Migration SNMP vers la télémétrie sur IOS XR

Table des matières

Introduction

Cet article présente les composants SNMP (Simple Network Management Protocol) et fournit une corrélation entre les mises en oeuvre actuelles basées sur la surveillance SNMP dans une approche MDT (Model Driven Telemetry).

SNMP

SNMP est un protocole de la couche applicative qui fournit un format de message pour les communications entre les gestionnaires et les agents SNMP. Le protocole SNMP fournit un cadre normalisé et un langage commun utilisés pour la surveillance et la gestion des périphériques d'un réseau

Composants de SNMP

L'infrastructure SNMP comporte les composants suivants, décrits dans les sections suivantes :

- Gestionnaire SNMP
- Agent SNMP
- SNMP MIB

Gestionnaire SNMP

Le gestionnaire SNMP est un système qui contrôle et surveille les activités des hôtes du réseau à l'aide du protocole SNMP. Le système de gestion le plus courant est un système de gestion de réseau (NMS). Le terme NMS peut s'appliquer soit à un dispositif dédié utilisé pour la gestion du réseau, soit aux applications utilisées sur un tel dispositif.

Agent SNMP

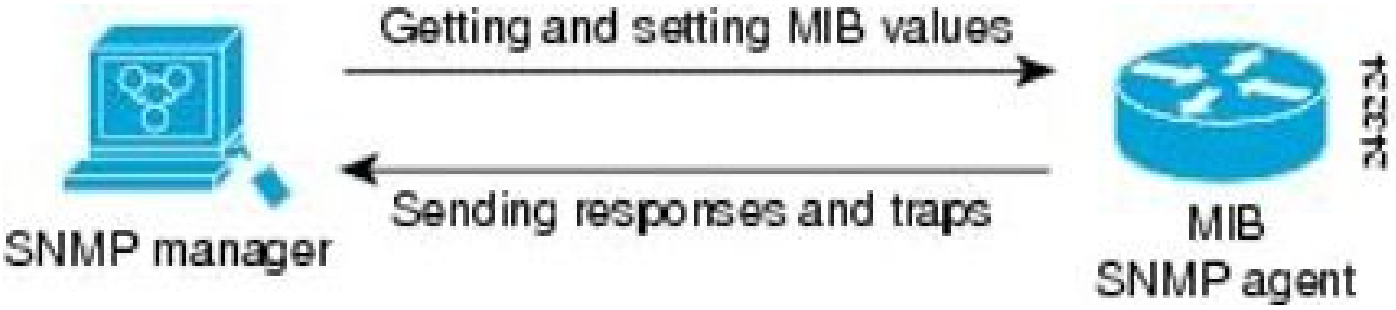
L'agent SNMP est le composant logiciel d'un périphérique géré qui conserve les données du périphérique et les transmet, si nécessaire, à la gestion des systèmes. L'agent réside sur le périphérique de routage (routeur, serveur d'accès ou commutateur).

SNMP MIB

Un agent SNMP contient des variables MIB, dont les valeurs peuvent être demandées ou modifiées par le gestionnaire SNMP par le biais d'opérations « Get » ou « Set ». Un responsable peut obtenir une valeur d'un agent ou stocker une valeur dans cet agent. L'agent collecte des données à partir de la MIB SNMP, le référentiel d'informations sur les paramètres des

périphériques et les données réseau. L'agent peut également répondre aux demandes du gestionnaire pour obtenir ou définir des données.

La figure ci-dessous illustre les communications entre le gestionnaire SNMP et l'agent. Un gestionnaire envoie à un agent des requêtes pour obtenir et définir les valeurs MIB SNMP. L'agent répond à ces demandes. Indépendamment de cette interaction, l'agent peut envoyer au gestionnaire des notifications non sollicitées (interceptions ou notifications) pour l'informer des conditions du réseau.



Opérations SNMP

Les applications SNMP effectuent les opérations suivantes pour récupérer des données, modifier des variables d'objet SNMP et envoyer des notifications :

- Obtention SNMP
- les opérations SNMP SET
- Notifications SNMP

Obtention SNMP

L'opération SNMP GET est effectuée par un NMS pour récupérer des variables d'objet SNMP. Il existe trois types d'opérations GET :

- GET : récupère l'instance d'objet exacte à partir de l'agent SNMP.
- GETNEXT : récupère la variable objet suivante, qui est un successeur lexicographique de la variable spécifiée.
- GETBULK : récupère une grande quantité de données de variable objet, sans avoir à répéter les opérations GETNEXT.

les opérations SNMP SET

L'opération SNMP SET est exécutée par un NMS pour modifier la valeur d'une variable objet.

Notifications SNMP

Une caractéristique clé de SNMP est sa capacité à générer des notifications non sollicitées à partir d'un agent SNMP.

Les notifications non sollicitées (asynchrones) peuvent être générées sous forme de

déroutements ou de demandes d'informations. Les déroutements sont des messages qui alertent le gestionnaire SNMP (Simple Network Management Protocol) d'un état sur le réseau. Les informations sont des déroutements qui incluent une demande de confirmation de réception de la part du gestionnaire SNMP. Les notifications peuvent indiquer une authentification incorrecte de l'utilisateur, des redémarrages, la fermeture d'une connexion, la perte de connexion à un périphérique voisin ou d'autres événements importants.

Les déroutements sont moins fiables que les informations, car le récepteur n'envoie pas d'accusé de réception lorsqu'il reçoit un déroutement. L'expéditeur ne sait pas si le déroutement a été reçu. Un gestionnaire SNMP qui reçoit une notification accuse réception du message avec une PDU (Protocol Data Unit) de réponse SNMP. Si l'expéditeur ne reçoit jamais de réponse, l'information peut être renvoyée. Ainsi, les informateurs sont plus susceptibles d'atteindre leur destination prévue.

Les déroutements sont souvent préférés, même s'ils sont moins fiables, car les informations consomment plus de ressources dans le périphérique et le réseau. Contrairement à un déroutement, qui est rejeté dès son envoi, une information doit être conservée en mémoire jusqu'à ce qu'une réponse soit reçue ou que la requête expire. En outre, les pièges ne sont envoyés qu'une seule fois, alors qu'une notification peut être envoyée plusieurs fois. Les nouvelles tentatives augmentent le trafic et contribuent à une surcharge plus importante sur le réseau. L'utilisation de pièges et d'informations nécessite un compromis entre fiabilité et ressources.

MIB et RFC

Les modules MIB (Management Information Base) sont généralement définis dans les documents RFC (Request for Comments) soumis à l'IETF (Internet Engineering Task Force), un organisme international de normalisation. Les documents RFC sont rédigés par des individus ou des groupes pour examen par l'Internet Society et la communauté Internet dans son ensemble, généralement dans l'intention d'établir une norme Internet recommandée. Avant de recevoir le statut RFC, les recommandations sont publiées sous forme de documents Internet Draft (I-D). Les documents RFC qui sont devenus des normes recommandées sont également étiquetés en tant que documents de normes (STD). Vous pouvez en savoir plus sur le processus de normalisation et les activités de l'IETF sur le site Internet de l'Internet Society à l'adresse http://www.isoc.org. Vous pouvez lire le texte intégral de toutes les RFC, I-D et STD référencées dans la documentation Cisco sur le site Web de l'IETF à l'adresse http://www.ietf.org.

L'implémentation Cisco de SNMP utilise les définitions des variables MIB II décrites dans la RFC 1213 et les définitions des déroutements SNMP décrites dans la RFC 1215.

Cisco fournit ses propres extensions privées MIB avec chaque système. Sauf indication contraire dans la documentation, les MIB d'entreprise Cisco sont conformes aux directives décrites dans les RFC correspondantes. Vous trouverez les fichiers de définition des modules MIB et la liste des MIB pris en charge sur chaque plate-forme Cisco sur le site Web de la base de données MIB Cisco à l'adresse Cisco.com.

Versions de SNMP

Actuellement, les périphériques Cisco prennent en charge les versions suivantes de SNMP :

- SNMPv1 : Simple Network Management Protocol : norme Internet complète, définie dans la RFC 1157. (La RFC 1157 remplace les versions précédentes publiées sous les noms de RFC 1067 et RFC 1098.) La sécurité est basée sur des chaînes de caractères de la communauté.
- SNMPv2c : cadre administratif basé sur des chaînes de communauté pour SNMPv2. SNMPv2c (le « c » signifie « communauté ») est un protocole Internet expérimental défini dans les documents RFC 1901, RFC 1905 et RFC 1906. SNMPv2c est une mise à jour des opérations de protocole et des types de données de SNMPv2p (SNMPv2 Classic) et utilise le modèle de sécurité basé sur la communauté de SNMPv1.
- SNMPv3 : version 3 de SNMP. SNMPv3 est un protocole interopérable normalisé défini dans les documents RFC 3413 à 3415. SNMPv3 fournit un accès sécurisé aux périphériques en authentifiant et en chiffrant les paquets sur le réseau.

Les fonctions de sécurité fournies dans SNMPv3 sont les suivantes :

- Intégrité des messages : s'assurer qu'un paquet n'a pas été altéré pendant son transit.
- Authentification : détermine que le message provient d'une source valide.
- Cryptage : brouillage du contenu d'un paquet pour empêcher qu'il ne soit appris par une source non autorisée.

SNMPv1 et SNMPv2c utilisent tous deux une forme de sécurité à caractère communautaire. La communauté des gestionnaires SNMP peut accéder à la MIB de l'agent définie par une chaîne de communauté.

La prise en charge de SNMPv2c inclut un mécanisme de récupération en masse et un rapport détaillé des messages d'erreur aux stations de gestion. Le mécanisme de récupération en bloc prend en charge la récupération de tables et de grandes quantités d'informations, réduisant ainsi le nombre d'allers-retours requis. La prise en charge améliorée de la gestion des erreurs SNMPv2c inclut des codes d'erreur étendus qui distinguent différents types d'erreurs ; ces conditions sont signalées par un code d'erreur unique dans SNMPv1. Les trois types d'exceptions suivants sont également signalés : aucun objet de ce type, aucune instance de ce type et fin de vue MIB.

SNMPv3 est un modèle de sécurité dans lequel une stratégie d'authentification est configurée pour un utilisateur et le groupe dans lequel l'utilisateur réside. Un niveau de sécurité est le niveau de sécurité permis dans un modèle de sécurité. Une combinaison d'un modèle de sécurité et d'un niveau de sécurité détermine quel mécanisme de sécurité est utilisé lors du traitement d'un paquet SNMP.

Trois modèles de sécurité sont disponibles : SNMPv1, SNMPv2c et SNMPv3. Le tableau ci-dessous répertorie les combinaisons de modèles et de niveaux de sécurité et leur signification.

Maquette	Niveau	Authentification	Chiffrement	Ce qui se passe

v1	noAuthNoPriv	Chaîne de communauté	Non	Utilise une chaîne de caractères de la communauté correspondante pour l'authentification.
v2c	noAuthNoPriv	Chaîne de communauté	Non	Utilise une chaîne de caractères de la communauté correspondante pour l'authentification.
v3	noAuthNoPriv	Nom d'utilisateur	Non	Utilise un nom d'utilisateur correspondant pour l'authentification.
v3	authNoPriv	Message Digest 5 (MD5) ou Secure Hash Algorithm (SHA)	Non	Fournit une authentification basée sur les algorithmes HMAC-MD5 ou HMAC-SHA.
v3	authPriv	MD5 ou SHA	DES (Data Encryption Standard)	Fournit une authentification basée sur les algorithmes HMAC-MD5 ou HMAC-SHA. Fournit un cryptage DES 56 bits en plus de l'authentification basée sur la norme CBC-DES (DES-56).

Un agent SNMP doit être implémenté afin d'utiliser la version de SNMP prise en charge par la station de gestion. Un agent peut communiquer avec plusieurs gestionnaires.

SNMPv3 prend en charge les RFC 1901 à 1908, 2104, 2206, 2213, 2214 et 2271 à 2275. Pour plus d'informations sur SNMPv3, reportez-vous à la RFC 2570, Introduction to Version 3 (Présentation de la version 3 du cadre de gestion de réseau standard Internet) (il ne s'agit pas d'un document standard).

Les modèles Yang

Les modèles Yang représentent une abstraction structurée arborescente d'une caractéristique spécifique ou des caractéristiques matérielles d'un système. Dans les éléments de réseau, un modèle Yang peut représenter un protocole de routage, des réseaux de capteurs physiques internes. Le langage et la terminologie YANG sont décrits dans la RFC 6020 et mis à jour ensuite dans la RFC 7950. Dans un modèle de haut niveau, un modèle Yang organise les données représentant la structure principale en sous-modules et conteneurs qui sont une liste de sous-noeuds associés. Plusieurs types de noeuds sont expliqués ci-dessous.

Un noeud feuille contient des données simples comme un entier ou une chaîne. Il a exactement une valeur d'un type particulier et aucun noeud enfant.

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

Une liste de noeuds leaf est une séquence de noeuds leaf avec exactement une valeur d'un type particulier par noeud leaf.

```
leaf-list domain-search {
          type string;
          description "List of domain names to search";
      }
```

Un noeud conteneur est utilisé pour regrouper les noeuds associés dans une sous-arborescence. Un conteneur n'a que des noeuds enfants et aucune valeur. Un conteneur peut contenir un nombre illimité de noeuds enfants de tout type (y compris des leafs, des listes, des conteneurs et des listes de feuilles).

```
container system {
          container login {
              leaf message {
                  type string;
                  description
                      "Message given at start of login session";
              }
          }
      }
```

Une liste définit une séquence d'entrées de liste. Chaque entrée ressemble à une structure ou à une instance d'enregistrement et est identifiée de manière unique par les valeurs de ses leafs clés. Une liste peut définir plusieurs leafs clés et peut contenir un nombre quelconque de noeuds enfants de tout type (y compris des leafs, des listes, des conteneurs, etc.).

