

# Livre blanc sur la télémétrie pilotée par le modèle ASR9K

## Table des matières

[Introduction](#)

[Public](#)

[Brève introduction à la télémétrie](#)

[Pourquoi la télémétrie](#)

[Nécessité de s'éloigner du protocole SNMP](#)

[Avantages de la télémétrie en continu](#)

[Spécifications techniques de télémétrie pilotée par modèle](#)

[Fonctions de télémétrie](#)

[Composants de télémétrie](#)

[YANG](#)

[Codage](#)

[Transport](#)

[Télémétrie basée sur la cadence et télémétrie basée sur les événements](#)

[Consignes de conception télémétrique](#)

[Comment sélectionner un schéma de codage](#)

[Considérations relatives à la conception du réseau de transport](#)

[Évaluation des options de configuration télémétrique](#)

[Exemples de configuration de télémétrie](#)

[IOS-XR](#)

[Dissociation de la configuration des appels sortants](#)

[Définir des groupes de capteurs](#)

[Définir des groupes de destinations](#)

[Définir l'abonnement](#)

[Exemple de configuration complète](#)

[Avantages de l'accès commuté](#)

[Dissociation de la configuration d'accès entrant](#)

[Activer gRPC](#)

[Définir des groupes de capteurs](#)

[Définir l'abonnement](#)

[Modèle et exemple de configuration complète](#)

[Avantages de l'accès commuté](#)

[télémétrie événementielle](#)

[Configuration de télémétrie pilotée par événements](#)

[Modèle de configuration complet et exemple pour DIAL-OUT](#)

[Modèle de configuration complet et exemple pour l'ACCÈS COMMUTÉ](#)

[Validation de la télémétrie avec les commandes SHOW](#)

[Pile de collecte télémétrique](#)

[Considérations relatives au déploiement de la télémétrie dans un réseau](#)

[Évolutivité](#)

[Diffuser uniquement les données requises](#)

[Tenez compte de la quantité de données en continu](#)

[Références](#)

## Introduction

### Public

Ce livre blanc a pour but d'aider les clients à comprendre rapidement la fonctionnalité de télémétrie pilotée par modèle (MDT) en général et la façon dont elle a été mise en oeuvre dans Aggregation Services Router 9000 (ASR9K), y compris certaines directives de conception et des détails de configuration. Elle inclut également des considérations relatives au déploiement, qui seront utiles lors du déploiement de cette fonctionnalité à l'aide de ASR9K. Dans l'ensemble, ce livre blanc peut être un guide de référence rapide pour toute personne travaillant sur cette fonctionnalité.

Bien que la télémétrie soit présentée comme une fonctionnalité générique, l'accent est mis sur l'implémentation ASR9K ; en d'autres termes, toutes les fonctionnalités prises en charge par d'autres plates-formes cisco ne sont pas prises en charge par la plate-forme ASR9K et certaines implémentations peuvent être spécifiques à ASR9K.

## Brève introduction à la télémétrie

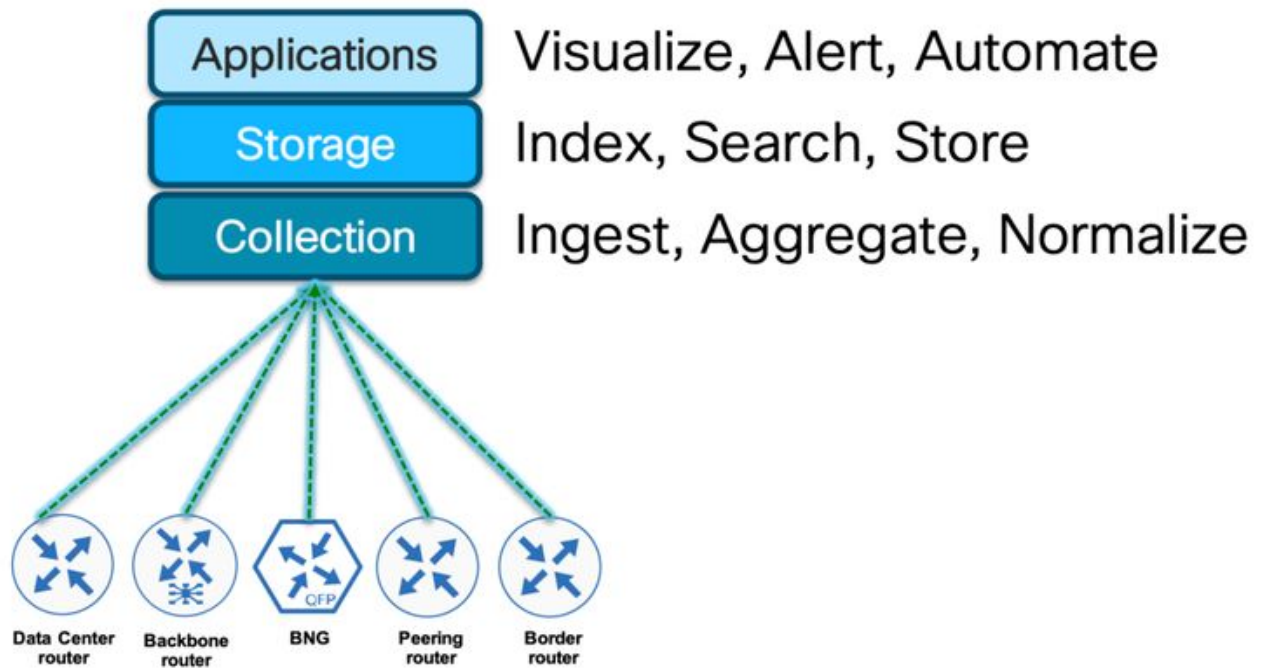
Pour commencer, en termes simples, la télémétrie est le processus de collecte de **données opérationnelles utiles**. Selon Wikipédia, la télémétrie est un processus de communication automatisé par lequel des mesures et d'autres données sont collectées à distance ou en des points inaccessibles et transmises à l'équipement récepteur pour surveillance. Le mot de télémétrie lui-même est dérivé des racines grecques : tele = distant, et metron = mesure.

Pour la gestion du réseau, les opérateurs de réseau s'appuient depuis longtemps sur le protocole SNMP (Simple Network Management Protocol). Bien que le protocole SNMP soit largement utilisé pour la surveillance du réseau, il n'a jamais été utilisé pour la configuration, même si la fonctionnalité de configuration avec SNMP a toujours été présente. Les opérateurs ont écrit des scripts d'automatisation pour gérer les tâches de configuration quotidiennes, mais les scripts sont difficiles à gérer pour de telles tâches.

Les opérateurs ont donc opté pour une gestion basée sur des modèles de données. La configuration du réseau est basée sur des modèles de données YANG poussés par des protocoles comme netconf par exemple. Désormais, le simple fait de pousser la configuration ne signifie pas que le service configuré est en cours d'exécution. Il doit exister un mécanisme capable de surveiller les données opérationnelles des services en même temps que la configuration. C'est là que les modèles de données utiles, que la télémétrie utilise pour extraire les informations du périphérique, sont utiles. Par conséquent, la configuration est basée sur le modèle de données YANG, donc doit être la vérification du service aussi bien ; comme dans le cas de la télémétrie, afin d'avoir la même sémantique d'objet. Par conséquent, le terme est appelé **télémétrie pilotée par modèle** ou télémétrie en continu.

La télémétrie pilotée par le modèle (MDT) a été introduite dans **cXR** (IOS XR 32 bits) depuis la version **6.1.1** et permet la collecte et les mesures de données critiques en temps quasi réel,

fournissant une réponse rapide à la plupart des problèmes opérationnels du réseau moderne.



