

# Whitepaper: ASR9K Model Driven Telemetry

## Inhalt

[Einleitung](#)

[Zielgruppe](#)

[Kurze Einführung in die Telemetry](#)

[Warum Telemetry?](#)

[Die Notwendigkeit einer Abkehr von SNMP](#)

[Vorteile von Streaming-Telemetry](#)

[Modellbasierte Telemetry - Technische Daten](#)

[Telemetryfunktionen](#)

[Telemetrykomponenten](#)

[YANG](#)

[Kodierung](#)

[Verkehr](#)

[Rhythmusbasierte Telemetry und ereignisbasierte Telemetry](#)

[Telemetry-Designrichtlinien](#)

[So wählen Sie ein Codierungsschema aus](#)

[Überlegungen zum Design des Transportnetzwerks](#)

[Evaluierung der Telemetry-Konfigurationsoptionen](#)

[Telemetry-Konfigurationsbeispiele](#)

[IOS-XR](#)

[Untergliederung der DFÜ-Konfiguration](#)

[Sensorgruppen definieren](#)

[Zielgruppen definieren](#)

[Abonnement definieren](#)

[Vollständiges Konfigurationsbeispiel](#)

[Vorteile bei der Hinauswahl](#)

[Untergliederung der Einwahlkonfiguration](#)

[gRPC aktivieren](#)

[Sensorgruppen definieren](#)

[Abonnement definieren](#)

[Vollständige Konfigurationsvorlage und Beispiel](#)

[Vorteile der Einwahl](#)

[Ereignisgesteuerte Telemetry](#)

[Ereignisgesteuerte Telemetrykonfiguration](#)

[Vollständige Konfigurationsvorlage und Beispiel für DIAL-OUT](#)

[Vollständige Konfigurationsvorlage und Beispiel für DIAL-IN](#)

[Telemetry mit SHOW-Befehlen validieren](#)

[Telemetry-Erfassungstapel](#)

[Überlegungen zur Bereitstellung von Telemetry in einem Netzwerk](#)

[Skalierung](#)

[Nur die erforderlichen Daten streamen](#)

# Einleitung

## Zielgruppe

Dieses Whitepaper soll Kunden einen schnellen Überblick über die MDT-Funktion (Model Driven Telemetry) im Allgemeinen und deren Implementierung in den Aggregation Services Router 9000 (ASR9K) vermitteln. Es enthält einige Designrichtlinien und Konfigurationsdetails. Darüber hinaus wird eine Bereitstellung in Betracht gezogen, was bei der Bereitstellung dieser Funktion mit ASR9K hilfreich sein wird. Insgesamt kann dieses Whitepaper eine Kurzreferenz für alle Benutzer sein, die an dieser Funktion arbeiten.

Obwohl die Telemetrie als allgemeine Funktion eingeführt wurde, liegt der Schwerpunkt auf der ASR9K-Implementierung, d. h. nicht alle Funktionen, die von anderen Cisco Plattformen unterstützt werden, werden von der ASR9K-Plattform unterstützt, und einige der Feature-Implementierungen sind möglicherweise spezifisch für ASR9K.