Enfin, un exemple de modèle qui relie tous ces types de notes ressemble à l'exemple suivant :

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

Cependant, la langue Yang utilisée sur les modèles Yang n'indique pas l'organisation des données en conteneurs/listes/feuillets. C'est pourquoi une certaine caractéristique d'un élément de réseau peut être représentée avec divers modèles Yang. Ce défi a été relevé avec les types de modèles Yang suivants :

- Modèles OpenConfig
- Modèles natifs

Modèles OpenConfig

Les modèles OpenConfig ont été développés en utilisant une organisation indépendante du fournisseur pour le modèle représentant une fonctionnalité spécifique. L'avantage de cette approche est qu'un NMS peut utiliser ces modèles pour interagir avec des éléments de réseau dans un environnement multi-fournisseur ou même multi-plateforme.

Comme son nom l'indique, ces modèles sont ouverts et sont accessibles au public pour inspection sur des référentiels comme github sur ce lien :

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

Par exemple, vous pouvez trouver un modèle openconfig pour Border Gateway Protocol (BGP), un autre pour Link Aggregation Control Protocol (LACP) et un autre pour ISIS, avec un modèle spécifique différent. Dans le cas de BGP, vous pouvez trouver un modèle pour les erreurs BGP, un autre pour la politique BGP et ainsi de suite. Les modèles pourraient être liés, et certains modèles peuvent appeler un autre paquet yang. Par exemple, openconfig-bgp-neighbor.yang appartient à openconfig-bgp.yang :

```
module openconfig-bgp {
   yang-version "1";

   ## namespace
   namespace "http://openconfig.net/yang/bgp";
   prefix "oc-bgp";


  ## import some basic inet types
  import openconfig-extensions { prefix oc-ext; }
  import openconfig-rib-bgp { prefix oc-bgprib; }

  ## Include the OpenConfig BGP submodules
  ## Common: defines the groupings that are common across more than
  ## one context (where contexts are neighbor, group, global)
  include openconfig-bgp-common;
  ## Multiprotocol: defines the groupings that are common across more
  ## than one context, and relate to Multiprotocol
  include openconfig-bgp-common-multiprotocol;
  ## Structure: defines groupings that are shared but are solely used for
  ## structural reasons.
  include openconfig-bgp-common-structure;
  ## Include peer-group/neighbor/global - these define the groupings
  ## that are specific to one context
  include openconfig-bgp-peer-group;
  include openconfig-bgp-neighbor;
  include openconfig-bgp-global;
```

En résumé, les modèles OpenConfig sont orientés pour les protocoles communs à toutes les plates-formes, comme les fonctions normalisées IETF ou RFC.

Modèles natifs

En revanche, les modèles natifs sont des modèles orientés constructeur qui couvrent en profondeur les structures spécifiques à une plate-forme particulière. Par exemple, les modèles qui regroupent des capteurs de valeurs physiques à l'intérieur d'un élément de réseau, comme les tensions, les températures, les compteurs ASIC, les compteurs Fabric, etc. Comme ils dépendent de la plate-forme, il est courant de trouver des modèles spécifiques pour NCS6K, ASR9K ou Cisco 8000.

En tant que modèles OpenConfig, les modèles natifs sont également disponibles dans le référentiel Github :

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

Comme ces modèles ont tendance à être beaucoup plus spécifiques et complets que les modèles OpenConfig, ils sont liés à une version logicielle spécifique et peuvent changer d'une version logicielle à l'autre.

Il existe deux catégories principales pour les modèles natifs :

- Modèles « Oper », utilisés pour extraire des informations d'un élément.

Par exemple, Cisco-IOS-XR-eigrp-oper.yang

- Modèles « Cfg », utilisés pour configurer un élément de réseau

Par exemple, Cisco-IOS-XR-eigrp-cfg.yang

En termes généraux, la télémétrie pilotée par modèle utilise des modèles « oper » pour diffuser les données à partir de l'infrastructure et les systèmes de gestion de réseau (NMS), comme NSO, utilisent des modèles « cfg » pour apporter des modifications à la configuration des éléments du réseau.

Les modèles Yang Native et OpenConfig sont présents sur le logiciel XR dans le dossier /pkg/yang et peuvent être listés pour savoir si un modèle Yang est disponible sur une plate-forme. Cet exemple concerne XRrv9k exécutant cXR 6.4.2 :

```
RP/0/RP0/CPU0:xrv9k1#run ls  /pkg/yang | grep isis
Tue Sep 22 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
```

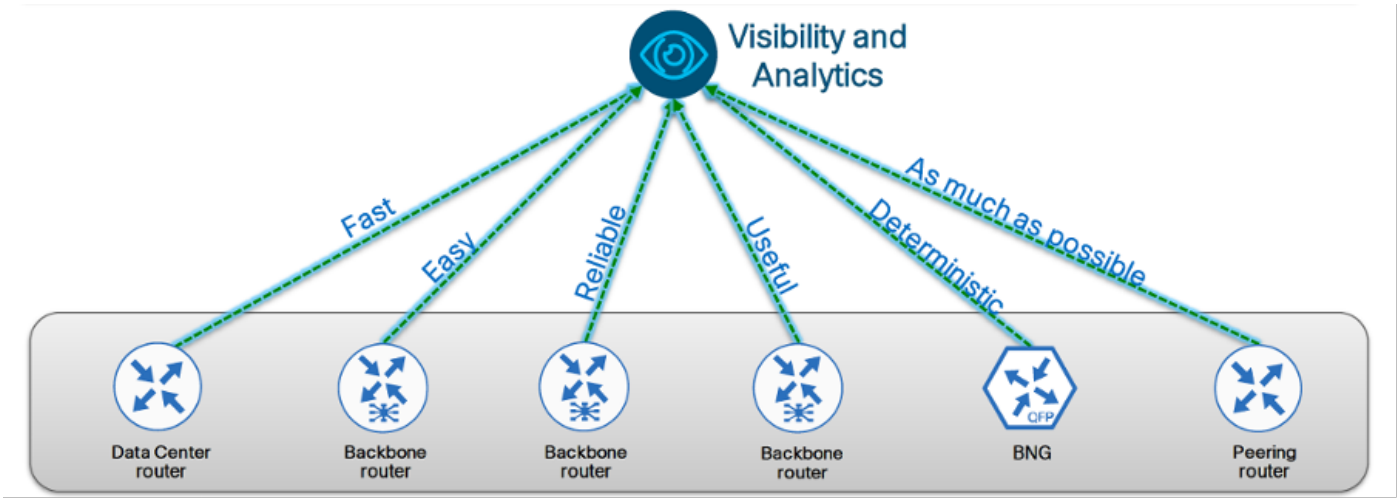
```
Cisco-IOS-XR-clns-isis-oper-sub2.yang
Cisco-IOS-XR-clns-isis-oper-sub3.yang
Cisco-IOS-XR-clns-isis-oper.yang
Cisco-IOS-XR-isis-act.yang
openconfig-isis-lsdb-types.yang
openconfig-isis-lsp.yang
openconfig-isis-policy.yang
openconfig-isis-routing.yang
openconfig-isis-types.yang
openconfig-isis.yang


RP/0/RP0/CPU0:xrv9k1#
```

Télémétrie

La télémétrie est un processus qui permet de collecter des informations à partir de différents éléments distants dans un emplacement central qui regroupe la couche de visibilité et d'analyse.

Dans les environnements réseau, les données peuvent être produites par chaque élément du réseau, les routeurs, les commutateurs entre les autres et les informations peuvent être liées à un très grand ensemble de protocoles spécifiques, de compteurs de performances ou de mesures à partir de capteurs physiques.



En général, les fonctions de visibilité et d'analyse sont situées dans des points centraux des réseaux, la transmission en continu des informations télémétriques est effectuée à l'aide de mécanismes de transport réseau, de sorte que les informations télémétriques doivent être aussi

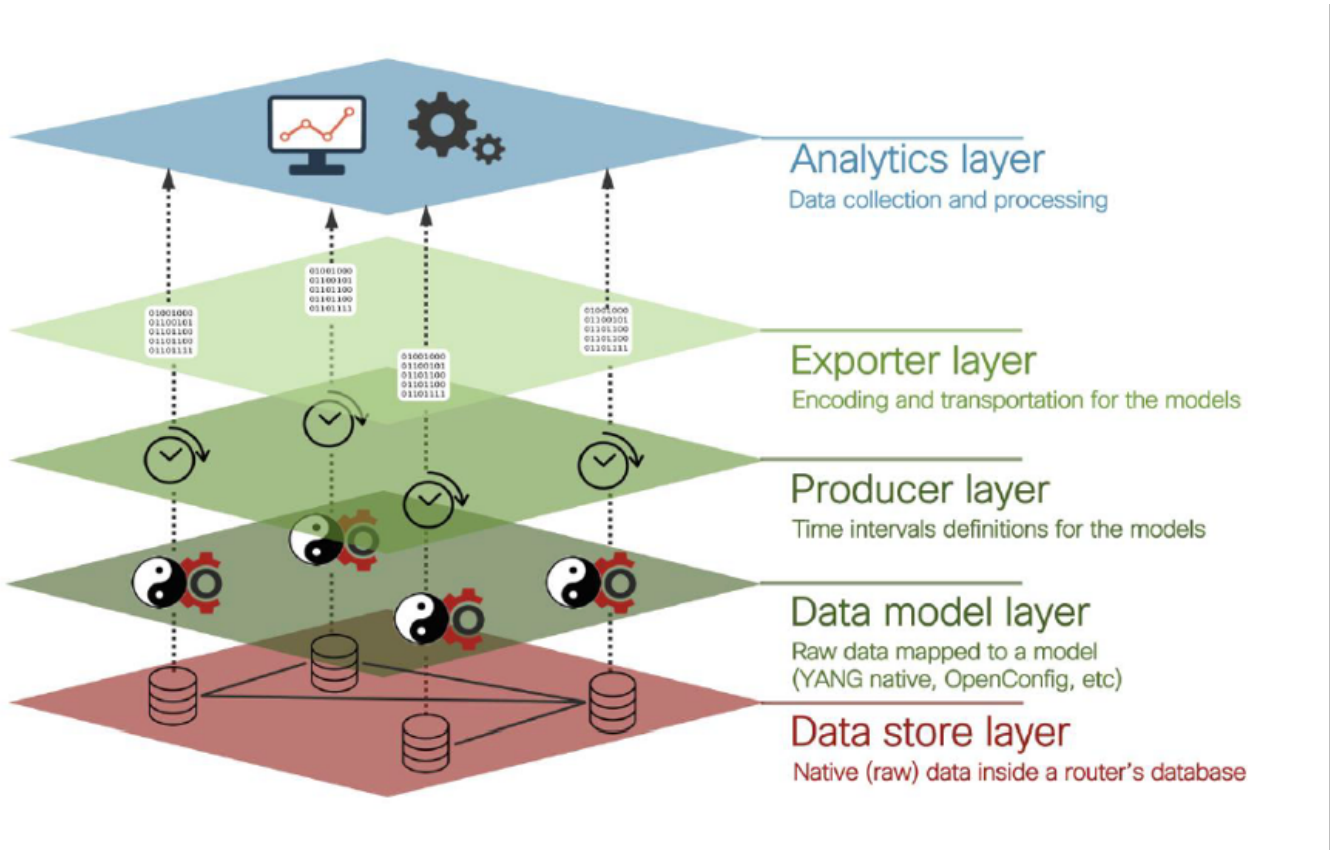
rapides que possible pour permettre une mise à l'échelle.

Contrairement aux anciens mécanismes SNMP, la télémétrie utilise un paradigme Push, selon lequel le réseau doit être configuré pour diffuser ses propres données sans être interrogé à intervalles réguliers, ce qui est la principale caractéristique de la surveillance basée sur SNMP. Cette disposition est souvent appelée abonnement, et elle est basée sur un ensemble de variables à surveiller, l'intervalle régulier pour l'intervalle d'échantillonnage pour la collecte de données, et le système distant pour envoyer ces données sur le réseau.

télémétrie pilotée par un modèle

Les états MDT pour la télémétrie pilotée par le modèle, et comme son nom l'indique, elle est basée sur les modèles Yang. Chaque aspect de l'équipement réseau peut être représenté avec les modèles Yang, par exemple la table de voisinage OSPF, RIB ou les capteurs de température pour chaque composant sur les systèmes modulaires.

En ce qui concerne l'architecture MDT, elle peut être divisée en plusieurs couches :



> Remarque : en ce qui concerne la couche productrice, dans la télémétrie pilotée par modèle, il existe une définition d'intervalle d'échantillonnage qui contrôle la fréquence à laquelle le périphérique consulte la base de données interne pour obtenir des données brutes et organise ces données dans la couche de modèle de données.