Architecture de télémétrie de haut niveau

MDT tire parti des modèles de données structurées pris en charge par le périphérique réseau et fournit les données critiques définies dans ces modèles de données. **La télémétrie** aide les clients à **gérer leur réseau multifournisseur** à l'aide d'un système, d'un processus et d'applications de gestion de réseau communs, car les données collectées à partir du réseau sont basées sur des normes et sont uniformes dans l'ensemble de la mise en oeuvre du fournisseur.

Plutôt que d'attendre la récupération (extraction) des données à partir d'une station de gestion centralisée (généralement SNMP NMS), avec MDT, les périphériques réseau envoient de manière proactive (**push**) des données de performances relatives aux fonctions vitales du réseau, telles que les informations de transfert de paquets, les statistiques d'erreur, l'état du système, les ressources processeur et mémoire, etc.

## Pourquoi la télémétrie

La collecte de données à des fins d'analyse et de dépannage a toujours été un aspect important de la surveillance de l'état d'un réseau. Plusieurs mécanismes sont disponibles, tels que SNMP, CLI et Syslog, pour collecter des données sur un réseau. Alors que ces méthodes ont longtemps servi le réseau, mais ne conviennent pas au réseau moderne où la demande d'automatisation est essentielle, les services à grande échelle sont fondamentaux. Les informations sur l'état du réseau, les statistiques de trafic et les informations sur l'infrastructure critique sont envoyées à une station distante dans NMS, où elles sont utilisées pour améliorer les performances opérationnelles et réduire le temps de dépannage. Un modèle d'extraction tel que snmp dans lequel un client interroge tous les noeuds du réseau n'est pas efficace. La charge de traitement sur les noeuds du réseau augmente lorsqu'il n'y a plus de clients à interroger. Au contraire, un modèle push a la capacité de diffuser en continu des données hors du réseau et d'avertir le client. La télémétrie active le modèle push, qui fournit un accès en temps quasi réel aux données de surveillance.

La télémétrie en continu fournit un mécanisme permettant de sélectionner les données d'intérêt à partir des routeurs et de les transmettre dans un format standard à des stations de gestion distantes pour surveillance. Ce mécanisme permet un réglage précis du réseau sur la base de

données en temps réel, ce qui est essentiel pour son fonctionnement transparent. La finesse de la granularité et la fréquence plus élevée des données disponibles par télémétrie permettent une meilleure surveillance des performances et donc un meilleur dépannage.

Elle contribue à une utilisation plus efficace de la bande passante au niveau du réseau, de l'utilisation des liaisons, de l'évaluation des risques et de l'évolutivité. Grâce à la télémétrie en continu, les opérateurs de réseau disposent de données en temps quasi réel, ce qui contribue à améliorer la prise de décision.

## Nécessité de s'éloigner du protocole SNMP

Le protocole SNMP existe depuis trois décennies et son mode de fonctionnement n'a pas changé pour répondre aux besoins de surveillance des réseaux modernes. Le véritable problème est la vitesse d'exécution du protocole SNMP.

Les trois principaux défis posés par le protocole SNMP font en fait partie de son comportement opérationnel fondamental. Par conséquent, le protocole SNMP offre peu ou pas de place pour l'amélioration et la télémétrie permet de résoudre ces trois problèmes de manière inhérente.

- **Vitesse d'exécution et nécessité d'une surveillance en temps réel**

SNMP utilise les opérations PULL Model - GetBulk / GetNext qui fonctionnent de façon linéaire en parcourant les tables d'une colonne à une autre jusqu'à. En outre, plusieurs requêtes sont requises dans le cas de tables volumineuses qui ne peuvent pas tenir dans un seul paquet. Il s'agit du plus grand goulot d'étranglement qui entraîne le ralentissement du protocole SNMP et les données envoyées sont souvent dépassées d'un certain facteur de temps en minutes. Ce délai n'est tout simplement pas acceptable pour les exigences de surveillance du réseau moderne.

MDT (Model Driven Telemetry) utilise le modèle PUSH et est intrinsèquement libre des limitations mentionnées ci-dessus car il sait quelles données doivent être envoyées à qui et à quel intervalle. Il ne nécessite qu'une seule recherche pour collecter les données et utilise des modèles internes prédéfinis pour accélérer les opérations internes, ce qui permet de fournir beaucoup plus de données en un temps considérablement réduit.

- **Frais généraux supplémentaires et manque d'options d'optimisation**

Les données extraites par SNMP sont en fait stockées en tant que structures de données internes et doivent être converties en interne par le noeud. Il s'agit d'un travail supplémentaire en arrière-plan, où le noeud réseau mappe les structures de données internes au format SNMP. Des optimisations internes ont été effectuées, mais elles ne sont pas encore suffisantes.

D'autre part, la télémétrie extrait directement les structures de données internes et effectue un traitement minimal avant d'envoyer ces données, fournissant ainsi les données les plus mises à jour avec le moins de temps et d'effort possible.

- **Nature linéaire de la charge de travail**

Chaque poste d'interrogation supplémentaire entraînera une charge de travail supplémentaire sur le noeud, même si nous interrogeons les mêmes données exactes au même moment. L'accès parallèle de la même MIB à partir de plusieurs stations d'interrogation peut ralentir la réponse et augmenter l'utilisation du CPU. Ceci est particulièrement évident dans le cas de grandes tables, où plusieurs stations accèdent à différentes parties de la même table MIB.

La télémétrie, quant à elle, doit extraire les données une seule fois et répliquer les paquets si les mêmes données sont requises par plusieurs destinations. Le modèle Push bat SNMP Pull pour la vitesse et l'évolutivité.

Avec le MDT, l'approche de la collecte de données change radicalement et ses principes fondamentaux sont listés dans le tableau ci-dessous et comparés aux points clés de la technologie SNMP.

#### Protocole de gestion de réseau simple (SNMP)

Informations non en temps réel

Peu évolutif

Modèle De Tirage

Non automatisé

#### Télémétrie pilotée par le modèle (MDT)

Informations en temps réel

Hautement évolutif

Modèle À Poussée

Prêt pour l'automatisation/piloté par modèle de données

## Avantages de la télémétrie en continu

Les données de télémétrie en temps réel diffusées en continu sont utiles pour :

**Planification de la capacité/optimisation du trafic** : lorsque l'utilisation de la bande passante et les pertes de paquets dans un réseau sont surveillées fréquemment, il est plus facile d'ajouter ou de supprimer des liaisons, de rediriger le trafic, de modifier la réglementation, etc. Grâce à des technologies telles que le réacheminement rapide, le réseau peut commuter vers un nouveau chemin et réacheminer plus rapidement que le mécanisme d'intervalle d'interrogation SNMP. La transmission en continu des données de télémétrie permet d'accélérer le temps de réponse et le trafic.

**Meilleure visibilité** : permet de détecter et d'éviter rapidement les situations de panne qui se produisent après un problème sur le réseau.

## Spécifications techniques de télémétrie pilotée par modèle

La section suivante traite des fonctions techniques et des principaux composants de la télémétrie pilotée par le modèle IOS XR, appelée MDT.

### Fonctions de télémétrie

Le cadre de télémétrie est organisé en trois blocs fonctionnels distincts et interconnectés.

Le premier bloc concerne la **représentation de données**, qui est la façon dont l'analyse ou les mesures de référence d'information est organisée à bord.

Le deuxième bloc concerne le **codage**. À chaque intervalle d'échantillonnage, la télémétrie traduit les données de mesure ci-dessus dans un format qui peut être sérialisé sur le fil. Bien entendu, le contrôleur à l'autre extrémité doit pouvoir décoder les données pour avoir une copie identique des données d'origine envoyées par le dispositif.

Le dernier bloc concerne le **transport**. Il s'agit de la pile de protocoles utilisée pour transférer des données entre les périphériques.