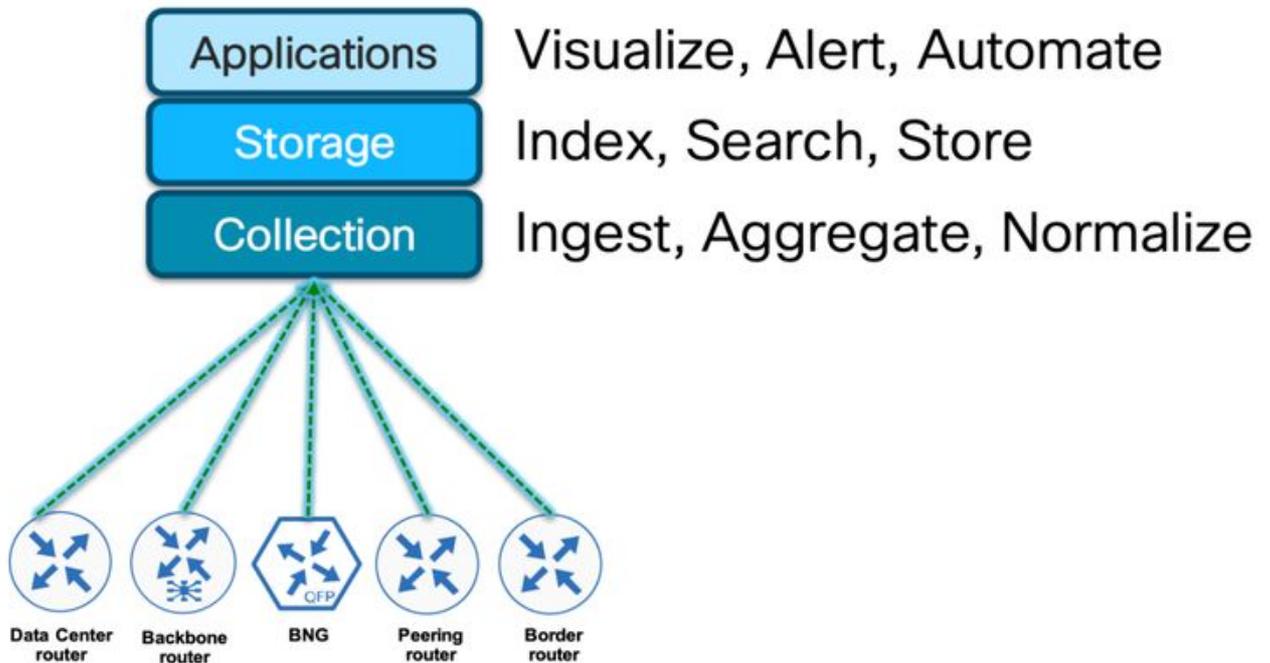
## Kurze Einführung in die Telemetrie

Zunächst einmal ist Telemetrie das Sammeln **nützlicher Betriebsdaten**. Laut Wikipedia ist Telemetrie ein automatisierter Kommunikationsprozess, bei dem Messungen und andere Daten an entfernten oder unzugänglichen Stellen gesammelt und zur Überwachung an Empfangsgeräte übertragen werden. Das Telemetriedokument selbst leitet sich von griechischen Wurzeln ab: tele = remote und metron = Maß.

Für das Netzwerkmanagement vertrauen Netzbetreiber seit langem auf das Simple Network Management Protocol (SNMP). Obwohl SNMP für die Netzwerküberwachung weit verbreitet ist, wurde es nie für die Konfiguration verwendet, obwohl die Möglichkeit zur Konfiguration mit snmp immer gegeben war. Die Bediener haben Automatisierungsskripte für die täglichen Konfigurationsaufgaben geschrieben, aber Skripte sind für solche Aufgaben schwierig und schwer zu verwalten.

Aus diesem Grund haben die Betreiber auf ein datenmodellbasiertes Management umgestellt. Die Netzwerkkonfiguration basiert auf YANG-Datenmodellen, die durch Protokolle wie z. B. netconf vorangetrieben werden. Wenn Sie die Konfiguration einfach nur verschieben, bedeutet das nicht, dass ein konfigurierter Service ausgeführt wird. Es muss einen Mechanismus geben, der die Betriebsdaten der Services gleichzeitig mit der Konfiguration überwachen kann. Hier kommen Betriebsdatenmodelle zum Einsatz, die Telemetrie nutzt, um Informationen aus dem Gerät zu übertragen. Daher ist die Konfiguration YANG-Datenmodell gesteuert, so muss die Überprüfung des Dienstes auch sein; wie der Fall mit Telemetrie, um die gleiche Objekt-Semantik haben. Daher wird der Begriff als **modellgesteuerte Telemetrie** oder Streaming-Telemetrie bezeichnet.

Model Driven Telemetry (MDT) wurde seit Version **6.1.1** in **cXR** (32-Bit IOS XR) eingeführt und ermöglicht die Erfassung und Messung kritischer Daten nahezu in Echtzeit, um die meisten betrieblichen Probleme des modernen Netzwerks schnell zu beheben.



## High-Level-Telemetry-Architektur

MDT nutzt strukturierte Datenmodelle, die vom Netzwerkgerät unterstützt werden, und stellt kritische Daten bereit, die in diesen Datenmodellen definiert sind. **Telemetry** unterstützt Kunden bei der **Verwaltung ihrer heterogenen Netzwerke** mithilfe eines gemeinsamen Netzwerkmanagementsystems, -prozesses und -anwendungen, da die aus dem Netzwerk gesammelten Daten standardbasiert und in der Implementierung einheitlich sind.