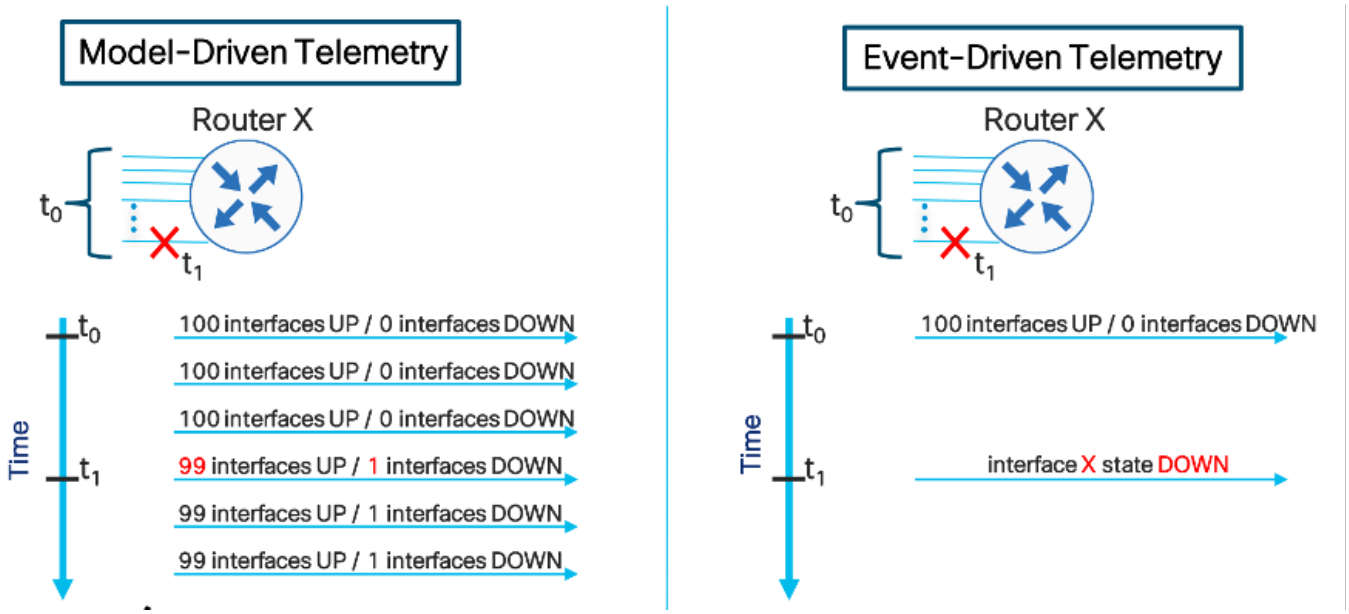
L'abonnement de télémétrie définit également les modèles et les conteneurs/chemins qui produiraient des données à diffuser dans la couche analytique. Cette définition aurait une incidence sur les renseignements pertinents à des fins commerciales. La définition MDT de ce

chemin de capteur serait analogique pour définir l'OID à récupérer via SNMP, puisque les deux techniques produisent des données structurées à un taux d'échantillonnage défini.

télémétrie événementielle

EDT est l'acronyme de Event Driven Telemetry et est également basé sur des modèles Yang pour la structure. La principale différence réside dans le fait que le déclencheur de la collecte et du flux de données n'est pas un intervalle régulier, mais un événement spécifique, tel qu'un franchissement de seuil, des événements de liaison, une défaillance matérielle, etc.

Une comparaison d'un événement avec la télémétrie pilotée par le modèle et la télémétrie pilotée par l'événement est présentée ci-dessous :



> Conseil : cette figure illustre les messages redondants utilisant MDT, mais uniquement les messages représentant des modifications à l'aide de EDT.

Transport

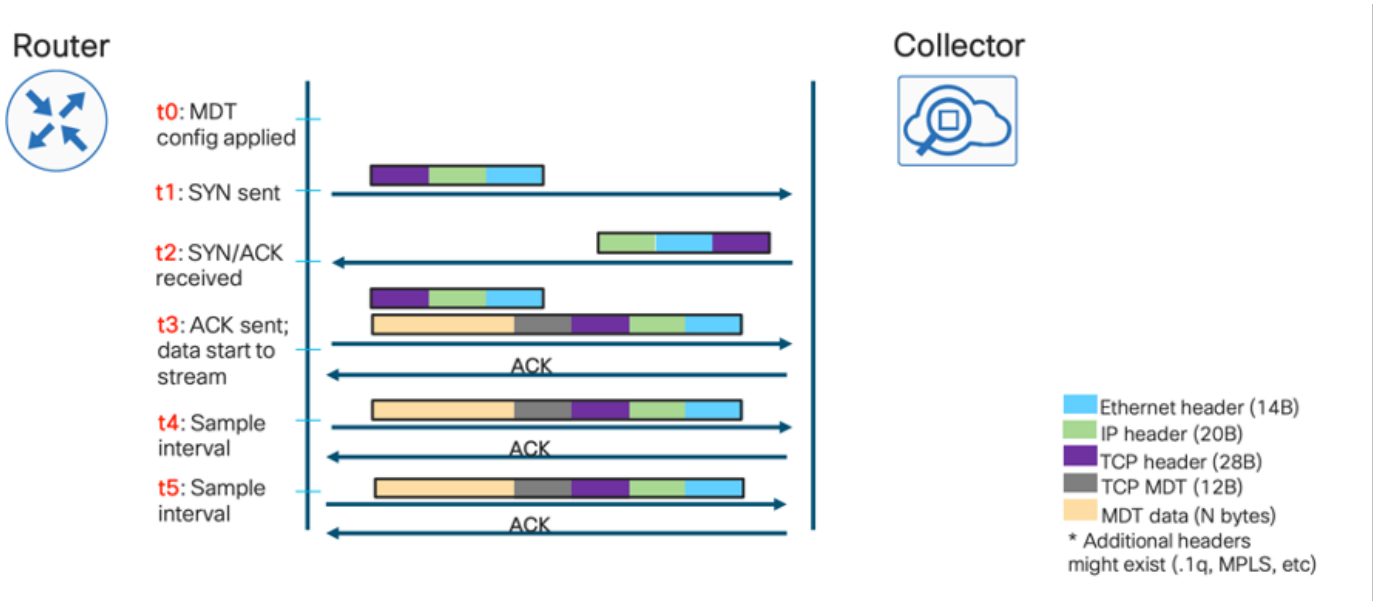
La télémétrie doit être aussi fiable que possible. Il est donc logique d'utiliser le transport basé sur le protocole TCP (Transmission Control Protocol) pour utiliser des sockets orientés session entre l'infrastructure et la couche analytique, qui doit mettre en oeuvre des collecteurs pour établir la session.

Il existe deux approches principales lors de l'utilisation de la télémétrie, et elles diffèrent l'une de l'autre dans le flux initial de connexion en trois étapes.

Remarque : en mode Dial-Out, la configuration de la session est lancée du côté de l'infrastructure, ce qui implique que les capteurs concernés doivent être configurés sur les éléments du réseau. En revanche, l'approche par ligne commutée permet une configuration plus légère sur les éléments du réseau, car le collecteur doit demander des chemins de capteur spécifiques lors de la phase de configuration.

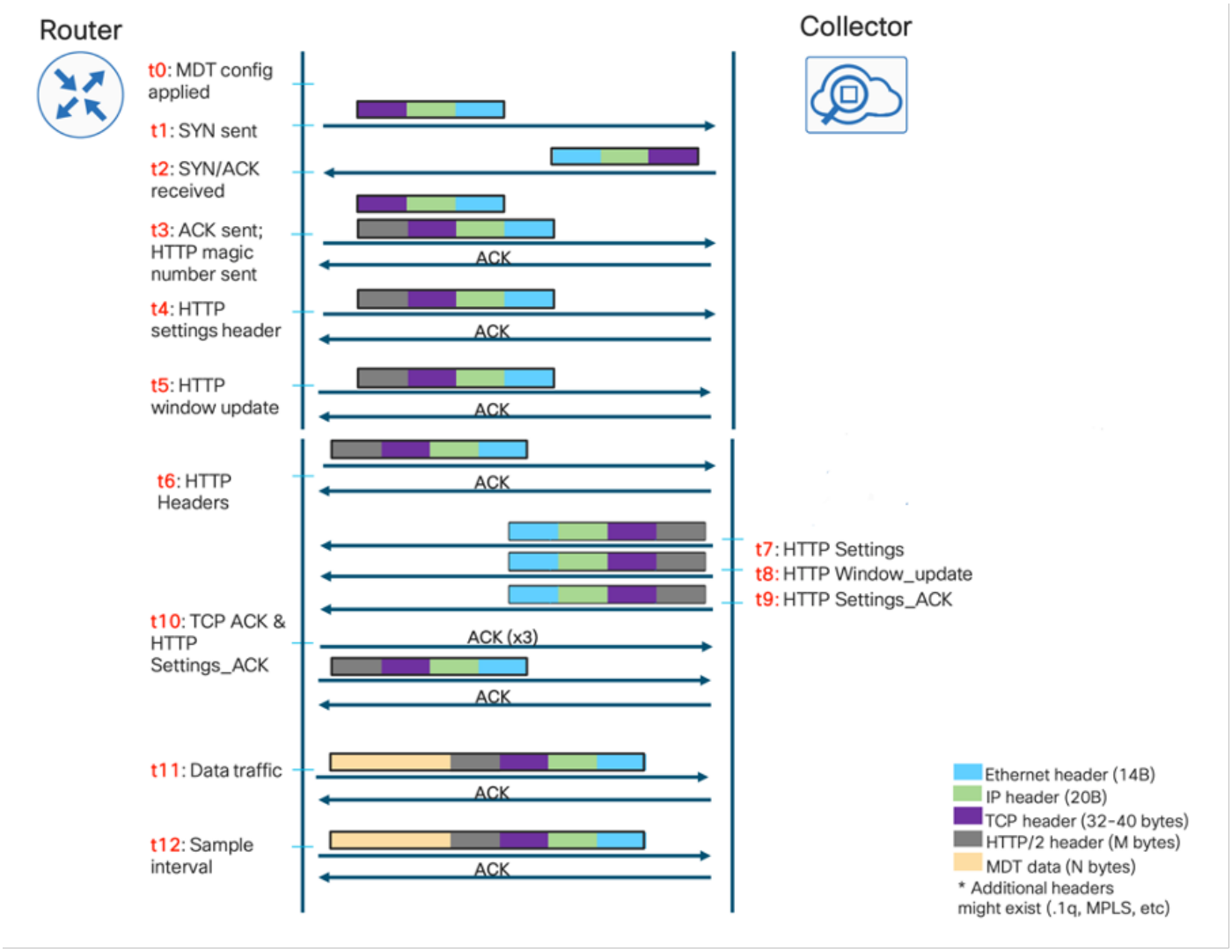
TCP

Le protocole TCP est le moyen le plus simple de créer une session orientée connexion entre un élément réseau et un collecteur de télémétrie. Le flux de données commence du routeur au collecteur, qui renvoie un accusé de réception au routeur à des fins de fiabilité :



gRPC

Puisque Google Protocol RPC (gRPC) fonctionne sur Hypertext Transfer Protocol/2 (HTTP/2), la session elle-même devrait se former à la configuration, et permet le contrôle de vitesse du côté du collecteur nativement :



gNMI/gNOI

gRPC Network Management Interface (gNMI) est un protocole de gestion de réseau gRPC développé par Google. gNMI fournit le mécanisme permettant d'installer, de manipuler et de supprimer la configuration des périphériques réseau, ainsi que de visualiser les données opérationnelles. Le contenu fourni par gNMI peut être modélisé à l'aide de YANG.

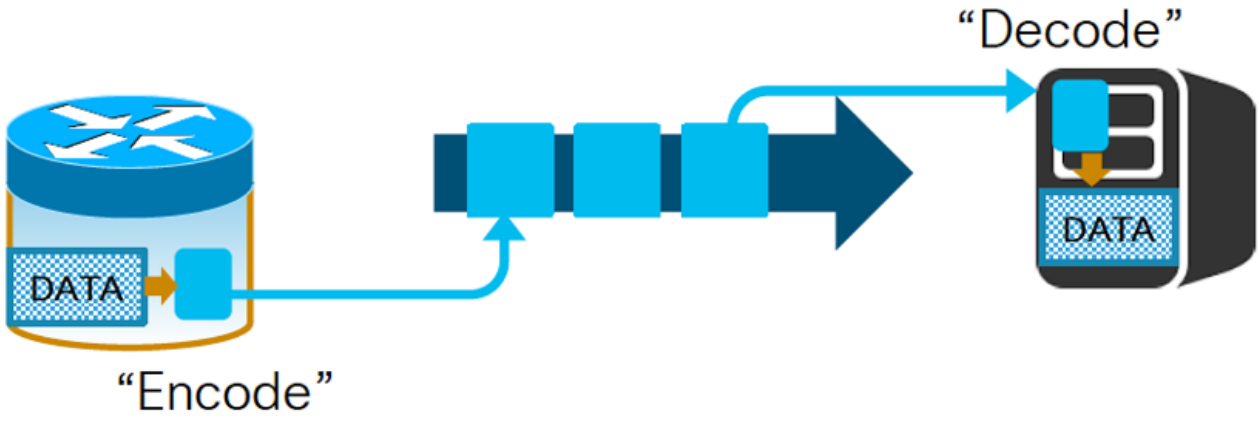
gNMI utilise gRPC-HTTP/2 pour établir une connexion et fournit un canal bidirectionnel entre les éléments du réseau et un NMS qui peut également être un collecteur de télémétrie, mais fournit également une interface pour gérer les périphériques.

Parmi les opérations prises en charge par ce protocole, on trouve gNMI Get, gNMI Set qui renvoient les informations demandées, les messages de réussite ou d'erreur.

gRPC Network Operations Interface (gNOI) est un ensemble de microservices qui utilise le même canal de communication que gNMI, mais qui permet des opérations génériques non liées à la configuration elle-même, telles que ping, redémarrage, modification des certificats SSL, effacement, etc.

Codage

Les modèles Yang définissent la structure des données, leur hiérarchie et le type de chaque noeud feuille qu'ils contiennent. Cependant, la modélisation n'indique pas comment ces données doivent être sérialisées. Ce processus régit la conversion des données structurées en un flux d'octets à envoyer sur la connexion TCP (TCP brut, gRPC, gNMI, etc.).