Le tableau suivant résume la structure principale des blocs de construction de la télémétrie pilotée

par modèle :

| Fonction                   | Composants                |
|----------------------------|---------------------------|
| Représentation des données | Modèles de données YANG   |
| Codage                     | GPB / GPB auto-descriptif |
| Transport                  | TCP/gRPC                  |

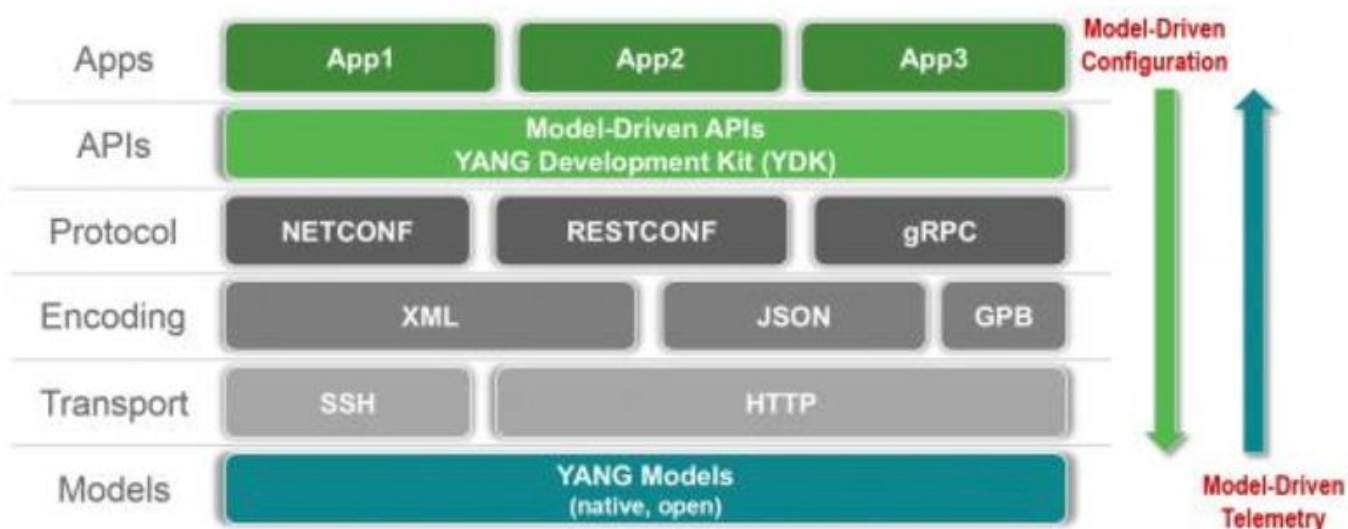
Tableau 3 Blocs de construction télémétriques

## Composants de télémétrie

Avant de comprendre le fonctionnement de la télémétrie et des éléments de configuration sous-jacents, il est important de comprendre les différents composants de la télémétrie afin d'évaluer une configuration optimale. La télémétrie repose sur la pile de programmabilité IOS XR, où une nouvelle infrastructure offre les fonctionnalités essentielles à l'automatisation du réseau.

YANG est récemment devenu une norme pour la modélisation de données, et elle est utilisée par la pile de programmabilité Cisco pour former un ensemble de données structurées qui peuvent être codées et transportées aussi rapidement que possible sur le réseau. La flexibilité de YANG donne le grand avantage d'être utilisé également comme outil de configuration pour les processus d'automatisation. Ces modèles de données sont associés à des formats de codage et des protocoles de transport spécifiques pour faire de MDT une solution complète pour Network Analytics.

Pour la configuration de la télémétrie pilotée par modèle, le modèle de données YANG devient un composant essentiel afin de permettre la transmission des données nécessaires à la collecte et à l'analyse.



Pile de programmabilité IOS XR

## YANG

Yang est défini comme le « langage de modélisation de données utilisé pour modéliser les données de configuration, les données d'état et les notifications pour les protocoles de gestion de réseau ». En raison de sa nature découplée d'une architecture de langage de programmation typique, YANG peut être mis en oeuvre pour interagir avec une grande variété d'outils.

La structure de données de modélisation YANG est construite autour du concept de modules et sous-modules qui définit une hiérarchie de données sous forme d'arbre, qui peut être utilisée pour plusieurs opérations, y compris les actions de configuration et la gestion des notifications.

Il existe plusieurs sources de modèles YANG disponibles, parmi lesquelles trois sont considérées comme principales :

- Modèles natifs / Spécifiques à Cisco
- OpenConfig
- IETF

**Modèles spécifiques à Cisco:** ces modèles sont également appelés **modèles natifs** et sont publiés par divers fournisseurs de périphériques, dont Cisco. par exemple, Cisco-IOS-XR-ptp-oper.yang

**Modèles OpenConfig :** OpenConfig est un groupe de travail informel composé d'opérateurs réseau. OpenConfig définit les modèles YANG courants que tous les fournisseurs doivent prendre en charge pour configurer les fonctions critiques. par exemple openconfig-interfaces.yang

**Modèles IETF:** l'IETF définit également quelques modules YANG courants qui décrivent la configuration de base des interfaces, la QoS et définissent d'autres types de données courants (comme IPv4, IPv6, etc.). par exemple, ietf-syslog-types.yang

Cisco prend en charge les modèles Openconfig disponibles. Les fournisseurs convergent vers une méthode normalisée de modélisation des données pour prendre en charge un environnement multifournisseur.

Il existe **trois types** de modèles Yang :

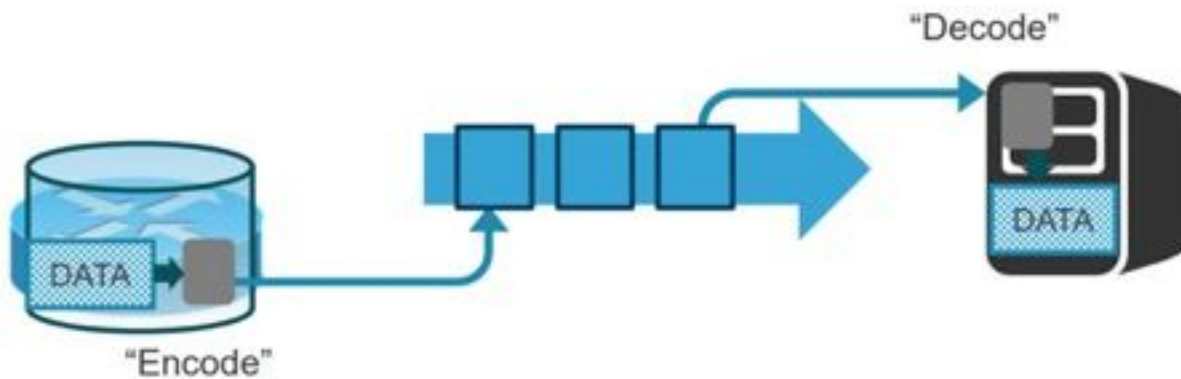
1. Opérationnel
2. Configuration
3. Action

La télémétrie ne s'intéresse qu'aux **modèles Yang opérationnels** qui peuvent être identifiés comme **\*-oper-\*.yang**.

YANG est défini dans la RFC 7950 : <https://tools.ietf.org/html/rfc7950>.

## Codage

Le codage (ou « sérialisation ») traduit les données (objets, état) dans un format qui peut être transmis sur le réseau. Lorsque le récepteur décode (« désérialise ») les données, il dispose d'une copie sémantiquement identique des données d'origine.



Au cours des premières étapes de développement de la télémétrie, XML était initialement considéré comme un format de codage de premier choix en raison de sa structure basée sur les balises. Cependant, le problème avec XML était sa structure de codage non compacte. Le protocole GPB (Google Protocol Buffers) a finalement été adopté par Cisco car il améliore l'efficacité et la vitesse des opérations de codage.

Il existe deux types de GPB comme options d'encodage pour la transmission en continu de télémétrie :

1. GPB compact
2. Auto-descripteur GPB

La principale différence entre les deux formats de télémétrie GPB est la manière dont ils représentent et codent les clés dans un flux de données de télémétrie.