Statt auf den Datenabruf (Pull) von einer zentralen Managementstation (in der Regel SNMP NMS) zu warten, senden Netzwerkgeräte mit MDT proaktiv Leistungsdaten zu den wichtigen Netzwerkfunktionen, wie Paketweiterleitungsinformationen, Fehlerstatistiken, Systemstatus, CPU- und Speicherressourcen usw., aus (**Push**).

## Warum Telemetrie?

Die Erfassung von Daten zu Analyse- und Fehlerbehebungszwecken war schon immer ein wichtiger Aspekt bei der Überwachung des einwandfreien Zustands eines Netzwerks. Es stehen mehrere Mechanismen zur Verfügung, z. B. SNMP, CLI und Syslog, um Daten aus einem Netzwerk zu erfassen. Diese Methoden haben dem Netzwerk zwar lange gedient, sind jedoch nicht für moderne Netzwerke geeignet, in denen die Nachfrage nach Automatisierung von grundlegender Bedeutung ist. Netzwerkstatusinformationen, Datenverkehrsstatistiken und wichtige Infrastrukturinformationen werden an eine Remote-Station im NMS gesendet, wo sie zur Verbesserung der Betriebsleistung und zur Verkürzung der Fehlerbehebungszeit verwendet werden. Ein Pull-Modell wie snmp, bei dem ein Client alle Netzwerkknoten abfragt, ist nicht effizient. Die Verarbeitungslast auf den Netzwerkknoten nimmt zu, wenn mehr abzufragende Clients vorhanden sind. Im Gegenteil, ein Push-Modell hat die Möglichkeit, Daten kontinuierlich aus dem Netzwerk zu streamen und den Client zu benachrichtigen. Telemetrie ermöglicht das Push-Modell, das einen nahezu Echtzeitzugriff auf Überwachungsdaten ermöglicht.

Streaming-Telemetrie bietet einen Mechanismus, um Daten von Routern auszuwählen und sie in einem Standardformat zur Überwachung an Remote-Verwaltungsstationen zu übertragen. Dieser Mechanismus ermöglicht eine Feinabstimmung des Netzwerks auf der Grundlage von Echtzeitdaten, was für seinen reibungslosen Betrieb von entscheidender Bedeutung ist. Die feinere Granularität und höhere Datenfrequenz, die durch Telemetrie verfügbar sind, ermöglichen

eine bessere Leistungsüberwachung und damit eine bessere Fehlerbehebung.

Es trägt zu einer serviceeffizienteren Bandbreitennutzung im Netzwerk, zur Verbindungsauslastung, zur Risikobewertung und zur Skalierbarkeit bei. Dank Streaming-Telemetrie stehen Netzbetreibern mehr Daten nahezu in Echtzeit zur Verfügung, was die Entscheidungsfindung verbessert.

## Die Notwendigkeit einer Abkehr von SNMP

SNMP gibt es seit drei Jahrzehnten, und die Art und Weise, wie es funktioniert, hat sich nicht geändert, um den Überwachungsanforderungen moderner Netzwerke gerecht zu werden. Das eigentliche Problem ist die Ausführungsgeschwindigkeit von SNMP.

Die drei größten Herausforderungen, die sich durch SNMP stellen, sind Teil des grundlegenden Betriebsverhaltens von SNMP. Daher bietet SNMP wenig/keinen Verbesserungsbedarf, und Telemetrie geht alle drei Probleme von Natur aus an.

- **Ausführungsgeschwindigkeit und Bedarf an Echtzeitüberwachung**

SNMP verwendet PULL Model - GetBulk / GetNext Operationen, die linear arbeiten, indem sie die Tabellen von einer Spalte zu einer anderen durchlaufen. Darüber hinaus sind bei großen Tabellen, die nicht in ein Paket passen, mehrere Anforderungen erforderlich. Dies ist der größte Engpass, der SNMP verlangsamt, und die gesendeten Daten sind oft um einen bestimmten Zeitfaktor in Minuten veraltet. Diese Verzögerung ist für moderne Anforderungen an die Netzwerküberwachung schlicht nicht akzeptabel.