> Remarque : ce processus doit être mis en oeuvre avec un mécanisme équivalent dans l'élément réseau qui doit coder les données, et le collecteur doit décoder ces données.

JSON

Le premier mécanisme de codage est le format JavaScript Object Notation (JSON), bien connu, mais orienté vers l'humain, car chaque clé est représentée sous la forme d'une chaîne inefficace en termes de taille de message. L'avantage principal de l'utilisation de JSON est qu'il est facile à analyser et à lire en texte comme dans l'exemple suivant :

```
{
“node_id_str":"test-IOSXR ",
"subscription_id_str":" if_rate",
"encoding_path":"Cisco-IOS-XR-infra-statsdoper:infra-statistics/interfaces/interface/latest/datarate",
"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

Le format de codage GPB-KV (Google Protocol Buffers-Key Value) est également appelé GPB autodescriptif, car il utilise des tampons de protocole pour utiliser des messages qui pointent vers des éléments particuliers sur des modèles Yang. Cela implique qu'un seul fichier .proto est nécessaire pour coder/décoder les objectifs, et les clés elles-mêmes à partir des données sont dans des chaînes auto-décrites.

```
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

Enfin, les tampons de protocole Google (GPB), également appelés GPB compacts, poussent cette approche un peu plus loin et nécessitent des fichiers .proto pour mapper chaque clé de la structure, ce qui la rend beaucoup plus efficace en termes de taille de message puisque tout est envoyé sous forme de valeurs binaires. Cependant, l'inconvénient est la nécessité de compiler chaque fichier .proto associé à chaque modèle Yang pris en charge par l'infrastructure/collecteur.

```
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
   }
  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
   }
collection_end_time: 1485794640480
```

Configuration MDT dans IOS XR

Les principaux composants utilisés dans les données de télémétrie pilotées par des modèles en continu sont les suivants :

- Session
- Trajectoire Du Capteur
- Abonnement

- Transport et codage

Les options de session peuvent être Dial-in ou Dial-out, comme nous l'avons vu précédemment. Afin de construire la configuration dans IOS XR.

Mode Dial-Out

pour le mode Dial-Out, le routeur lance une session vers les destinations en fonction de l'abonnement et le processus doit inclure les étapes suivantes :

- Créer un groupe de destinations
- Créer un groupe de capteurs
- Créer un abonnement
- Valider la configuration des appels sortants

Pour créer un groupe de destinations, vous devez connaître l'adresse IPv4 (Internet Protocol Version 4) / IPv6 (Internet Protocol Version 6) du collecteur et le port qui desservira cette application. Vous devez également spécifier le protocole et le codage qui doivent être convenus sur le périphérique réseau et le collecteur.

Enfin, vous devrez peut-être spécifier le VRF (Virtual Routing and Forwarding) utilisé pour communiquer avec l'adresse réseau du collecteur.

Ensuite, un exemple de configuration d'accès sortant est présenté :

```
telemetry model-driven
 destination-group DG1
  vrf MGMT
  address-family ipv4 192.168.122.20 port 5432
   encoding self-describing-gpb
   protocol tcp
  !
 !
```

Les options de codage sont présentées ci-dessous :

<#root>

```
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#encoding ?
  gpb                   GPB encoding
  json                  JSON encoding
  self-describing-gpb  Self describing GPB encoding
```

← Also known as GPB-KV

```
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#encoding
```

Les options des protocoles :

```
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 message compression
  no-tls        No TLS
  tls-hostname  TLS hostname
```

```
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#protocol tcp ?
```

```
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#protocol udp ?
  packetsize  UDP packet size
```

```
RP/0/RP0/CPU0:C8000-1(config-model-driven-dest-addr)#protocol udp
```

Le protocole TCP est simple et nécessite uniquement les paramètres de port associés à l'adresse IPv4/IPv6. Le protocole UDP (User Datagram Protocol) est par contre non orienté connexion, de sorte que l'état du groupe de destination est toujours actif.

La compression dans gRPC peut être réalisée par l'utilisation du mot clé gzip facultatif. gRPC utilise TLS par défaut, donc un certificat doit être installé localement sur le routeur pour cette

utilisation. Ce comportement peut être remplacé par la configuration du mot clé no-tls. Enfin, vous pouvez spécifier un nom d'hôte différent à des fins de certificat en utilisant le mot clé tls-hostname.

Ensuite, il faut ajouter la section sensor-group qui répertorie les chemins de capteurs qui nous intéressent. Cette section est simple, mais il est important de savoir que le chemin du capteur lui-même permet le filtrage pour optimiser plusieurs ressources comme l'unité centrale (UC) et la bande passante.

```
telemetry model-driven
 sensor-group SG1
  sensor-path Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
  sensor-path Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface[interface-na
 !
!
```

> Remarque : le format requis pour le chemin d'accès d'un capteur est <nom-modèle>:<chemin-conteneur>

Ce document présente le mappage de la surveillance basée sur SNMP en utilisant l'OID représentant les « feuilles » dans cette approche héritée dans les modèles YANG, représentés avec des XPATH qui correspondent aux mêmes « feuilles ».

L'étape de configuration finale doit consister à configurer un abonnement, qui lie le groupe de capteurs à une cadence pour la transmission télémétrique en continu vers un groupe de destination.

```
telemetry model-driven
 subscription SU1
  sensor-group-id SG1 sample-interval 5000
  destination-id DG1
 !
!
```

Cet exemple utilise un intervalle d'échantillonnage de 5 000 millisecondes (5 secondes) relatif à la fin de la collecte précédente. Pour modifier ce comportement, vous pouvez modifier le mot clé sample-interval avec l'option strict-timer.

Pour la vérification, vous pouvez utiliser la commande suivante qui couvre l'état de l'abonnement. Cette méthode permet également de couvrir les informations de groupe de capteurs et de groupe de destinations.

```
RP/0/RP0/CPU0:C8000-1#sh telemetry model-driven subscription SU1
Wed Nov 18 15:38:01.397 UTC
Subscription:  SU1
-------------
  State:       ACTIVE
  Sensor groups:
  Id: SG1
    Sample Interval:      5000 ms
    Heartbeat Interval:   NA
    Sensor Path:          Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface[
    Sensor Path State:    Resolved
    Sensor Path:          Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
    Sensor Path State:    Resolved


  Destination Groups:
  Group Id: DG1
    Destination IP:       192.168.122.10
    Destination Port:     5432
    Destination Vrf:      MGMT(0x60000001)
    Encoding:             self-describing-gpb
    Transport:            tcp
    State:                Active
    TLS :                 False
    Total bytes sent:     636284346
    Total packets sent:   4189
    Last Sent time:       2020-11-18 15:37:58.1700077650 +0000
```

```
Collection Groups:
------------------
  Id: 9
  Sample Interval:      5000 ms
  Heartbeat Interval:   NA
  Heartbeat always:     False
  Encoding:             self-describing-gpb
  Num of collection:    1407
  Collection time:      Min:     4 ms Max:    13 ms
  Total time:           Min:     8 ms Avg:    10 ms Max:    20 ms
  Total Deferred:       0
  Total Send Errors:    0
  Total Send Drops:     0
  Total Other Errors:   0
  No data Instances:    1407
  Last Collection Start:2020-11-18 15:37:57.1699545994 +0000
  Last Collection End:  2020-11-18 15:37:57.1699555589 +0000
  Sensor Path:          Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/

  Id: 10
  Sample Interval:      5000 ms
  Heartbeat Interval:   NA
  Heartbeat always:     False
  Encoding:             self-describing-gpb
  Num of collection:    1391
  Collection time:      Min:   178 ms Max:   473 ms
  Total time:           Min:   247 ms Avg:   283 ms Max:   559 ms
  Total Deferred:       0
  Total Send Errors:    0
```

```
    Total Send Drops:     0
    Total Other Errors:   0
    No data Instances:    0
    Last Collection Start:2020-11-18 15:37:58.1699805906 +0000
    Last Collection End:  2020-11-18 15:37:58.1700078415 +0000
    Sensor Path:          Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization


RP/0/RP0/CPU0:C8000-1#
```

Mode d'appel entrant

En mode Dial In, le collecteur établit la connexion avec les éléments du réseau. Ensuite, le collecteur doit indiquer l'intérêt de créer un abonnement.

La configuration comprend les étapes suivantes :

- Activer le service gRPC
- Configurer les groupes de capteurs
- Vérification

Pour activer le service gRPC, la configuration s'affiche :

```
!
grpc
 vrf MGMT
 port 57400
 no-tls
 address-family dual
!
```

Les options sont simples, y compris le VRF et le port TCP. Par défaut, gRPC utilise TLS mais il peut être désactivé avec le mot clé no-tls. Enfin, l'option address-family dual permet la connexion à l'aide des protocoles IPv4 et IPv6.

Ensuite, la numérotation entrante nécessite la définition de groupes de capteurs localement, qui seront utilisés par le collecteur ultérieurement pour définir un abonnement.

```
telemetry model-driven
 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
 !
!
```

À ce stade, pointez sur la configuration du mode Dial-In, qui est terminée, et le collecteur lui-même peut effectuer un abonnement au routeur à l'aide de gRPC. Pour la vérification, vous pouvez utiliser la même approche qu'en mode de numérotation :

```
RP/0/RP0/CPU0:C8000-1#sh telemetry model-driven subscription anx-1605878175837
Fri Nov 20 13:58:37.894 UTC
Subscription:  anx-1605878175837
-------------
  State:       ACTIVE
  Sensor groups:
  Id: SG3
    Sample Interval:      15000 ms
    Heartbeat Interval:   NA
    Sensor Path:          Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
    Sensor Path State:    Resolved
    Sensor Path:          Cisco-IOS-XR-fib-common-oper:fib-statistics/nodes/node/drops
    Sensor Path State:    Resolved


  Destination Groups:
  Group Id: DialIn_1003
    Destination IP:       192.168.122.10
    Destination Port:     46974
    Compression:          gzip
```

```
  Encoding:             json
  Transport:            dialin
  State:                Active
  TLS :                 False
  Total bytes sent:     71000035
  Total packets sent:   509
  Last Sent time:       2020-11-20 13:58:32.1030932699 +0000


Collection Groups:
------------------
  Id: 5
  Sample Interval:      15000 ms
  Heartbeat Interval:   NA
  Heartbeat always:     False
  Encoding:             json
  Num of collection:    170
  Collection time:      Min:   273 ms Max:   640 ms
  Total time:           Min:   276 ms Avg:   390 ms Max:   643 ms
  Total Deferred:       0
  Total Send Errors:    0
  Total Send Drops:     0
  Total Other Errors:   0
  No data Instances:    0
  Last Collection Start:2020-11-20 13:58:32.1030283276 +0000
  Last Collection End:  2020-11-20 13:58:32.1030910008 +0000
  Sensor Path:          Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization


  Id: 6
  Sample Interval:      15000 ms
  Heartbeat Interval:   NA
```

```
    Heartbeat always:     False
    Encoding:             json
    Num of collection:    169
    Collection time:      Min:    15 ms Max:    33 ms
    Total time:           Min:    17 ms Avg:    22 ms Max:    33 ms
    Total Deferred:       0
    Total Send Errors:    0
    Total Send Drops:     0
    Total Other Errors:   0
    No data Instances:    0
    Last Collection Start:2020-11-20 13:58:32.1030910330 +0000
    Last Collection End:  2020-11-20 13:58:32.1030932787 +0000
    Sensor Path:          Cisco-IOS-XR-fib-common-oper:fib-statistics/nodes/node/drops


RP/0/RP0/CPU0:C8000-1#
```

> Conseil : notez qu'aucune cadence, aucun codage, aucune adresse IP de collecteur ou aucun transport n'est codé en dur sur le routeur pour le mode d'appel entrant.

Migration SNMP vers MDT

Afin d'effectuer la migration du protocole SNMP traditionnel vers le modèle de télémétrie, les aspects suivants doivent être traités :

- Migration MIB vers XPATH
- Migration des déroutements vers la télémétrie
- Considérations de sécurité

Migration MIB vers XPATH

Pour cela, nous pourrions classer la MIB en utilisant sa propre hiérarchie qui pourrait être mappée (au moins à un niveau élevé) à une fonctionnalité particulière.

BGP4-MIB

La table suivante représente le nom et le numéro OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par modèle liés aux sessions d'appairage BGP.