GPB – "compact"

```
1: GigabitEthernet0/0/0/0
50: 449825
51: 41624083
52: 360333
53: 29699362
54: 91299
<snip>
```

GPB – "self-describing"

```
{InterfaceName:
GigabitEthernet0/0/0/0
GenericCounters {
PacketsSent: 449825
BytesSent: 41624083
PacketsReceived: 360333
BytesReceived: 29699362
MulticastPacketsReceived: 91299
<snip>
```

JSON est un autre schéma de codage convivial disponible qui est très facile à comprendre et presque toutes les applications seront en mesure de décoder.

Du point de vue du déploiement, un schéma de codage présente peu d'avantages et d'inconvénients. La comparaison des différents schémas de codage est donnée dans la section Directives de conception de télémétrie.

## Transport

La télémétrie offre trois choix possibles pour les protocoles de transport :

- TCP
- gRPC
- UDP



La télémétrie définit également deux modes d'initiation différents afin de démarrer une session entre le noeud et le collecteur :

- Accès commuté
- Accès à distance

La différence entre les deux modes réside uniquement dans la manière dont la session de transport est établie.

Pendant les sessions de numérotation, le périphérique initie la connexion en envoyant un paquet syn vers le port de serveur préconfiguré. Une fois la connexion établie, les flux de données sont immédiatement éloignés du périphérique.

Pour les sessions d'accès à distance, le routeur écoute passivement un port TCP en attente d'une connexion serveur.

Cependant, une fois la session établie, le routeur n'est pas interrogé par le serveur lui-même, car le périphérique est toujours responsable des opérations de transmission de données. En effet, dans le MDT, le concept d'interrogation des données n'existe même pas.

TCP est, par défaut, la méthode de transport prédéfinie pour la télémétrie, car il est fiable et très facile à configurer en option.

gRPC est un framework open source moderne conçu pour fonctionner dans n'importe quel environnement. Il est construit sur HTTP/2 et fournit un ensemble de fonctionnalités améliorées et riches.

## **Télémétrie basée sur la cadence et télémétrie basée sur les événements**

Les données de l'ensemble de données abonné sont diffusées vers la destination soit à un intervalle périodique configuré, soit uniquement lorsqu'un événement se produit. Ce comportement dépend de la configuration de MDT pour la télémétrie basée sur la cadence ou la télémétrie basée sur les événements.

La configuration de la télémétrie basée sur les événements est similaire à celle de la télémétrie basée sur la cadence, avec seulement l'intervalle d'échantillonnage comme différenciateur. La configuration de la valeur d'intervalle d'échantillonnage sur zéro définit l'abonnement pour la télémétrie basée sur les événements, tandis que la configuration de l'intervalle sur toute valeur non nulle définit l'abonnement pour la télémétrie basée sur la cadence.

Il est recommandé d'utiliser la télémétrie événementielle pour les événements liés au changement.

## **Consignes de conception télémétrique**

Comme expliqué, il existe de nombreux composants dans la pile de télémétrie, voici quelques directives à prendre en compte lors de la mise en oeuvre de la télémétrie sur des périphériques XR.

## Comment sélectionner un schéma de codage

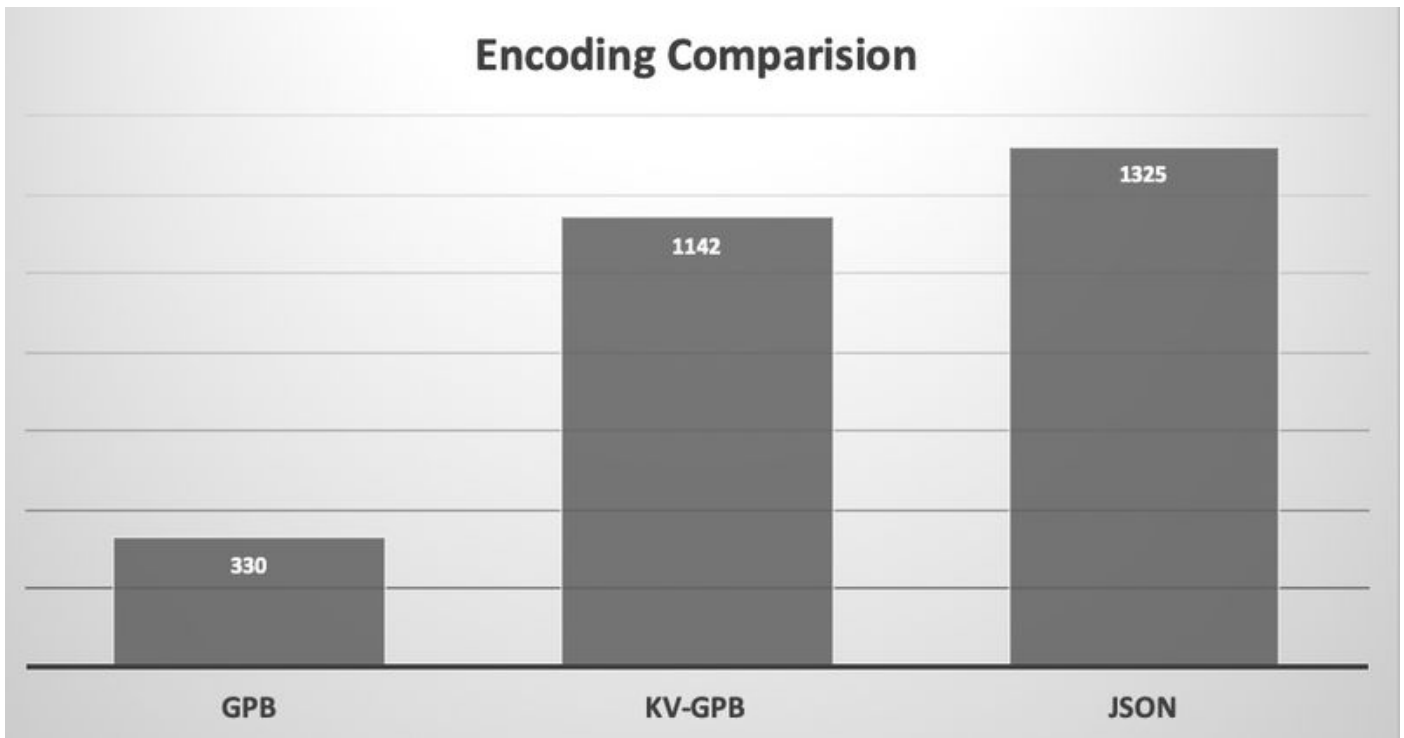
Comme indiqué, le codage ou la sérialisation convertit les données (objets, état) dans un format qui peut être transmis sur le réseau. Lorsque le récepteur décode ou désérialise les données, il dispose d'une copie sémantiquement identique des données d'origine.

Diverses options de codage varient en termes d'efficacité du câblage et de facilité d'utilisation.

| Codage                      | Brève description   | Efficacité sur fil | Autre contrepartie   |
|-----------------------------|---|--------------------|--|
| GPB (compact)               | Tout au format binaire (sauf les valeurs qui sont des chaînes)<br>2 fois plus rapide et plus complexe sur le plan opérationnel (mais sans rapport avec le protocole SNMP)<br>Clés de chaîne et valeurs binaires | Élevé              | Fichier de profil par modèle   |
| GPB - KV (paire clé-valeur) | (sauf les valeurs qui sont des chaînes)<br>3 Fois Plus Grand,<br>Modèles natifs : toujours besoin d'une heuristique pour les noms de clés   | Moyenne à faible   | Fichier .proto unique le décodage                                      |
| JSON                        | Tout en Chaînes : Clés et valeurs   | Faible             | Convivial. Lisible par l'homme, facile à analyser et facile à utiliser |

GPB-KV donne un bon point médian et équilibré pour le schéma de codage.