MDT (Model Driven Telemetry) verwendet das PUSH-Modell und unterliegt keinen oben genannten Beschränkungen, da bekannt ist, welche Daten an wen und in welchem Intervall gesendet werden sollen. Es ist nur eine Suche erforderlich, um Daten zu sammeln, und es werden vorgefertigte interne Vorlagen verwendet, um die internen Vorgänge extrem schnell zu gestalten. So können wesentlich mehr Daten in deutlich kürzerer Zeit bereitgestellt werden.

- **Zusätzliche Gemeinkosten und fehlende Optimierungsoptionen**

Die vom SNMP abgefragten Daten werden als interne Datenstrukturen gespeichert und müssen vom Knoten intern konvertiert werden. Dies ist eine zusätzliche Aufgabe hinter den Kulissen, bei der der Netzknoten interne Datenstrukturen dem SNMP-Format zuordnet. Es werden interne Optimierungen durchgeführt, die jedoch noch nicht ausreichen.

Andererseits zieht Telemetrie die internen Datenstrukturen direkt heraus und führt eine minimale Verarbeitung durch, bevor sie diese Daten aussendet, sodass die aktuellsten Daten mit möglichst wenig Zeit und Aufwand bereitgestellt werden.

- **Linearität der Workload**

Jede zusätzliche Polling-Station führt zu einer zusätzlichen Arbeitsbelastung des Knotens, selbst wenn wir zum gleichen Zeitpunkt exakte Daten abfragen. Der parallele Zugriff auf dieselbe MIB von mehreren Polling-Stationen kann zu einer langsameren Antwort und einer höheren CPU-Auslastung führen. Dies zeigt sich insbesondere bei großen Tischen, bei denen mehrere Stationen auf unterschiedliche Teile derselben MIB-Tabelle zugreifen.

Telemetrie hingegen muss Daten einmalig abrufen und die Pakete replizieren, wenn dieselben Daten für mehrere Ziele benötigt werden. Push-Modell schlägt SNMP Pull für Geschwindigkeit

und Skalierung.

Mit MDT ändert sich der Ansatz zur Datenerfassung grundlegend, und die grundlegenden Prinzipien werden in der nachfolgenden Tabelle aufgelistet und mit den Schlüsselpunkten der SNMP-Technologie verglichen.

### Simple Network Management Protocol (SNMP)

Nicht-Echtzeitinformationen  
Unzureichend skalierbar  
Pull-Modell  
Nicht automatisiert

### Modellgestützte Telemetrie (MDT)

Echtzeitinformationen  
Hochgradig skalierbar  
Push-Modell  
Bereit für Automatisierung/datenmodellgesteuert

## Vorteile von Streaming-Telemetrie

Die gestreamten Echtzeit-Telemetriedaten sind nützlich in:

**Kapazitätsplanung/Datenverkehrsoptimierung:** Wenn die Bandbreitennutzung und Paketverluste in einem Netzwerk häufig überwacht werden, ist es einfacher, Links hinzuzufügen oder zu entfernen, Datenverkehr umzuleiten, Richtlinien zu ändern usw. Mit Technologien wie schnellem Reroute kann das Netzwerk auf einen neuen Pfad wechseln und schneller als mit dem SNMP-Abfrageintervallmechanismus eine neue Route erstellen. Streaming von Telemetriedaten sorgt für kurze Reaktionszeiten für schnelleren Datenverkehr.

**Bessere Transparenz:** Trägt zu einer schnellen Erkennung und Vermeidung von Ausfällen bei, die nach einer Störung im Netzwerk auftreten können.

## Modellbasierte Telemetrie - Technische Daten

Im folgenden Abschnitt werden die technischen Funktionen und Hauptkomponenten der modellgesteuerten IOS XR-Telemetrie (MDT) behandelt.

### Telemetriefunktionen

Das Telemetrie-Framework ist in drei separate und miteinander verbundene Funktionsblöcke unterteilt.

Im ersten Block geht es um die **Datendarstellung**, d. h. darum, wie die informationsbezogene Analyse oder Messung an Bord organisiert wird.

Im zweiten Block geht es um **Kodierung**. Jedes Abtastintervall übersetzt Telemetry die oben genannten Messdaten in ein Format, das über die Leitung serialisiert werden kann. Selbstverständlich muss der Controller am anderen Ende in der Lage sein, die Daten zu dekodieren, um eine identische Kopie der ursprünglichen Daten vom Gerät senden zu lassen.

Im letzten Block geht es um **Transport**. Dies ist der Protokoll-Stack, der zum Übertragen von Daten zwischen Geräten verwendet wird.