Nom OID	Numéro OID	Description OID	XPATH
bgpPeerLastError	1.3.6.1.2.1.15.3.1.14	Dernier code d'erreur et sous-code vus par cet homologue sur cette connexion. Si aucune erreur ne s'est produite, ce champ est égal à zéro. Sinon, le premier octet de cette chaîne d'octets contient le code d'erreur et le deuxième octet contient le sous-code.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/instance-active/default-vrf/neighbor-missing-eor-table/neighbor/last-notify-error-code
BgpPeerOutUpdates	1.3.6.1.2.1.15.3.1.11	Nombre de messages BGP UPDATE transmis sur cette connexion.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/update-messages-out
BgpPeerInUpdates	1.3.6.1.2.1.15.3.1.10	Nombre de messages BGP UPDATE reçus sur cette connexion.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/update-messages-in
bgpPeerNegotiatedVersion	1.3.6.1.2.1.15.3.1.4	Version négociée de BGP exécutée entre les deux homologues.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/negotiation-protocol-version

		Cette entrée DOIT être égale à zéro (0) sauf si bgpPeerState est à l'état openconfirm ou established. Notez que les valeurs autorisées pour cet objet sont comprises entre 0 et 255.	
bgpPeerState	1.3.6.1.2.1.15.3.1.2	État de la connexion homologue BGP.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connection-state
BgpPeerRemoteAddr	1.3.6.1.2.1.15.3.1.7	Adresse IP distante de l'homologue BGP de cette entrée.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connection-remote-address
BgpPeerAdresseLocale	1.3.6.1.2.1.15.3.1.5	Adresse IP locale de la connexion BGP de cette entrée.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connection-local-address
BgpPeerFsmDuréeÉtablie	1.3.6.1.2.1.15.3.1.16	Ce compteur indique combien de temps (en secondes) cet homologue a été dans l'état établi ou combien de temps depuis	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connection-established-time

		que cet homologue a été dans l'état établi pour la dernière fois. Il est mis à zéro lorsqu'un nouvel homologue est configuré ou lorsque le routeur est démarré.	
bgpPeerAdminStatus	1.3.6.1.2.1.15.3.1.3	État souhaité de la connexion BGP. Une transition de 'stop' à 'start' provoquera la génération de l'événement de démarrage manuel BGP. Une transition de 'start' à 'stop' entraînera la génération de l'événement d'arrêt manuel BGP. Ce paramètre peut être utilisé pour redémarrer les connexions d'homologues BGP. Veillez à fournir un accès en écriture à cet objet sans authentification	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/connection-admin-status

		adéquate.	

CISCO-BGP4-MIB

La table suivante représente le nom et le numéro OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par modèle liés à l'échange d'état et de préfixe de session BGP.

Nom OID	Numéro OID	Description OID	XPATH
cbgpPeer2DistantAs	1.3.6.1.4.1.9.9.187.1.2.5.1.11	Numéro du système autonome distant reçu dans le message BGP OPEN.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/i active/default-vrf/sessions/session/remote-a
cbgpPeer2ÉtatPréc	1.3.6.1.4.1.9.9.187.1.2.5.1.29	État précédent de la connexion homologue BGP.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/i active/default-vrf/afs/af/neighb table/neighbor/previous-conn state
cbgpPeer2State	1.3.6.1.4.1.9.9.187.1.2.5.1.3	État de la connexion homologue BGP.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/i active/default-vrf/afs/af/neighb table/neighbor/connection-sta
cbgpPeer2AdresseLocale	1.3.6.1.4.1.9.9.187.1.2.5.1.6	Adresse IP locale de la connexion BGP de cette entrée.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/i active/default-vrf/afs/af/neighb table/neighbor/connection-loc address
cbgpPeer2PréfixesAnnoncés	1.3.6.1.4.1.9.9.187.1.2.8.1.6	Ce compteur est incrémenté lorsqu'un préfixe de route, qui	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/i active/default-vrf/afs/af/neighb table/neighbor/af-data/prefixe adverited

		appartient à une famille d'adresses, est annoncé sur cette connexion. Il est initialisé à zéro lorsque la connexion est soumise à une réinitialisation matérielle.	
cbgpPeer2PréfixesAcceptés	1.3.6.1.4.1.9.9.187.1.2.8.1.1	Nombre de préfixes de route acceptés sur cette connexion, qui appartiennent à une famille d'adresses.	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/i active/default-vrf/afs/af/neighb table/neighbor/af-data/prefixe
cbgpPeerPrefixLimit	1.3.6.1.4.1.9.9.187.1.2.1.1.3	Nombre maximal de préfixes de routage acceptés sur cette connexion	Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/i active/default-vrf/afs/af/neighb table/neighbor/af-data/max-pr
cbgpPeer2PrefixThreshold	1.3.6.1.4.1.9.9.187.1.2.8.1.4	Valeur de seuil de préfixe (%) pour une famille d'adresses sur cette connexion à laquelle un message d'avertissement indiquant que le nombre de préfixes est dépassé ou	Cisco-IOS-XR-ipv4-bgp-oper:bgp/config-instances/cor instance/config-instance-defa vrf/entity-configurations/entity configuration/af-dependent-config/max-prefix-warn-thresh

		une notification SNMP correspondante est générée.	

MIB-QOS BASÉE SUR LA CLASSE CISCO

Le tableau suivant représente le nom et le numéro de l'OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par les modèles et liés aux statistiques dans les classes/politiques de qualité de service (QoS).

Nom OID	Numéro OID	Description OID	XPATH
CbQosCMDropBitRate	1.3.6.1.4.1.9.9.166.1.15.1.1.18	Le débit binaire des abandons par classe comme résultat de toutes les fonctionnalités qui peuvent produire des abandons (par exemple, police, détection aléatoire, etc.).	Cisco-IOS-XR-qos-ma-oper:qos/table-interface/interface/entrée/r politiques-de-service/insta politique-de-service/statistiques/stats-classe/stats-générales/tau d'abandon total Cisco-IOS-XR-qos-ma-oper:qos/table-interface/interface/output/s policy-names/service-polic instance/statistics/class-stats/general-stats/total-dr
CbQosCMDropPkt64	1.3.6.1.4.1.9.9.166.1.15.1.1.14	Compteur de 64 bits de paquets abandonnés par classe résultant de toutes les fonctionnalités pouvant produire des abandons (par exemple, police,	Cisco-IOS-XR-qos-ma-oper:qos/table-interface/interface/input/se policy-names/service-polic instance/statistics/class-stats/general-stats/total-dr packets Cisco-IOS-XR-qos-ma-oper:qos/table-interface/interface/output/s policy-names/service-polic instance/statistics/class-stats/general-stats/total-dr

		détection aléatoire, etc.).	packets
cbQosCMPrePolicyPkt64	1.3.6.1.4.1.9.9.166.1.15.1.1.3	Nombre de paquets entrants de 64 bits avant l'exécution de toute stratégie QoS.	Cisco-IOS-XR-qos-ma-oper:qos/table-interface/interface/input/se policy-names/service-polic instance/statistics/class-stats/general-stats/pre-pol matches-packets Cisco-IOS-XR-qos-ma-oper:qos/table-interface/interface/output/s policy-names/service-polic instance/statistics/class-stats/general-stats/pre-pol matches-packets
cbQosCMName	1.3.6.1.4.1.9.9.166.1.7.1.1.1	Nom de la carte de classe.	Cisco-IOS-XR-qos-ma-oper:qos/table-interface/interface/entrée/r politiques-de-service/insta politique-de-service/statistiques/stats-classe/nom-classe
cbQosCMPostePolitiqueOctet64	1.3.6.1.4.1.9.9.166.1.15.1.1.10	Nombre d'octets sortants sur 64 bits après l'exécution des stratégies QoS.	Cisco-IOS-XR-qos-ma-oper:qos/table-interface/interface/input/se policy-names/service-polic instance/statistics/class-stats/child-policy/class-stats/general-stats/transmi Cisco-IOS-XR-qos-ma-oper:qos/table-interface/interface/output/s policy-names/service-polic instance/statistics/class-stats/child-policy/class-stats/general-stats/transmi

cbQosIfIndex	1.3.6.1.4.1.9.9.166.1.1.1.1.4	ifIndex pour l'interface à laquelle ce service est connecté. Ce champ n'a de sens que si l'interface logique a un ifIndex snmp. Par exemple, la valeur de ce champ n'a pas de sens lorsque cbQosIfType est controlPlane.	Cisco-IOS-XR-infra-policyr oper:policy-manager/global/policy-map/policy-map-types/poli map-type/policy-maps
cbQosConfigIndex	1.3.6.1.4.1.9.9.166.1.5.1.1.2	Index de configuration arbitraire (attribué au système) (indépendant de l'instance) pour chaque objet. Chaque objet ayant la même configuration partage le même index de configuration.	Cisco-IOS-XR-infra-policyr oper:policy-manager/global/policy-map/policy-map-types/poli map-type/policy-maps
cbQosCMPreOctetPolitique64	1.3.6.1.4.1.9.9.166.1.15.1.1.6	Nombre d'octets entrants sur 64 bits avant l'exécution de toute stratégie QoS.	Cisco-IOS-XR-qos-ma-oper:qos/table-interface/interface/input/se policy-names/service-polic instance/statistics/class-stats/child-policy/class-stats/general-stats/pre-pol match-bytes

			Cisco-IOS-XR-qos-ma-oper:qos/table-interface/interface/output/s policy-names/service-polic instance/statistics/class-stats/child-policy/class-stats/general-stats/pre-pol match-bytes

CISCO-ENHANCED-MEMPOOL-MIB

Le tableau suivant représente le nom et le numéro de l'OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par les modèles et liés à l'utilisation de la mémoire.

Nom OID	Numéro OID	Description OID	XPATH
cempPoolMémUtilisé	1.3.6.1.4.1.9.9.221.1.1.1.1.7	Indique le nombre d'octets du pool de mémoire actuellement utilisés par les applications sur l'entité physique.	Cisco-IOS-XR-nto-misc-oper:memory-summary/nodes/node/summa
cempMemPoolHCUsed	1.3.6.1.4.1.9.9.221.1.1.1.1.18	Indique le nombre d'octets du pool de mémoire actuellement utilisés par les applications sur l'entité physique. Cet objet est une version 64 bits de cempMemPoolUsed.	Cisco-IOS-XR-not-misc-oper:memory-summary/nodes/node/detail/to used
cempMemPoolHCFree	1.3.6.1.4.1.9.9.221.1.1.1.1.20	Indique le nombre d'octets du pool de mémoire actuellement inutilisés sur l'entité physique. Cet objet est une version 64 bits de	Cisco-IOS-XR-nto-misc-oper:memory-summary/nodes/node/detail/fr physical-memory

		cempMemPoolFree.	

CISCO-ENTITY-FRU-CONTROL-MIB

Le tableau suivant représente le nom et le numéro OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par modèle et associés aux unités remplaçables sur site du système surveillé.

Nom OID	Numéro OID	Description OID	XPATH
ÉtatFonctionnementFRUPec	1.3.6.1.4.1.9.9.117.1.1.2.1.2	État d'alimentation FRU opérationnel.	Cisco-IOS-XR-in oper:inventaire/e info/power-opera
CefcFRUPowerAdminStatus	1.3.6.1.4.1.9.9.117.1.1.2.1.1	État d'alimentation FRU souhaité par l'administrateur.	Cisco-IOS-XR-in oper:inventaire/e info/power-admir
cefcModuleStatusLastChangeTime	1.3.6.1.4.1.9.9.117.1.2.1.1.4	La valeur de sysUpTime au moment de la modification de cefcModuleOperStatus.	Cisco-IOS-XR-in oper:inventaire/e info/last-operatio
DuréeDisponibilitéModuleCefc	1.3.6.1.4.1.9.9.117.1.2.1.1.8	Cet objet indique la durée de disponibilité du module depuis sa dernière réinitialisation. Cet objet n'est pas persistant ; si un module est réinitialisé, redémarré, mis hors tension, le temps de fonctionnement commence à zéro.	Cisco-IOS-XR-in oper:inventaire/e info/card-up-time
CefcModuleRaisonRéinitialisation	1.3.6.1.4.1.9.9.117.1.2.1.1.3	Cet objet identifie la raison de la dernière réinitialisation effectuée sur le module.	Cisco-IOS-XR-in oper:inventaire/e info/card-reset-re
CefcModuleOperStatus	1.3.6.1.4.1.9.9.117.1.2.1.1.2	Cet objet indique l'état	Cisco-IOS-XR-in

		opérationnel du module.	oper:inventaire/e info/card-operati
CefcModuleAdminStatus	1.3.6.1.4.1.9.9.117.1.2.1.1.1	Cet objet permet de contrôler l'administration du module.	Cisco-IOS-XR-in oper:inventaire/e info/card-adminis

CISCO-ENTITY-SENSOR-MIB

La table suivante représente le nom et le numéro OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par modèle liés aux entités de capteur sur le noeud.

Nom OID	Numéro OID	Description OID	XPATH
ValeurCapteurClient	1.3.6.1.4.1.9.9.91.1.1.1.1.4	Cette variable indique la mesure la plus récente vue par le capteur. Pour afficher ou interpréter correctement la valeur de cette variable, vous devez également connaître entSensorType, entSensorScale et entSensorPrecision. Cependant, vous pouvez comparer entSensorValue avec les valeurs de seuil indiquées dans entSensorThresholdTable sans connaissance sémantique.	Cisco-IOS-XR-in oper:inventaire/e sensor-info/valeu
ÉvaluationSeuilCapteurClient	1.3.6.1.4.1.9.9.91.1.2.1.1.5	Cette variable indique le résultat de la dernière évaluation du seuil. Si la condition de seuil est true, entSensorThresholdEvaluation est true(1). Si la condition de seuil est false, entSensorThresholdEvaluation est false(2). Les seuils sont évalués au taux indiqué par	Cisco-IOS-XR-in oper:inventaire/e

		entSensorValueUpdateRate.	

CISCO-FLASH-MIB

Le tableau suivant représente le nom et le numéro de l'OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par modèle liés au stockage flash sur le système.