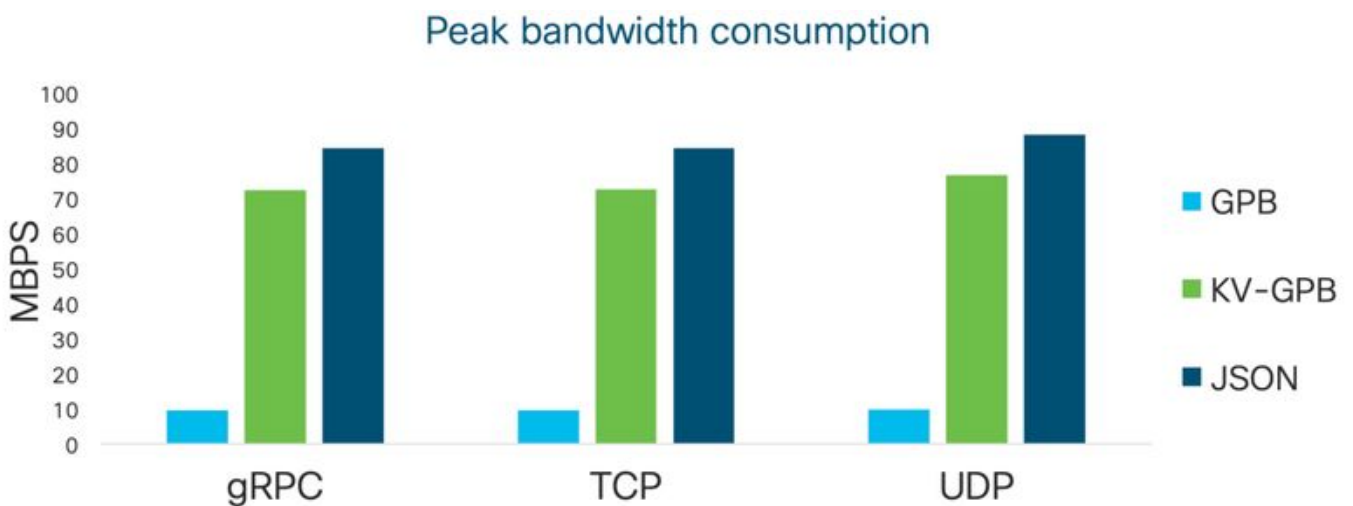
En ce qui concerne la longueur de message pour le schéma de codage choisi, la comparaison sur le fil est présentée ci-dessous.



Comparaison de codage - Longueur du message en octets

### Considérations relatives à la conception du réseau de transport

Différentes options de codage entraînent des besoins en bande passante différents. Lors de l'examen de la télémétrie, l'opérateur réseau doit s'assurer que la bande passante est suffisante, conformément au schéma de codage choisi. Juste pour avoir une idée juste, en dessous de la consommation de bande passante par comparaison de schéma de codage.



Comparaison de bande passante réseau

Cisco recommande d'utiliser KV-GPB. Il constitue un bon point de milieu entre l'efficacité et la commodité.

### Évaluation des options de configuration télémétrique

Lors de la configuration de la télémétrie pilotée par modèle, l'opérateur doit avoir une compréhension de tous les différents composants impliqués dans la télémétrie. En fonction des options disponibles pour le transport, l'encodage et la direction de la transmission en continu

comme détaillé ci-dessus, on peut choisir la combinaison qui convient le mieux à un environnement.

Les quatre composants clés sont :

1. Transport
2. Codage
3. Direction de session
4. Modèles de données YANG

**Transport** : Comme indiqué, le noeud peut transmettre des données de télémétrie via TCP, UDP ou gRPC sur HTTP/2.

Bien que le protocole TCP soit le choix privilégié pour la simplicité, gRPC offre une fonctionnalité TLS facultative qui peut être considérée comme un avantage supplémentaire du point de vue de la sécurité.

**Codage** : Le routeur peut fournir des données de télémétrie dans deux types différents de tampons de protocole Google : compact et GPB à auto-description.

Le format compact GPB est le codage le plus efficace, mais il nécessite un .proto unique pour chaque modèle YANG diffusé en continu. Le GPB autodéscriptif est moins efficace, mais il utilise un seul fichier .proto pour décoder tous les modèles YANG, car les clés sont passées sous forme de chaînes dans le fichier .proto.

**Direction de la session** : Il existe deux options pour lancer une session dans un déploiement de télémétrie. Le routeur peut passer des appels sortants vers le collecteur ou des appels entrants vers le routeur.

**YANG** est la norme reconnue pour la modélisation de données et la pile de programmabilité Cisco l'utilise également pour former un ensemble de données structurées qui peut être codé et transporté aussi rapidement que possible sur le réseau.

Ces modèles de données associés à des formats de codage et des protocoles de transport spécifiques, comme nous l'avons vu précédemment, font de la télémétrie pilotée par modèle (MDT) une solution complète pour l'analytique.

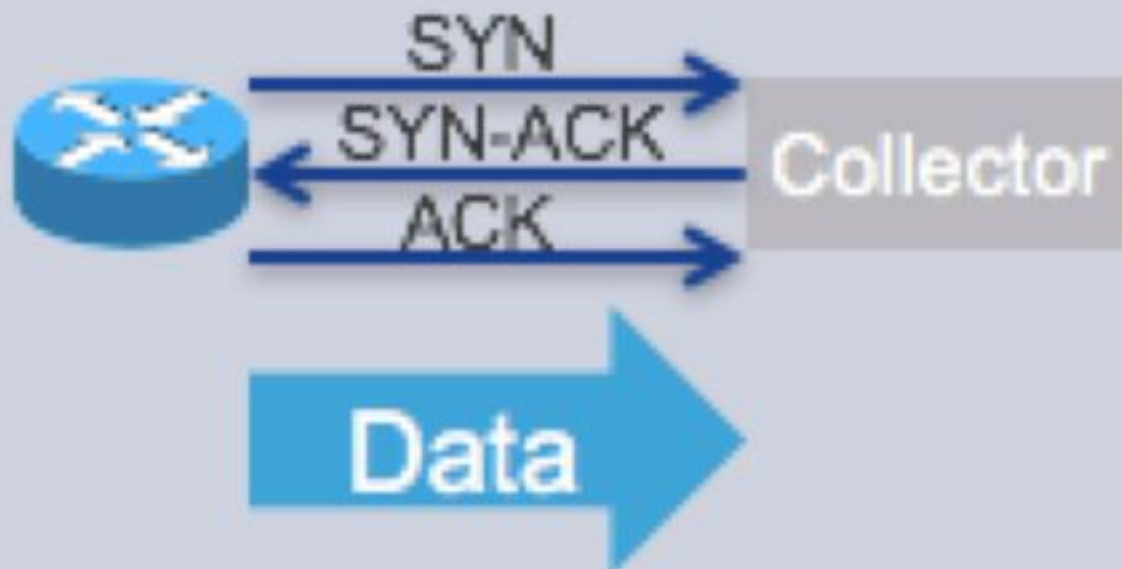
## Exemples de configuration de télémétrie

### IOS-XR

#### Dissociation de la configuration des appels sortants

En mode Dial-Out, le routeur est chargé de lancer une session TCP vers le collecteur et envoie les données spécifiées par le groupe de capteurs dans l'abonnement.

# Dial-Out



Acc

ès commuté à distance par télémétrie Du point de vue de la configuration, la configuration de la télémétrie est un processus en trois étapes. Tout d'abord, nous identifions les informations que nous voulons diffuser et les capturons sous la configuration Sensor Group. Deuxièmement, nous identifions la destination vers laquelle les informations doivent être diffusées en continu et les capturons dans la configuration du groupe de destinations. Troisièmement, nous utilisons les informations identifiées dans les deux étapes précédentes pour configurer l'abonnement réel.

1. Définir des groupes de capteurs
2. Définir des groupes de destinations
3. Définir un abonnement

## Définir des groupes de capteurs

La configuration du groupe Capteur identifie les informations qui doivent être diffusées en continu. Le modèle de configuration suivant fournit la configuration requise pour configurer les groupes de capteurs.

