開始するには、オープンソースの収集スタックを使用できます。この使用例では、収集スタックの次のツールを使用します。

- **Pipeline** は、データを収集するために使用される軽量ツールです。[Network Telemetry Pipeline](#) は、[Github](#) からダウンロードできます。pipeline.conf ファイルを使用して、コレクタがルータと通信する方法と処理されたデータを送信する場所を定義します。
- **Telegraph** または **InfluxDB** は、可視化ツールによって取得されるテレメトリ データを保存する時系列データベース (TSDB) です。[InfluxDB](#) は、[Github](#) からダウンロードできます。metrics.json ファイルを使用して、TSDB に含めるデータを定義します。
- **Grafana** は、ルータからストリーミングされたデータのグラフおよびカウンタを表示する可視化ツールです。

つまり、Pipeline は TCP および gRPC テレメトリ ストリームを受け入れてデータを変換し、そのデータを InfluxDB データベースにプッシュします。Grafana は、InfluxDB データベースからのデータを使用してダッシュボードおよびグラフを作成します。Pipeline と InfluxDB は、同じサーバ上でも異なるサーバ上でも実行できます。

ルータを次のパラメータでモニタすることを検討してください。

- メモリおよび CPU 使用率
- インターフェイス カウンタおよびインターフェイス サマリー
- 光コントローラからのトランスミッタおよびレシーバの電力レベル
- ISIS ルートカウントおよび ISIS インターフェイス
- BGP ネイバー、パス カウント、およびプレフィックス カウント
- MPLS-TE トンネルの概要
- 各インターフェイスでの RSVP 制御メッセージと帯域幅の割り当て

### 手順

**ステップ 1** Pipeline を開始し、ルータのクレデンシャルを入力します。

例 :

```
$ bin/pipeline -config pipeline.conf

Startup pipeline
Load config from [pipeline.conf], logging in [pipeline.log]

CRYPT Client [grpc_in_mydmtrouter], [http://172.0.0.0:5432]
Enter username: <username>
Enter password: <password>
Wait for ^C to shutdown
```

ストリーミングされたテレメトリ データは、InfluxDB に保存されます。

**ステップ 2** Grafana を使用して、ダッシュボードを作成し、ストリーミングされたデータを可視化します。

図 2: テレメトリ データを使用したネットワーク ヘルス の視覚的分析

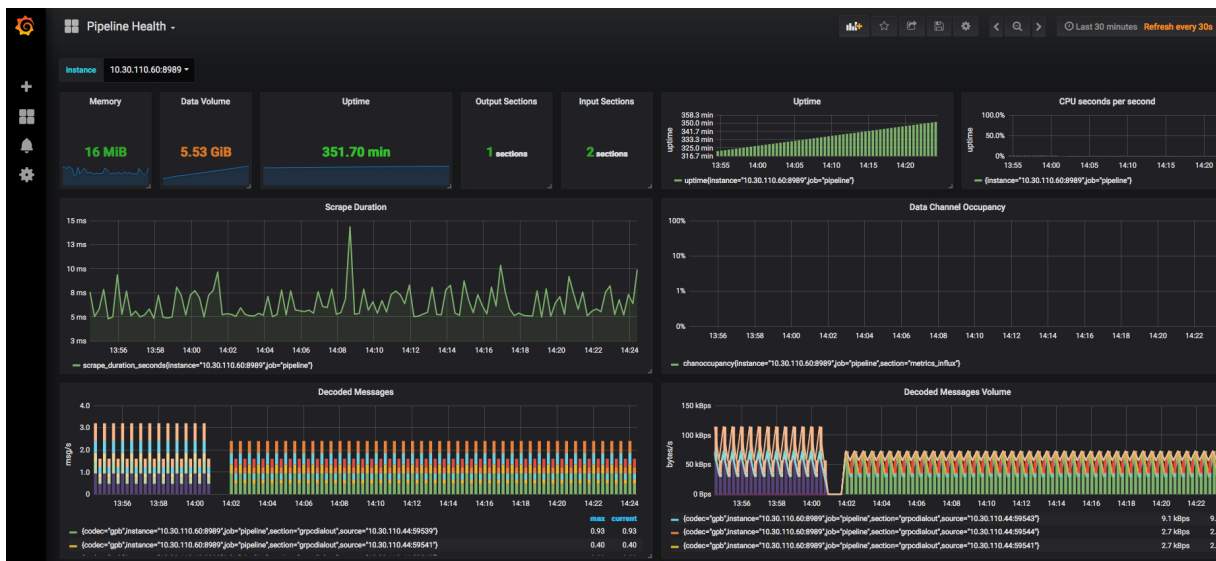


図 3: テレメトリ データを使用したシステム モニタリング の視覚的分析



結論として、テレメトリデータは、ネットワークのさまざまなパラメータを同時にモニタできることを示しています。このデータは、ネットワークのパフォーマンスに影響を与えることな

■ ネットワークの詳細な分析のためにテレメトリ データを操作する

く、ほぼリアルタイムでストリーミングされます。このデータを使用すると、ネットワークの可視性が向上します。

---