Nom OID	Numéro OID	Description OID
NomPartitionFlashCisco	1.3.6.1.4.1.9.9.10.1.1.4.1.1.10	Nom de partition Flash utilisé pour faire référence à une partition par le système. Il peut s'agir de n'importe quelle chaîne de caractères alphanumériques de la forme AAAAAAnn, où A représente un caractère alphanumérique facultatif et n un caractère numérique. Tous les caractères numériques doivent toujours constituer la partie finale de la chaîne. Le système supprime les caractères alphabétiques et utilise la partie numérique pour mapper à un index de partition. Les opérations Flash sont dirigées vers une partition de périphérique basée sur ce nom. Le système utilise le concept de partition par défaut. Il s'agit de la première partition du périphérique. Le système dirige une opération vers la partition par défaut chaque fois qu'un nom de partition n'est pas spécifié. Le nom de la partition est donc obligatoire, sauf si l'opération est effectuée sur la partition par défaut ou si le périphérique ne possède qu'une seule partition (qui n'est pas partitionnée).
TaillePartitionCiscoFlashÉtendue	1.3.6.1.4.1.9.9.10.1.1.4.1.1.13	Taille de la partition Flash. Il doit

		être un multiple entier de ciscoFlashDeviceMinPartitionSize S'il y a une seule partition, cette taille sera égale à ciscoFlashDeviceSize. Cet objet est une version 64 bits de ciscoFlashPartitionSize
CiscoFlashPartitionFreeSpaceExtended	1.3.6.1.4.1.9.9.10.1.1.4.1.1.14	Espace libre dans une partition Flash. Notez que la taille réelle d'un fichier dans Flash inclut une petite surcharge qui représente l'en-tête du fichier du système de fichiers. Certains systèmes de fichiers peuvent également avoir une surcharge d'en-tête de partition ou de périphérique à prendre en compte lors du calcul de l'espace libre. L'espace libre sera calculé comme la taille totale de la partition moins la taille de tous les fichiers existants (fichiers valides/invalides/supprimés et y compris l'en-tête de fichier de chaque fichier), moins la taille de tout en-tête de partition, moins la taille de l'en-tête du fichier suivant à copier. En bref, cet objet donnera la taille du plus grand fichier qui peut être copié dans. L'entité de gestion n'est pas censée connaître ou utiliser les frais généraux tels que les longueurs d'en-tête de fichier et de partition, car ils peuvent varier d'un système de fichiers à l'autre. Les fichiers supprimés dans Flash ne libèrent pas d'espace. Il peut être nécessaire d'effacer une partition pour récupérer l'espace occupé par les fichiers. Cet objet est une version 64 bits de ciscoFlashPartitionFreeSpace

CISCO-PROCESS-MIB

Le tableau suivant représente le nom et le numéro de l'OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par modèle relatifs à l'utilisation du processeur et à l'allocation des ressources pour les processus.

Nom OID	Numéro OID	Description OID	XPATH
cpmCPUTotal1minRév	1.3.6.1.4.1.9.9.109.1.1.1.1.7	Le pourcentage général de CPU occupé pendant les 1 dernières minutes. Cet objet désapprouve l'objet cpmCPUTotal1min et augmente la plage de valeurs à (0..100).	Cisco-IOS-X oper:surveilla système/utilis cpu-une minu
cpmCPUTotal5minRév	1.3.6.1.4.1.9.9.109.1.1.1.1.8	Le pourcentage général de CPU occupé pendant les 5 dernières minutes. Cet objet désapprouve l'objet cpmCPUTotal5min et augmente la plage de valeurs à (0..100).	Cisco-IOS-X oper:surveilla système/utilis cpu-cinq min
cpmCPUTotal15minRév	1.3.6.1.4.1.9.9.109.1.1.1.1.31	Le pourcentage général de CPU occupé pendant les 15 dernières minutes. Cet objet désapprouve l'objet cpmCPUTotal15min et augmente la plage de valeurs à (0..100).	Cisco-IOS-X oper:surveilla système/utilis cpu-quinze m
cpmProcessName	1.3.6.1.4.1.9.9.109.1.2.1.1.2	Nom associé à ce processus. Si le nom comporte plus de 32 caractères, il sera tronqué jusqu'aux 31 premiers caractères, et un « * » sera ajouté comme dernier caractère pour indiquer qu'il s'agit d'un nom de processus tronqué.	Cisco-IOS-X oper:surveilla système/utilis processus/no

cpmProcessTextSegmentSize	1.3.6.1.4.1.9.9.109.1.2.3.1.15	Indique la mémoire de texte d'un processus et de tous ses objets partagés.	Cisco-IOS-X oper:process memory/nod ids/process-i
cpmProcessDynamicMemorySize	1.3.6.1.4.1.9.9.109.1.2.3.1.18	Indique la quantité de mémoire dynamique utilisée par le processus.	Cisco-IOS-X oper:process memory/nod ids/process-i
cpmProcessDataSegmentSize	1.3.6.1.4.1.9.9.109.1.2.3.1.16	Indique le segment de données d'un processus et tous ses objets partagés.	Cisco-IOS-X oper:process memory/nod ids/process-i
CpmProcExtMemAlloué Rév	1.3.6.1.4.1.9.9.109.1.2.3.1.1	Somme de toute la mémoire allouée dynamiquement que ce processus a reçue du système. Cela inclut la mémoire qui peut avoir été retournée. La somme de la mémoire libérée est fournie par cpmProcExtMemFreedRev. Cet objet désapprouve cpmProcExtMemAllocated.	Cisco-IOS-X oper:process memory/nod ids/process-i
CpmProcExtMemFreedRev	1.3.6.1.4.1.9.9.109.1.2.3.1.2	Somme de toute la mémoire que ce processus a retournée au système. Cet objet désapprouve cpmProcExtMemFreed.	Cisco-IOS-X oper:process memory/nod ids/process-i

ENTITY-MIB

Le tableau suivant représente le nom et le numéro OID et le XPATH correspondant à configurer sur les entités physiques associées aux groupes de capteurs de télémétrie pilotés par modèle sur le système.

Nom OID	Numéro OID	Description OID	XPATH

NomPhysiqueEnt	1.3.6.1.2.1.47.1.1.1.1.7	Nom textuel de l'entité physique. La valeur de cet objet doit correspondre au nom du composant attribué par le périphérique local et doit pouvoir être utilisée dans les commandes entrées sur la console du périphérique. Il peut s'agir d'un nom textuel, tel que « console » ou d'un numéro de composant simple (par exemple, numéro de port ou de module), tel que « 1 », en fonction de la syntaxe d'attribution de noms de composant physique du périphérique. S'il n'y a pas de nom local ou si cet objet n'est pas applicable, alors cet objet contient une chaîne vide. Notez que la valeur de entPhysicalName pour deux entités physiques sera la même dans le cas où l'interface de la console ne fait pas de distinction entre elles, par exemple, logement-1 et carte dans logement-1.	Cisco-IOS-XR-snm oper:entity-physica
DescrLogiqueRéseau	1.3.6.1.2.1.47.1.2.1.1.2	Description textuelle de l'entité logique. Cet objet doit contenir une chaîne qui identifie le nom du fabricant pour l'entité logique et doit être défini sur une valeur distincte pour chaque version de l'entité logique.	Cisco-IOS-XR-snm oper:snmp/informa
DescrPhysiqueRéseau	1.3.6.1.2.1.47.1.1.1.1.2	Description textuelle d'une entité physique. Cet objet doit contenir une chaîne qui identifie le nom du fabricant pour l'entité physique et doit être défini sur une valeur	Cisco-IOS-XR-snm oper:snmp/Cisco-I entitymib-oper:enti indexes/

		distincte pour chaque version ou modèle de l'entité physique.	
ENTPhysicalContainerIn	1.3.6.1.2.1.47.1.1.1.1.4	Valeur de entPhysicalIndex pour l'entité physique qui « contient » cette entité physique. La valeur zéro indique que cette entité physique n'est contenue dans aucune autre entité physique. Notez que l'ensemble des relations de « confinement » définit une hiérarchie stricte ; c'est-à-dire que la récursivité n'est pas autorisée. Dans le cas où une entité physique est contenue par plusieurs entités physiques (par exemple, des modules de largeur double), cet objet doit identifier l'entité contenant la valeur la plus faible de entPhysicalIndex.	Cisco-IOS-XR-invn oper:inventaire/ent basic-bag/unique-i
ClassePhysiqueENT	1.3.6.1.2.1.47.1.1.1.1.5	Indication du type de matériel général de l'entité physique. Un agent doit affecter à cet objet la valeur d'énumération standard qui indique le plus précisément la classe générale de l'entité physique, ou la classe principale s'il y en a plusieurs. S'il n'existe aucun identificateur d'enregistrement standard approprié pour cette entité physique, la valeur « other(1) » est renvoyée. Si la valeur est inconnue par cet agent, la valeur « unknown(2) » est renvoyée.	Cisco-IOS-XR-invn oper:inventaire/ent

RévMatérielPhysiqueENT	1.3.6.1.2.1.47.1.1.1.1.8	Chaîne de révision matérielle spécifique au fournisseur pour l'entité physique. La valeur préférée est l'identificateur de révision matérielle réellement imprimé sur le composant lui-même (le cas échéant). Notez que si les informations de révision sont stockées en interne dans un format non imprimable (par exemple, binaire), l'agent doit convertir ces informations dans un format imprimable, d'une manière spécifique à la mise en oeuvre. Si aucune chaîne de révision matérielle spécifique n'est associée au composant physique, ou si ces informations sont inconnues de l'agent, cet objet contiendra une chaîne de longueur nulle.	Cisco-IOS-XR-invn oper:inventaire/ent basic-bag/hardwar
révMicrologicielPhysiqueENT	1.3.6.1.2.1.47.1.1.1.1.9	Chaîne de révision du microprogramme spécifique au fournisseur pour l'entité physique. Notez que si les informations de révision sont stockées en interne dans un format non imprimable (par exemple, binaire), l'agent doit convertir ces informations dans un format imprimable, d'une manière spécifique à la mise en oeuvre. Si aucun microprogramme spécifique n'est associé au composant physique ou si ces informations sont inconnues de l'agent, cet objet contient une chaîne vide.	Cisco-IOS-XR-invn oper:inventaire/ent basic-bag/firmware
RévLogicielsPhysiquesOt	1.3.6.1.2.1.47.1.1.1.1.10	Chaîne de révision logicielle	Cisco-IOS-XR-invn

		spécifique au fournisseur pour l'entité physique. Notez que si les informations de révision sont stockées en interne dans un format non imprimable (par exemple, binaire), l'agent doit convertir ces informations dans un format imprimable, d'une manière spécifique à la mise en oeuvre. Si aucun logiciel spécifique n'est associé au composant physique ou si cette information est inconnue de l'agent, cet objet contient une chaîne vide.	oper:inventaire/ent basic-bag/software
NumSériePhysiqueClient	1.3.6.1.2.1.47.1.1.1.1.11	Chaîne de numéro de série spécifique au fournisseur pour l'entité physique. La valeur préférée est la chaîne du numéro de série réellement imprimée sur le composant lui-même (le cas échéant). Lors de la première instanciation d'une entité physique, la valeur de entPhysicalSerialNum associée à cette entité est définie sur le numéro de série attribué par le fournisseur correct, si cette information est disponible pour l'agent. Si un numéro de série est inconnu ou inexistant, le paramètre entPhysicalSerialNum est défini sur une chaîne de longueur nulle. Notez que les implémentations qui peuvent identifier correctement les numéros de série de toutes les entités physiques installées n'ont pas besoin de fournir un accès en	Cisco-IOS-XR-invm oper:inventaire/ent basic-bag/numéro-

		écriture à l'objet entPhysicalSerialNum. Les agents qui ne peuvent pas fournir de stockage non volatile pour les chaînes entPhysicalSerialNum ne sont pas requis pour implémenter l'accès en écriture pour cet objet. Tous les composants physiques n'ont pas de numéro de série, ni même besoin d'en avoir un. Les entités physiques pour lesquelles la valeur associée de l'objet entPhysicalIsFRU est égale à 'false(2)' (par exemple, les ports de répéteur dans un module de répéteur) n'ont pas besoin de leur propre numéro de série unique. Un agent n'a pas besoin de fournir un accès en écriture pour ces entités et peut renvoyer une chaîne de longueur nulle. Si l'accès en écriture est implémenté pour une instance de entPhysicalSerialNum et qu'une valeur est écrite dans l'instance, l'agent doit conserver la valeur fournie dans l'instance de entPhysicalSerialNum associée à la même entité physique tant que cette entité reste instanciée. Cela inclut les instanciations à travers toutes les réinitialisations/redémarrages du système de gestion du réseau, y compris ceux qui entraînent une modification de la valeur entPhysicalIndex de l'entité physique.	

NomFabricationPhysiqueClient	1.3.6.1.2.1.47.1.1.1.1.12	Nom du fabricant de ce composant physique. La valeur préférée est la chaîne de nom du fabricant réellement imprimée sur le composant lui-même (le cas échéant). Notez que les comparaisons entre les instances des objets entPhysicalModelName, entPhysicalFirmwareRev, entPhysicalSoftwareRev et entPhysicalSerialNum n'ont de sens que parmi les objets entPhysicalEntries ayant la même valeur que entPhysicalMfgName. Si la chaîne du nom du fabricant associée au composant physique est inconnue de l'agent, cet objet contient alors une chaîne de longueur nulle.	Cisco-IOS-XR-invm oper:inventaire/ent basic-bag/nom-fab
NomModèlePhysiqueENT	1.3.6.1.2.1.47.1.1.1.1.13	Chaîne d'identificateur de nom de modèle spécifique au fournisseur associée à ce composant physique. La valeur préférée est la référence visible par le client, qui peut être imprimée sur le composant lui-même. Si la chaîne de nom de modèle associée au composant physique est inconnue de l'agent, cet objet contient alors une chaîne de longueur nulle.	Cisco-IOS-XR-invm oper:inventaire/ent basic-bag/nom-mo

IF-MIB

Le tableau suivant représente le nom et le numéro de l'OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par les modèles et liés aux caractéristiques et aux compteurs de l'interface.