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#config
RP/0/RP0/CPU0:XR(config)#telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)#sensor-group <Sensor-Group-Name>
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# sensor-path <Sensor-Path>
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# commit
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# end
RP/0/RP0/CPU0:XR#
```

L'exemple suivant illustre un exemple réel de configuration du groupe de capteurs à partir de l'interface de ligne de commande du routeur :

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#config
RP/0/RP0/CPU0:XR(config)#telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)#sensor-group SensorGroup101
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# sensor-path Cisco-IOS-XR-infra-statsd-
oper:infra-statistics/interfaces/interface/latest/generic-counters
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# commit
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# end
RP/0/RP0/CPU0:XR#
```

Plusieurs chemins de capteur peuvent faire partie de la même définition de SensorGroup :

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#config
RP/0/RP0/CPU0:XR(config)#telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)#sensor-group SensorGroup101
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# sensor-path sensor-path Cisco-IOS-XR-infra-
statsd-oper:infra-statistics/interfaces/interface/data-rate
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# sensor-path Cisco-IOS-XR-infra-statsd-
oper:infra-statistics/interfaces/interface/latest/generic-counters
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# commit
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# end
RP/0/RP0/CPU0:XR#
```

## Définir des groupes de destinations

La configuration du groupe de destinations identifie la destination vers laquelle les informations doivent être diffusées en continu.

Il comporte trois paramètres clés

1. Direction de session
2. Codage à utiliser
3. Protocole de transport à utiliser

En voici un exemple :

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#config
RP/0/RP0/CPU0:XR(config)# telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)# destination-group DestGroup101
RP/0/RP0/CPU0:XR(config-model-driven-dest)# address family ipv4 10.1.1.1 port 5432
RP/0/RP0/CPU0:XR(config-model-driven-dest-addr)# encoding self-describing-gpb
RP/0/RP0/CPU0:XR(config-model-driven-dest-addr)# protocol tcp
RP/0/RP0/CPU0:XR(config-model-driven-dest-addr)# commit
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# end
RP/0/RP0/CPU0:XR#
```

## Définir l'abonnement

L'abonnement associe les informations Sensor-Group et Destination-Group comme élément final

de la configuration. L'intervalle d'échantillonnage est défini dans le cadre de l'abonnement.

En voici un exemple :

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#config
RP/0/RP0/CPU0:XR(config)telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)#subscription Subscription101
RP/0/RP0/CPU0:XR(config-model-driven-subs)#sensor-group-id SensorGroup101 sample-interval 30000
RP/0/RP0/CPU0:XR(config-model-driven-subs)#destination-id DestGroup101
RP/0/RP0/CPU0:XR(config-model-driven-subs)# commit
RP/0/RP0/CPU0:XR(config-model-driven-subs)# end
RP/0/RP0/CPU0:XR#
```

## Exemple de configuration complète

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#conf
RP/0/RP0/CPU0:XR(config)#
RP/0/RP0/CPU0:XR(config)#telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)#sensor-group SensorGroup101
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# sensor-path Cisco-IOS-XR-infra-statsd-
oper:infra-statistics/interfaces/interface/latest/generic-counters
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)#destination-group DestGroup101
RP/0/RP0/CPU0:XR(config-model-driven-dest)#address family ipv4 10.1.1.2 port 5432
RP/0/RP0/CPU0:XR(config-model-driven-dest-addr)#encoding self-describing-gpb
RP/0/RP0/CPU0:XR(config-model-driven-dest-addr)#protocol tcp
RP/0/RP0/CPU0:XR(config-model-driven-dest-addr)#subscription Subscription101
RP/0/RP0/CPU0:XR(config-model-driven-subs)#sensor-group-id SensorGroup101 sample-interval 30000
RP/0/RP0/CPU0:XR(config-model-driven-subs)#destination-id DestGroup101
RP/0/RP0/CPU0:XR(config-model-driven-subs)#commit
RP/0/RP0/CPU0:XR(config-model-driven-subs)#end
RP/0/RP0/CPU0:XR#
```

## Avantages de l'accès commuté

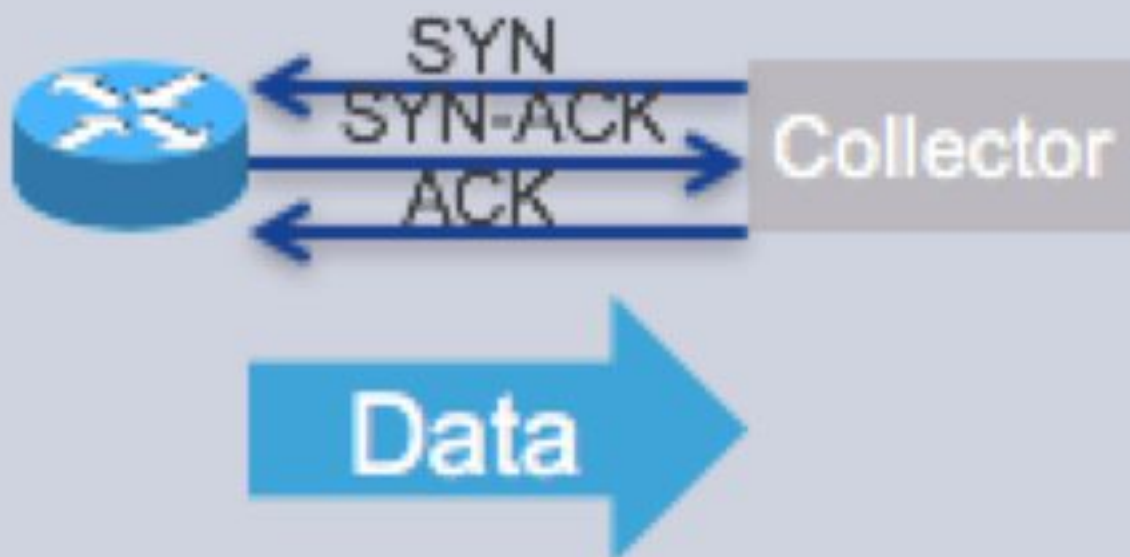
- Flexibilité accrue pour les options de transport.
- Pas besoin d'ouvrir des ports pour le trafic de gestion entrant.
- Anycast et équilibrage de charge.

## Dissociation de la configuration d'accès entrant

En mode Dial-In, un collecteur / récepteur / orchestrateur MDT compose un numéro sur le routeur et s'abonne dynamiquement à un ou plusieurs chemins de détection ou abonnements. Le routeur agit en tant que serveur et le client en tant que récepteur.

Une seule session est formée et le routeur transmet des données de télémétrie via la même session. Cet abonnement dynamique prend fin lorsque le destinataire annule l'abonnement ou lorsque la session prend fin.

# Dial-In



Acc

ès commuté par télémétrie

Comme le collecteur « compose » le numéro du routeur, il n'est pas nécessaire de spécifier chaque destination MDT dans la configuration. Il vous suffit d'activer le service gRPC sur le routeur, de connecter votre client et d'activer dynamiquement l'abonnement de télémétrie souhaité.

Du point de vue de la configuration, la configuration de la télémétrie est un processus en trois étapes similaire à celui décrit ci-dessus. Premièrement, nous devons activer gRPC. Deuxièmement, nous identifions la destination vers laquelle les informations doivent être diffusées en continu et les capturons dans la configuration du groupe de capteurs. Troisièmement, nous utilisons les informations identifiées dans les deux étapes précédentes pour configurer l'abonnement réel.

1. Activer gRPC
2. Définir des groupes de capteurs
3. Définir un abonnement

[Cliquez pour développer](#)

Le mode d'appel entrant est uniquement pris en charge avec gRPC

Le mode d'appel entrant est uniquement pris en charge avec gRPC

## Activer gRPC

Tout d'abord, nous devons activer le serveur gRPC sur le routeur pour accepter les connexions entrantes du collecteur.