Nom OID	Numéro OID	Description OID	XPATH
siMtu	1.3.6.1.2.1.2.2.1.4	Taille du plus grand paquet pouvant être envoyé/reçu sur l'interface, spécifiée en octets. Pour les interfaces utilisées pour la transmission de datagrammes réseau, il s'agit de la taille du plus grand datagramme réseau pouvant être envoyé sur l'interface.	Cisco-IOS-XR-p cmd-oper:interfaces/i xr/interface/mtu
AdresseSiPhys	1.3.6.1.2.1.2.2.1.6	Adresse de l'interface au niveau de sa sous-couche de protocole. Par exemple, pour une interface 802.x, cet objet contient normalement une adresse MAC. La base MIB spécifique au support de l'interface doit définir l'ordre des bits et des octets, ainsi que le format de la valeur de cet objet. Pour les interfaces qui n'ont pas une telle adresse (par exemple, une ligne série), cet objet doit contenir une chaîne d'octets de longueur nulle.	Cisco-IOS-XR-p cmd-oper:interfaces/i xr/interface-type information/bunc information/mem address
ifType	1.3.6.1.2.1.2.2.1.3	Type d'interface. Des valeurs supplémentaires pour ifType sont attribuées par l'IANA (Internet Assigned Numbers Authority), par mise à jour de la syntaxe de la convention textuelle IANAifType.	Cisco-IOS-XR-p cmd-oper:interfaces/i xr/interface/inter
PaquetsAppelsSortants	1.3.6.1.2.1.2.2.1.17	Nombre total de paquets	Cisco-IOS-XR-p

		que des protocoles de niveau supérieur ont demandé à transmettre et qui n'ont pas été adressés à une adresse de multidiffusion ou de diffusion au niveau de cette sous-couche, y compris ceux qui ont été rejetés ou qui n'ont pas été envoyés. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	cmd-oper:interfaces/i xr/interface-statis interface-stats/pa sent
SiPaquetsHCOutUcast	1.3.6.1.2.1.31.1.1.1.11	Nombre total de paquets que des protocoles de niveau supérieur ont demandé à transmettre et qui n'ont pas été adressés à une adresse de multidiffusion ou de diffusion au niveau de cette sous-couche, y compris ceux qui ont été rejetés ou qui n'ont pas été envoyés. Cet objet est une version 64 bits de ifOutUcastPkts. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	Cisco-IOS-XR-p cmd-oper:interfaces/i xr/interface-statis interface-stats/pa sent
PaquetsifInUcast	1.3.6.1.2.1.2.2.1.11	Nombre de paquets, délivrés par cette sous-couche à une (sous-)couche supérieure, qui	Cisco-IOS-XR-p cmd-oper:interfaces/i xr/interface-statis

		n'étaient pas adressés à une adresse de multidiffusion ou de diffusion au niveau de cette sous-couche. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	interface-stats/pa received
siPaquetsInmoulésHCI	1.3.6.1.2.1.31.1.1.1.7	Nombre de paquets, délivrés par cette sous-couche à une (sous-)couche supérieure, qui n'étaient pas adressés à une adresse de multidiffusion ou de diffusion au niveau de cette sous-couche. Cet objet est une version 64 bits de ifInUcastPkts. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	Cisco-IOS-XR-pf cmd-oper:interfaces/i xr/interface-statis interface-stats/pa received
siErreursSortantes	1.3.6.1.2.1.2.2.1.20	Pour les interfaces orientées paquets, nombre de paquets sortants qui n'ont pas pu être transmis en raison d'erreurs. Pour les interfaces orientées caractères ou de longueur fixe, nombre d'unités de transmission sortantes qui n'ont pas pu être transmises en raison	Cisco-IOS-XR-pf cmd-oper:interfaces/i xr/interface-statis interface-stats/ou errors

		d'erreurs. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	
SiAbandonsSortants	1.3.6.1.2.1.2.2.1.19	Nombre de paquets sortants qui ont été choisis pour être rejetés même si aucune erreur n'a été détectée pour empêcher leur transmission. L'une des raisons possibles de l'abandon d'un tel paquet pourrait être de libérer de l'espace dans la mémoire tampon. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface-statis interface-stats/o drops
PaquetsMultidiffusionSiSortie	1.3.6.1.2.1.31.1.1.1.4	Nombre total de paquets que les protocoles de niveau supérieur ont demandé à transmettre et qui ont été adressés à une adresse de multidiffusion au niveau de cette sous-couche, y compris ceux qui ont été rejetés ou non envoyés. Pour un protocole de couche MAC, cela inclut les adresses de groupe et les adresses fonctionnelles. Des discontinuités dans la	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface-statis interface-stats/m packets-sent

		valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	
PaquetsMultidiffusionSiCoupureHCO	1.3.6.1.2.1.31.1.1.1.12	Nombre total de paquets que les protocoles de niveau supérieur ont demandé à transmettre et qui ont été adressés à une adresse de multidiffusion au niveau de cette sous-couche, y compris ceux qui ont été rejetés ou non envoyés. Pour un protocole de couche MAC, cela inclut les adresses de groupe et les adresses fonctionnelles. Cet objet est une version 64 bits de ifOutMulticastPkts. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface-statis interface-stats/m packets-sent
PaquetsMultidiffusionSiEntrant	1.3.6.1.2.1.31.1.1.1.2	Nombre de paquets, délivrés par cette sous-couche à une (sous-)couche supérieure, qui ont été adressés à une adresse de multidiffusion au niveau de cette sous-couche. Pour un protocole de couche MAC, cela inclut les adresses de groupe et les adresses fonctionnelles. Des	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface-statis interface-stats/m packets-received

		discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	
siPaquetsMultidiffusionHCIn	1.3.6.1.2.1.31.1.1.1.8	Nombre de paquets, délivrés par cette sous-couche à une (sous-)couche supérieure, qui ont été adressés à une adresse de multidiffusion au niveau de cette sous-couche. Pour un protocole de couche MAC, cela inclut les adresses de groupe et les adresses fonctionnelles. Cet objet est une version 64 bits de ifInMulticastPkts. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface-statis interface-stats/m packets-received
ErreursEntrantes	1.3.6.1.2.1.2.2.1.14	Pour les interfaces orientées paquets, nombre de paquets entrants qui contenaient des erreurs les empêchant d'être transmis à un protocole de couche supérieure. Pour les interfaces orientées caractères ou de longueur fixe, nombre d'unités de transmission entrantes qui contenaient des erreurs les empêchant d'être	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface-statis interface-stats/in errors

		transmises à un protocole de couche supérieure. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	
SiAppelsEntrantsAbandonnés	1.3.6.1.2.1.2.2.1.13	Nombre de paquets entrants qui ont été choisis pour être rejetés même si aucune erreur n'a été détectée pour empêcher leur remise à un protocole de couche supérieure. L'une des raisons possibles de l'abandon d'un tel paquet pourrait être de libérer de l'espace dans la mémoire tampon. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface-stati interface-stats/in drops
ifOutOctets	1.3.6.1.2.1.2.2.1.16	Nombre total d'octets transmis hors de l'interface, y compris les caractères de tramage. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface-stati interface-stats/b

ifHCOutOctets	1.3.6.1.2.1.31.1.1.1.10	Nombre total d’octets transmis hors de l’interface, y compris les caractères de tramage. Cet objet est une version 64 bits de ifOutOctets. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface-statis interface-stats/by
ifInOctets	1.3.6.1.2.1.2.2.1.10	Nombre total d’octets reçus sur l’interface, y compris les caractères de tramage. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface-statis interface-stats/by received
ifHCInOctets	1.3.6.1.2.1.31.1.1.1.6	Nombre total d’octets reçus sur l’interface, y compris les caractères de tramage. Cet objet est une version 64 bits de ifInOctets. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface-statis interface-stats/by received
PaquetsDiffusionSortante	1.3.6.1.2.1.31.1.1.1.5	Nombre total de paquets que des protocoles de niveau supérieur ont	Cisco-IOS-XR-p cmd- oper:interfaces/i

		demandé à transmettre et qui ont été adressés à une adresse de diffusion au niveau de cette sous-couche, y compris ceux qui ont été rejetés ou non envoyés. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	xr/interface/inter statistics/full-inte stats/broadcast- sent
ifHCOutBroadcastPkts	1.3.6.1.2.1.31.1.1.1.13	Nombre total de paquets que des protocoles de niveau supérieur ont demandé à transmettre et qui ont été adressés à une adresse de diffusion au niveau de cette sous-couche, y compris ceux qui ont été rejetés ou non envoyés. Cet objet est une version 64 bits de ifOutBroadcastPkts. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface/inter statistics/full-inte stats/broadcast- sent
PaquetsSiDiffusionEntrant	1.3.6.1.2.1.31.1.1.1.3	Nombre de paquets, délivrés par cette sous-couche à une (sous-)couche supérieure, qui ont été adressés à une adresse de diffusion au niveau de cette sous-couche. Des discontinuités	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface-stati interface-stats/b packets-received

		dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	
siPaquetsDiffusionHCIn	1.3.6.1.2.1.31.1.1.1.9	Nombre de paquets, délivrés par cette sous-couche à une (sous-)couche supérieure, qui ont été adressés à une adresse de diffusion au niveau de cette sous-couche. Cet objet est une version 64 bits de ifInBroadcastPkts. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion, et à d'autres moments comme indiqué par la valeur de ifCounterDiscontinuitéTime.	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface-stati interface-stats/b packets-receive
IndexIF	1.3.6.1.2.1.2.2.1.1	Une valeur unique, supérieure à zéro, pour chaque interface. Il est recommandé d'attribuer des valeurs de manière contiguë à partir de 1. La valeur de chaque sous-couche d'interface doit rester constante au moins d'une réinitialisation du système de gestion de réseau de l'entité à la réinitialisation suivante.	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface/if-inc
DescrSi	1.3.6.1.2.1.2.2.1.2	Chaîne textuelle contenant des informations sur	Cisco-IOS-XR-p cmd-

		l'interface. Cette chaîne doit inclure le nom du fabricant, le nom du produit et la version du matériel/logiciel de l'interface.	oper:interfaces/i xr/interface/desc
siVitesse	1.3.6.1.2.1.2.2.1.5	Une estimation de la bande passante actuelle de l'interface en bits par seconde. Pour les interfaces dont la bande passante ne varie pas ou pour celles pour lesquelles aucune estimation précise ne peut être effectuée, cet objet doit contenir la bande passante nominale. Si la bande passante de l'interface est supérieure à la valeur maximale signalée par cet objet, cet objet doit indiquer sa valeur maximale (4 294 967 295) et si HighSpeed doit être utilisé pour indiquer la vitesse de l'interface. Pour une sous-couche qui n'a pas de concept de bande passante, cet objet doit être zéro.	Cisco-IOS-XR-p cmd- oper:interfaces/i xr/interface/banc
ÉtatOpérateur	1.3.6.1.2.1.2.2.1.8	État opérationnel actuel de l'interface. L'état testing(3) indique qu'aucun paquet opérationnel ne peut être transmis. Si ifAdminStatus est down(2), ifOperStatus doit être down(2). Si l'état ifAdminStatus passe à up(1), alors ifOperStatus passe à up(1) si l'interface est prête à transmettre et à recevoir du trafic réseau ; il passe à dormant(5) si	Cisco-IOS-XR-p cmd- oper:interfaces/i non-dynamic/int non-dynamic/op

		l'interface attend des actions externes (comme une ligne série en attente d'une connexion entrante) ; il doit rester à l'état down(2) si et seulement si une défaillance l'empêche de passer à l'état up(1) ; il doit rester à l'état notPresent(6) si l'interface a des composants manquants (généralement, matériels).	
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	État souhaité de l'interface. L'état testing(3) indique qu'aucun paquet opérationnel ne peut être transmis. Lors de l'initialisation d'un système géré, toutes les interfaces commencent par ifAdminStatus à l'état down(2). Suite à une action de gestion explicite ou à des informations de configuration conservées par le système géré, ifAdminStatus passe alors à l'état up(1) ou testing(3) (ou reste à l'état down(2)).	Cisco-IOS-XR-p cmd- oper:interfaces/i non-dynamics/in non-dynamic/ad
NomIF	1.3.6.1.2.1.31.1.1.1.1	Nom textuel de l'interface. La valeur de cet objet doit correspondre au nom de l'interface attribué par le périphérique local et doit pouvoir être utilisée dans les commandes entrées sur la console du périphérique. Il peut s'agir d'un nom textuel, tel que « le0 » ou d'un numéro de port simple, tel que « 1 », selon la syntaxe d'attribution de noms	Cisco-IOS-XR-p cmd- oper:interfaces/i briefs/interface-b interface

		d'interface du périphérique. Si plusieurs entrées de la table ifTable représentent ensemble une interface unique nommée par le périphérique, alors chacune aura la même valeur de ifName. Notez que pour un agent qui répond à des requêtes SNMP concernant une interface sur un autre périphérique (proxy), alors la valeur de ifName pour une telle interface est le nom local du périphérique proxy pour elle. S'il n'y a pas de nom local ou si cet objet n'est pas applicable, alors cet objet contient une chaîne de longueur nulle.	
siHautDébit	1.3.6.1.2.1.31.1.1.1.15	Une estimation de la bande passante actuelle de l'interface en unités de 1 000 000 bits par seconde. Si cet objet indique une valeur de « n », la vitesse de l'interface est comprise entre « n-500 000 » et « n+499 999 ». Pour les interfaces dont la bande passante ne varie pas ou pour celles pour lesquelles aucune estimation précise ne peut être effectuée, cet objet doit contenir la bande passante nominale. Pour une sous-couche qui n'a pas de concept de bande passante, cet objet doit être zéro.	Cisco-IOS-XR-p cmd- oper:interfaces/i briefs/interface- brief/bandwidth6

IP-MIB

Le tableau suivant représente le nom et le numéro de l'OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par les modèles et associés aux statistiques et aux valeurs opérationnelles du protocole Internet (IP).

Nom OID
IPCMinDestUnreachs
ICMPinParmProbs
ICMPinSrcQuenchs
icmpInEchos
icmpInEchoRep
icmpInHorodatages

MasquesAdrEntréeICMP
RépMasqueAdrIcmp
icmpOutMsgs
IcmpAppelsSortantsInaccessibles
IcmpDuréeSortieExcds

IcmpOutParmProbs
ICMPoutSrcQuenchs
RedirectionsSortantesICMP
icmpOutEchos
RépÉchoSortantICMP
icmpOutTimestamps

MasquesAdrSortantICMP
RépMasqueAdrSortantICMP
IndexSiAdrAdr
AdrAdrAdrAdrAdrAdrAdrAdr

ipAdEntNetMask
AdresseDiffusionAdminAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrAdrA

AdressePhysRéseauSupport

IPMIB-COMMUN

Le tableau suivant représente le nom et le numéro de l'OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par modèle et associés aux statistiques IP.

Nom OID	Numéro OID	Description OID	XPATH
ipIfStatsHCOutTransmit	1.3.6.1.2.1.4.31.3.1.31	Nombre total de datagrammes IP que cette entité a fournis aux couches inférieures pour transmission. Cet objet compte les mêmes datagrammes que ipIfStatsOutTransmit, mais autorise des valeurs plus élevées. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion et à d'autres moments, comme indiqué par la valeur de ipIfStatsDiscontinuitéTime.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/nodes/node/statistics/traffic-stats/packets-forwarded
ipIfStatsEntrantReçoit	1.3.6.1.2.1.4.31.3.1.3	Nombre total de datagrammes IP d'entrée reçus, y compris ceux reçus par erreur. Des discontinuités dans la valeur de ce compteur peuvent se produire lors	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/nodes/node/statistics/traffic-stats/input-packets

		de la réinitialisation du système de gestion et à d'autres moments, comme indiqué par la valeur de ipIfStatsDiscontinuitéTime.	
ipIfStatsHCInReçoit	1.3.6.1.2.1.4.31.3.1.4	Nombre total de datagrammes IP d'entrée reçus, y compris ceux reçus par erreur. Cet objet compte les mêmes datagrammes que ipIfStatsInReceives, mais autorise des valeurs plus élevées. Des discontinuités dans la valeur de ce compteur peuvent se produire lors de la réinitialisation du système de gestion et à d'autres moments, comme indiqué par la valeur de ipIfStatsDiscontinuitéTime.	Cisco-IOS-XR-ipv4-io-oper:ipv4-network/nodes/node/statistics/t stats/input-packets

LLDP-MIB

Le tableau suivant représente le nom et le numéro OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par modèle liés aux données opérationnelles LLDP (Link Layer Discovery Protocol) sur le noeud surveillé.

Nom OID	Numéro OID	Description OID	XPATH
IDPlocPortId	1.0.8802.1.1.2.1.3.7.1.3	Valeur de chaîne utilisée pour identifier le composant de port associé à un port donné dans le système local.	Cisco-IOS-XR-ethernet-lldp-oper:lldp/nodes/node/neighbors/de neighbor/port-id-detail
lldpLocPortIdSubtype	1.0.8802.1.1.2.1.3.7.1.2	Type de codage	Cisco-IOS-XR-ethernet-lldp-

		d'identificateur de port utilisé dans l'objet 'lldpLocPortId' associé.	oper:lldp/noeuds/noeuds/voisins/pe voisin/mib/id-port-sous-type
lldpLocChassisIdSousType	1.0.8802.1.1.2.1.3.1	Type de codage utilisé pour identifier le châssis associé au système local.	Cisco-IOS-XR-ethernet-lldp-oper:lldp/noeuds/noeuds/voisins/pe voisin/mib/ID-châssis-sous-type
lldpLocSysName	1.0.8802.1.1.2.1.3.3	Valeur de chaîne utilisée pour identifier le nom système du système local. Si l'agent local prend en charge la norme IETF RFC 3418, l'objet lldpLocSysName doit avoir la même valeur que l'objet sysName.	Cisco-IOS-XR-ethernet-lldp-oper:lldp/noeuds/noeuds/voisins/pe voisin/détail/nom-système
lldpRemSysName	1.0.8802.1.1.2.1.4.1.1.9	Valeur de chaîne utilisée pour identifier le nom système du système distant.	Cisco-IOS-XR-ethernet-lldp-oper:lldp/noeuds/noeuds/voisins/pe voisin/détail/nom-système
lldpRemChassisId	1.0.8802.1.1.2.1.4.1.1.5	Valeur de chaîne utilisée pour identifier le composant de châssis associé au système distant.	Cisco-IOS-XR-ethernet-lldp-oper:lldp/noeuds/noeuds/voisins/pe voisin/ID de châssis

lldpRemChassisIdSousType	1.0.8802.1.1.2.1.4.1.1.4	Type de codage utilisé pour identifier le châssis associé au système distant.	Cisco-IOS-XR-ethernet-lldp-oper:lldp/noeuds/noeuds/voisins/pe voisin
lldpRemPortIdSousType	1.0.8802.1.1.2.1.4.1.1.6	Type de codage d'identificateur de port utilisé dans l'objet 'lldpRemPortId' associé.	Cisco-IOS-XR-ethernet-lldp-oper:lldp/noeuds/noeuds/voisins/pe voisin
lldpRemPortId	1.0.8802.1.1.2.1.4.1.1.7	Valeur de chaîne utilisée pour identifier le composant de port associé au système distant.	Cisco-IOS-XR-ethernet-lldp-oper:lldp/noeuds/noeuds/voisins/pe voisin
lldpLocChassisId	1.0.8802.1.1.2.1.3.2	Valeur de chaîne utilisée pour identifier le composant de châssis associé au système local.	Cisco-IOS-XR-ethernet-lldp-oper:lldp/noeuds/noeuds/voisins/de châssis

MPLS-TE-STD-MIB

Le tableau suivant représente le nom et le numéro OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par modèle liés aux valeurs opérationnelles de l'ingénierie de trafic MPLS (Multiprotocol Label Switching) sur le périphérique géré.

Nom OID	Numéro OID	Description OID	XPATH
mplsTunnelName	1.3.6.1.2.1.10.166.3.2.2.1.5	Nom canonique attribué au tunnel. Ce nom peut être utilisé pour faire référence au tunnel sur le port de console du LSR.	Cisco-IOS-XR-mpls-p2mp-tunnel/tunnel-head/tunnel-name

		Si mplsTunnelIsIf est défini sur true, alors le ifName de l'interface correspondant à ce tunnel doit avoir une valeur égale à mplsTunnelName. Reportez-vous également à la description de ifName dans la RFC 2863.	
mplsTunnelDescr	1.3.6.1.2.1.10.166.3.2.2.1.6	Chaîne textuelle contenant des informations sur le tunnel. En l'absence de description, cet objet contient une chaîne de longueur nulle. Cet objet n'est peut-être pas signalé par les protocoles de signalisation MPLS, par conséquent la valeur de cet objet au niveau des LSR de transit et de sortie PEUT être générée automatiquement ou absente.	openconfig-network-i instances/network-instance/mpls/lsps/co path/tunnels/tunnel/s
mplsTunnelPerfDHCPackets	1.3.6.1.2.1.10.166.3.2.9.1.2	Compteur de capacité élevée pour le nombre de paquets transférés par le tunnel.	openconfig-network-i instances/network-instance/mpls/lsps/co path/tunnels/tunnel/s
mplsTunnelPerfHCBytes	1.3.6.1.2.1.10.166.3.2.9.1.5	Compteur haute capacité pour le nombre d'octets transférés par le tunnel.	openconfig-network-i instances/network-instance/mpls/lsps/co path/tunnels/tunnel/s
AdrIpSautTunnelMpls	1.3.6.1.2.1.10.166.3.2.4.1.5	Adresse de saut de tunnel pour ce saut de tunnel. Le type de cette adresse est déterminé par la valeur du mplsTunnelHopAddrType correspondant. La valeur	Cisco-IOS-XR-mpls-t p2mp-tunnel/tunnel-h head/destination/des

		de cet objet ne peut pas être modifiée si la valeur de l'objet mplsTunnelHopRowStatus correspondant est 'active'.	

RFC2465-MIB

Le tableau suivant représente le nom et le numéro de l'OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par modèle et associés aux valeurs globales IPv6.

Nom OID	Numéro OID	Description OID	XPATH
ipv6AddrPfxLength	1.3.6.1.2.1.55.1.8.1.2	Longueur du préfixe (en bits) associé à l'adresse IPv6 de cette entrée.	Cisco-IOS-XR-ipv6-ma-oper:ipv6-network/nodes/node/interface-data/vrfs/vrf/briefs/brief/address/prefix-length
ipv6AddrAnycastFlag	1.3.6.1.2.1.55.1.8.1.4	Cet objet a la valeur 'true(1)', si cette adresse est une adresse anycast et la valeur 'false(2)' sinon.	Cisco-IOS-XR-ipv6-ma-oper:ipv6-network/nodes/node/interface-data/vrfs/vrf/briefs/brief/address/is-anycast

SNMP-MIB

Le tableau suivant représente le nom et le numéro de l'OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par modèle et associés à l'agent SNMP lui-même, s'il est disponible.

Nom OID	Numéro OID	Description OID	XPATH
DuréeAttenteSys	1.3.6.1.2.1.1.3	Chaîne représentant la disponibilité du système	Cisco-IOS-XR-snmp-agent-oper:snmp/information/system-

			up-time/
IDobjetSys	2.1.3.6.1.2.1.1.2.0	Chaîne représentant l'OID du système	Cisco-IOS-XR-snmp-agent-oper:snmp/information/system-oid/
sysDescr	1.3.6.1.2.1.1.1	Chaîne représentant la description du système	Cisco-IOS-XR-snmp-agent-oper:snmp/information/system-descr

TCP-MIB

La table suivante représente le nom et le numéro OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par modèle et associés à des compteurs spécifiques TCP.

Nom OID	Numéro OID	Description OID	XPATH
ErrTcpIn	1.3.6.1.2.1.6.14	Nombre total de segments reçus par erreur (par exemple, sommes de contrôle TCP incorrectes).	Cisco-IOS-XR-ip-tcp-oper:tcp/nodes/node/statistics/ipv4-traffic/tcp-checksum-error-packets
tcpInSegs	1.3.6.1.2.1.6.10	Nombre total de segments reçus, y compris ceux reçus par erreur. Ce nombre inclut les segments reçus sur les connexions actuellement établies.	Cisco-IOS-XR-ip-tcp-oper:tcp/nodes/node/statistics/ipv4-traffic/tcp-input-packets
tcpOutSegs	1.3.6.1.2.1.6.11	Nombre total de segments envoyés, y compris ceux sur les connexions actuelles, mais à l’exclusion de ceux contenant uniquement des octets retransmis.	Cisco-IOS-XR-ip-tcp-oper:tcp/nodes/node/statistics/ipv4-traffic/tcp-output-packets

UDP-MIB

Le tableau suivant représente le nom et le numéro OID et le XPATH correspondant à configurer sur les groupes de capteurs de télémétrie pilotés par modèle liés aux compteurs spécifiques UDP.

Nom OID	Numéro OID	Description OID	XPATH
udpOutDatagrams	1.3.6.1.2.1.7.4	Nombre total de datagrammes UDP envoyés par cette entité.	Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/statistics/ipv4-traffic/udp-output-packets Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/statistics/ipv6-traffic/udp-output-packets
udpNoPorts	1.3.6.1.2.1.7.2	Nombre total de datagrammes UDP reçus pour lesquels il n'y avait aucune application sur le port de destination.	Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/statistics/ipv4-traffic/udp-no-ports-packets Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/statistics/ipv6-traffic/udp-no-ports-packets
Erreurs udpIn	1.3.6.1.2.1.7.3	Nombre de datagrammes UDP reçus qui n'ont pas pu être livrés pour des raisons autres que l'absence d'une application sur le port de destination.	Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/statistics/ipv4-traffic/udp-checksum-error-packets Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/statistics/ipv6-traffic/udp-checksum-error-packets
udpInDatagrammes	1.3.6.1.2.1.7.1	Nombre total de datagrammes UDP remis aux utilisateurs UDP.	Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/statistics/ipv4-traffic/udp-input-packets Cisco-IOS-XR-ip-udp-oper:/udp/nodes/node/statistics/ipv6-traffic/udp-input-packets

Migration des déroutements SNMP

Les déroutements SNMP sont des messages déclenchés par des événements dynamiques sur le périphérique géré. Par conséquent, ces messages se comportent de la même façon que le concept d'HAE que nous avons traité précédemment.

Côté configuration, MDT permet la même structure pour EDT, qui dépend de la mise en oeuvre sur le collecteur de télémétrie en termes de choix ou de capacités d'accès entrant ou sortant.

Considérations de sécurité

SNMPv2 utilise uniquement la communauté comme mécanisme d'authentification/autorisation. Cependant, comme nous l'avons vu précédemment dans la section SNMP, pourrait utiliser des informations d'identification pour l'authentification et un modèle de chiffrement AES pour protéger les informations.

Dans l'approche de télémétrie, IOS XR permet l'utilisation de techniques gRPC/TLS basées sur des certificats pour effectuer l'authentification. Ces certificats peuvent être utilisés avec un point de confiance central (un serveur AC par exemple). Après le processus d'établissement d'une relation de confiance, tous les messages de télémétrie sont envoyés à l'intérieur d'une session gRPC qui est chiffrée avec TLS réalisant les mêmes avantages de SNMPv3.

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.