## [Cliquez pour développer](#)

- La plage <port-number> est comprise entre 57344 et 57999. Si un numéro de port n'est pas disponible, une erreur s'affiche.

La plage <port-number> est comprise entre 57344 et 57999. Si un numéro de port n'est pas disponible, une erreur s'affiche.

```
RP/0/RP0/CPU0:XR#config
RP/0/RP0/CPU0:XR(config)# grpc
RP/0/RP0/CPU0:XR(config-grpc)#port 57890
RP/0/RP0/CPU0:XR(config-grpc)#commit
RP/0/RP0/CPU0:XR(config-grpc)#end
RP/0/RP0/CPU0:XR#
```

## Définir des groupes de capteurs

La configuration du groupe Capteur identifie les informations qui doivent être diffusées en continu. Le modèle de configuration suivant fournit la configuration requise pour configurer les groupes de capteurs.

L'exemple suivant illustre un exemple réel de configuration du groupe de capteurs dans l'interface de ligne de commande du routeur

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#config
RP/0/RP0/CPU0:XR(config)#telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)#sensor-group SensorGroup101
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# sensor-path openconfig-
interfaces:interfaces/interface
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# commit
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# end
RP/0/RP0/CPU0:XR#
```

## Définir l'abonnement

L'abonnement lie le groupe de capteurs et gRPC en tant que dernière partie de la configuration. L'intervalle d'échantillonnage est défini dans le cadre de l'abonnement.

Le modèle de configuration suivant fournit la configuration requise pour configurer l'abonnement.

L'exemple suivant illustre un exemple réel de l'interface de ligne de commande du routeur, dans lequel nous créons un abonnement et associons le groupe de capteurs et le groupe de destinations, puis définissons également le taux d'échantillonnage.

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#config
RP/0/RP0/CPU0:XR(config)telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)#subscription Subscription101
RP/0/RP0/CPU0:XR(config-model-driven-subs)#sensor-group-id SensorGroup101 sample-interval 30000
RP/0/RP0/CPU0:XR(config-model-driven-subs)# commit
RP/0/RP0/CPU0:XR(config-model-driven-subs)# end
RP/0/RP0/CPU0:XR#
```

## Modèle et exemple de configuration complète

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#config
RP/0/RP0/CPU0:XR(config)# grpc
RP/0/RP0/CPU0:XR(config-grpc)#port 57890
RP/0/RP0/CPU0:XR(config-grpc)telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)#subscription Subscription101
RP/0/RP0/CPU0:XR(config-model-driven-subs)#sensor-group-id SensorGroup101 sample-interval 30000
RP/0/RP0/CPU0:XR(config-model-driven-subs)# commit
RP/0/RP0/CPU0:XR(config-model-driven-subs)# end
RP/0/RP0/CPU0:XR#
```

## Avantages de l'accès commuté

- Un canal unique pour la configuration et la diffusion en continu
- Port d'écoute sur le routeur/périphérique
- Connexion Transitoire
- Actuellement, seul gRPC/gNMI est disponible

## téléométrie événementielle

Dans la téléométrie événementielle, les données de l'ensemble de données abonné sont diffusées en continu uniquement lorsqu'un événement se produit.

## Configuration de téléométrie pilotée par événements

La configuration de la téléométrie événementielle est similaire à la téléométrie basée sur la cadence, la seule différence de configuration de la téléométrie événementielle étant celle de l'intervalle échantillon. La configuration de la valeur d'intervalle d'échantillonnage sur zéro définit l'abonnement pour la téléométrie basée sur les événements.

## Modèle de configuration complet et exemple pour DIAL-OUT

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#conf
RP/0/RP0/CPU0:XR(config)#
RP/0/RP0/CPU0:XR(config)#telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)#sensor-group <Sensor-Group-Name>
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# sensor-path <Sensor-Path>
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)#destination-group <Destination-Group-Name>
RP/0/RP0/CPU0:XR(config-model-driven-dest)#address family ipv4 <Destination-IP> port
<Destination-Port>
RP/0/RP0/CPU0:XR(config-model-driven-dest-addr)#encoding <Encoding-Type>
RP/0/RP0/CPU0:XR(config-model-driven-dest-addr)#protocol <Transport-Protocol>
RP/0/RP0/CPU0:XR(config-model-driven-dest-addr)#subscription <Subscription-Name>
RP/0/RP0/CPU0:XR(config-model-driven-subs)#sensor-group-id <Sensor-Group-Name> sample-interval
<0>
RP/0/RP0/CPU0:XR(config-model-driven-subs)#destination-id <Destination-Group-Name>
RP/0/RP0/CPU0:XR(config-model-driven-subs)#commit
RP/0/RP0/CPU0:XR(config-model-driven-subs)#end
RP/0/RP0/CPU0:XR#
```

L'exemple suivant illustre un exemple réel de l'interface de ligne de commande du routeur.

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#conf
RP/0/RP0/CPU0:XR(config)#
RP/0/RP0/CPU0:XR(config)#telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)#sensor-group SensorGroup101
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)# sensor-path Cisco-IOS-XR-infra-statsd-
oper:infra-statistics/interfaces/interface/latest/generic-counters
RP/0/RP0/CPU0:XR(config-model-driven-snsr-grp)#destination-group DestGroup101
RP/0/RP0/CPU0:XR(config-model-driven-dest)#address family ipv4 10.1.1.2 port 5432
RP/0/RP0/CPU0:XR(config-model-driven-dest-addr)#encoding self-describing-gpb
RP/0/RP0/CPU0:XR(config-model-driven-dest-addr)#protocol tcp
RP/0/RP0/CPU0:XR(config-model-driven-dest-addr)#subscription Subscription101
RP/0/RP0/CPU0:XR(config-model-driven-subs)#sensor-group-id SensorGroup101 sample-interval 0
RP/0/RP0/CPU0:XR(config-model-driven-subs)#destination-id DestGroup101
RP/0/RP0/CPU0:XR(config-model-driven-subs)#commit
RP/0/RP0/CPU0:XR(config-model-driven-subs)#end
RP/0/RP0/CPU0:XR#
```

## Modèle de configuration complet et exemple pour l'ACCÈS COMMUTÉ

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#config
RP/0/RP0/CPU0:XR(config)# grpc
RP/0/RP0/CPU0:XR(config-grpc)#port <port-number>
RP/0/RP0/CPU0:XR(config-grpc)telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)#subscription <Subscription-Name>
RP/0/RP0/CPU0:XR(config-model-driven-subs)#sensor-group-id <Sensor-Group-Name> sample-interval
<0>
RP/0/RP0/CPU0:XR(config-model-driven-subs)#commit
RP/0/RP0/CPU0:XR(config-model-driven-subs)#end
RP/0/RP0/CPU0:XR#
```

L'exemple suivant illustre un exemple réel de l'interface de ligne de commande du routeur.

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#config
RP/0/RP0/CPU0:XR(config)# grpc
RP/0/RP0/CPU0:XR(config-grpc)#port 57890
RP/0/RP0/CPU0:XR(config-grpc)telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)#subscription Subscription101
RP/0/RP0/CPU0:XR(config-model-driven-subs)#sensor-group-id SensorGroup101 sample-interval 0
RP/0/RP0/CPU0:XR(config-model-driven-subs)# commit
RP/0/RP0/CPU0:XR(config-model-driven-subs)# end
RP/0/RP0/CPU0:XR#
```

## Validation de la télémétrie avec les commandes SHOW

Du point de vue du routeur, nous pouvons vérifier les paramètres configurés pour chaque groupe de capteurs, groupe de destinations et abonnement

// ALL CONFIGURED SUBSCRIPTIONS  
RP/0/RP0/CPU0:XR#show telemetry model-driven subscription

Subscription: Subscription101 State: ACTIVE

-----

Sensor groups:

| Id             | Interval (ms) | State           |
|----------------|---------------|-----------------|
| SensorGroup101 | 30000         | <b>Resolved</b> |

Destination Groups:

| Id           | Encoding            | Transport | State  | Port | IP           |
|--------------|---------------------|-----------|--------|------|--------------|
| DestGroup101 | self-describing-gpb | tcp       | Active | 5432 | 172.16.128.3 |

// DETAILS ON A PARTICULAR SUBSCRIPTION

RP/0/RP0/CPU0:XR#show telemetry model-driven subscription Subscription101

Subscription: Subscription101

-----

**State: ACTIVE**

Sensor groups:

Id: SensorGroup101

Sample Interval: 30000 ms

Sensor Path: Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters

Sensor Path State: **Resolved**

Destination Groups:

Group Id: DestGroup101

Destination IP: 172.16.128.3

Destination Port: 5432

Encoding: self-describing-gpb

Transport: tcp

State: Active

Total bytes sent: 4893

Total packets sent: 1

Last Sent time: 2019-11-01 10:04:11.2378949664 +0000

Collection Groups:

-----

Id: 1

Sample Interval: 30000 ms

Encoding: self-describing-gpb

**Num of collection: 5**

**Collection time: Min: 6 ms Max: 29 ms**

Total time: Min: 6 ms Avg: 12 ms Max: 29 ms

Total Deferred: 0

Total Send Errors: 0

Total Send Drops: 0

Total Other Errors: 0

Last Collection Start: 2019-11-01 10:06:11.2499000664 +0000

Last Collection End: 2019-11-01 10:06:11.2499000664 +0000

Sensor Path: Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters

RP/0/RP0/CPU0:XR#

// ALL CONFIGURED DESTINATIONS

RP/0/RP0/CPU0:XR#show telemetry model-driven destination

| Group Id     | IP           | Port | Encoding            | Transport | State         |
|--------------|--------------|------|---------------------|-----------|---------------|
| DestGroup101 | 172.16.128.3 | 5432 | self-describing-gpb | tcp       | <b>Active</b> |

RP/0/RP0/CPU0:XR#

```
// PARTICULAR DESTINATION
RP/0/RP0/CPU0:XR#show telemetry model-driven destination DestGroup101
Destination Group:  DestGroup101
-----
Destination IP:      172.16.128.3
Destination Port:    5432
State:               Active
Encoding:            self-describing-gpb
Transport:           tcp
Total bytes sent:  83181
Total packets sent: 17
Last Sent time:      2019-11-01 10:12:11.2859133664 +0000

Collection Groups:
-----
Id: 1
Sample Interval:     30000 ms
Encoding:            self-describing-gpb
Num of collection:  17
Collection time:     Min:      5 ms Max:      29 ms
Total time:          Min:      6 ms Max:      29 ms Avg:      10 ms
Total Deferred:      0
Total Send Errors:   0
Total Send Drops:    0
Total Other Errors:  0
Last Collection Start:2019-11-01 10:12:11.2859128664 +0000
Last Collection End:  2019-11-01 10:12:11.2859134664 +0000
Sensor Path:         Cisco-IOS-XR-infra-statsd-oper:infra-
statistics/interfaces/interface/latest/generic-counters
```

RP/0/RP0/CPU0:XR#

```
// ALL CONFIGURED SENSOR GROUPS
```

```
RP/0/RP0/CPU0:XR#show telemetry model-driven sensor-group
Sensor Group Id:SensorGroup101
Sensor Path:      Cisco-IOS-XR-infra-statsd-oper:infra-
statistics/interfaces/interface/latest/generic-counters
Sensor Path State: Resolved
```

```
// PARTICULAR SENSOR GROUPS
```

```
RP/0/RP0/CPU0:XR#show telemetry model-driven sensor-group SensorGroup101
Sensor Group Id:SensorGroup101
Sensor Path:      Cisco-IOS-XR-infra-statsd-oper:infra-
statistics/interfaces/interface/latest/generic-counters
Sensor Path State: Resolved
```

RP/0/RP0/CPU0:XR#

## Pile de collecte télémétrique

Outre la configuration du routeur, une solution basée sur la télémétrie nécessitait plusieurs composants tels que le collecteur, la base de données et le logiciel de surveillance/analyse. Ces composants peuvent être configurés séparément ou faire partie d'un seul produit complet.

- Un décodeur doit être installé pour prendre les paquets entrants et les transmettre pour stockage ultérieur.
- Une base de données de séries temporelles (également appelée TSDB) est nécessaire pour stocker les informations en flux continu.

- Un outil graphique est également nécessaire pour visualiser les données extraites de la base de données interne.

La description détaillée de la pile de collecte n'entre pas dans le cadre de ce projet. Cisco Crossworks Health Insights permet la télémétrie sans intervention, où les périphériques sont automatiquement configurés à l'aide de la télémétrie et les tables/schémas sont créés dans une base de données de séries chronologiques (TSDB). Il rationalise les frais généraux d'exploitation et de gestion du réseau liés à la collecte et au nettoyage des données, permettant ainsi aux opérateurs de se concentrer sur leurs objectifs commerciaux. L'utilisation d'un collecteur commun pour collecter les données des périphériques réseau via SNMP, CLI et la télémétrie pilotée par modèle permet d'éviter la duplication des données et réduit également la charge sur les périphériques et le réseau.

## Considérations relatives au déploiement de la télémétrie dans un réseau

Les points suivants doivent être pris en compte lors de l'analyse du déploiement de la télémétrie dans un réseau.

### Évolutivité

La télémétrie peut transmettre une quantité considérable de données et il est recommandé d'examiner attentivement les aspects relatifs à l'évolutivité.

### Diffuser uniquement les données requises

Chaque modèle Yang aura plusieurs noeuds leaf. Il est conseillé de préciser les informations requises et celles qui ne le sont pas. Il est recommandé d'explorer les modèles Yang et d'identifier le chemin de données requis par les exemples d'utilisation de la télémétrie.

### Tenez compte de la quantité de données en continu

La quantité totale de données télémétriques diffusées en continu devra être prise en compte sur les points suivants :

1. Allocation de bande passante au sein du réseau
2. QoS
3. Performances des applications/logiciels supplémentaires tels que Logiciel de collecte base de données Séries chronologiques Logiciel d'analyse/visualisation
4. L'efficacité du codage (abordée à la section « Lignes directrices pour la conception de la télémétrie ») a une incidence directe sur la quantité de données transmises. Un GPB compact est recommandé chaque fois que possible.
5. L'intervalle de collecte a un impact direct sur plusieurs aspects, notamment les suivants.  
Utilisation de la bande passante sur le réseau  
Besoin de stockage sur la base de données  
Performances de l'appareil diffusant les données en continu

Il est recommandé d'évaluer la fréquence de la collecte en fonction des exigences de l'application.

Dans l'ensemble, il est recommandé d'envisager de filtrer les données indésirables à la source ou à la destination, si cela est possible. Nous avons la possibilité de filtrer les données indésirables.

Le filtrage peut être effectué sur deux niveaux : -

1. À la source : les périphériques diffusent les données en continu.
2. À la destination : le collecteur collecte et normalise les données. (Le filtrage au niveau du collecteur sort du cadre de ce document)

L'exemple suivant montre que les données sont filtrées uniquement pour les interfaces Hundred Gig dans les chemins des capteurs en appliquant des caractères génériques.

```
sensor-path Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface[interface-name='HundredGigE*']/latest/generic-counters
```

## Références

<https://blogs.cisco.com/sp/the-limits-of-snmp>

<https://blogs.cisco.com/sp/why-you-should-care-about-model-driven-telemetry>

<https://www.cisco.com/c/en/us/td/docs/iosxr/asr9000/telemetry/b-telemetry-cg-asr9000-61x.html>

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.