In der folgenden Tabelle ist die Hauptstruktur der modellgesteuerten Telemetrie-Bausteine zusammengefasst:

**Funktion**

**Komponenten**

Datendarstellung  
Kodierung  
Verkehr

YANG-Datenmodelle  
Selbstbeschreibendes GPB/GPB  
TCP/gRPC

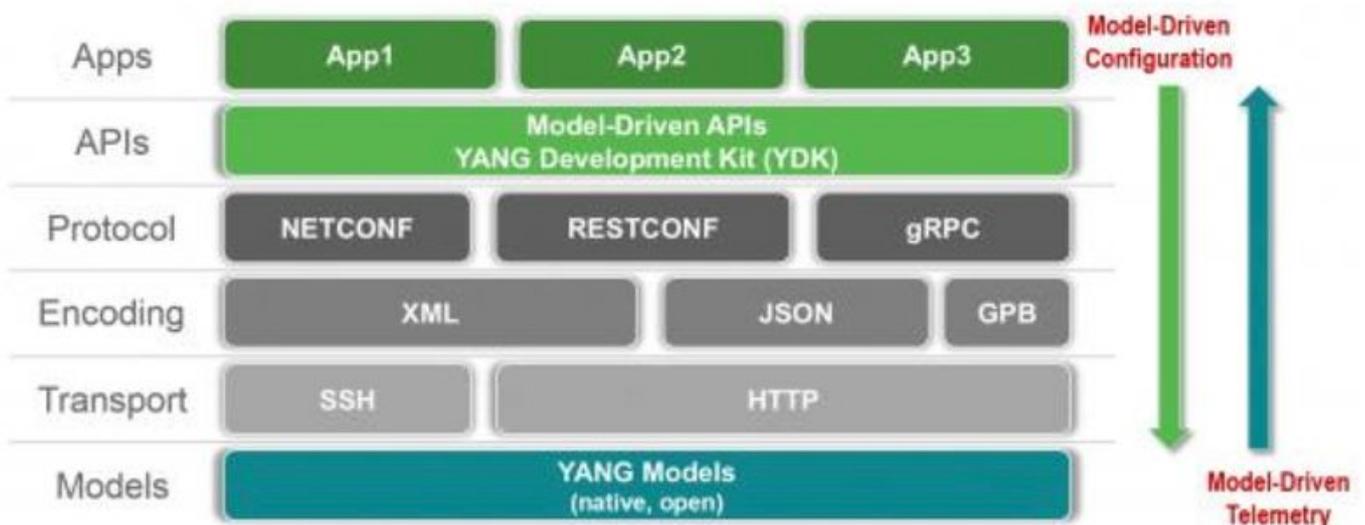
Tabelle 3 Telemetrie-Bausteine

## Telemetrikomponenten

Bevor Sie verstehen, wie Telemetrie und die zugrunde liegenden Konfigurationselemente funktionieren, müssen Sie die verschiedenen Komponenten von Telemetrie kennen, um eine optimale Einrichtung bewerten zu können. Telemetrie basiert auf dem IOS XR Programmability Stack, in dem ein neues Infrastruktur-Framework die wesentlichen Funktionen für die Netzwerkautomatisierung bereitstellt.

YANG wurde vor kurzem zu einem Standard für Datenmodellierung und wird vom Cisco Programmability Stack verwendet, um einen strukturierten Datensatz zu bilden, der so schnell wie möglich codiert und über das Netzwerk übertragen werden kann. Die Flexibilität von YANG bietet den großen Vorteil, auch als Konfigurationstool für Automatisierungsprozesse eingesetzt werden zu können. Diese Datenmodelle werden mit speziellen Kodierungsformaten und Transportprotokollen kombiniert, um MDT zu einer umfassenden Lösung für Netzwerkanalysen zu machen.

Für die modellgetriebene Telemetrie-Konfiguration wird das YANG-Datenmodell zu einer wichtigen Komponente, um das notwendige Datenstreaming für die Erfassung und Analyse zu ermöglichen.



IOS XR-Programmierbarkeitsstapel

## YANG

Yang wird als "Datenmodellierungssprache zur Modellierung von Konfigurationsdaten, Statusdaten und Benachrichtigungen für Netzwerkmanagementprotokolle" definiert. Aufgrund seiner Abkopplung von einer typischen Programmiersprachenarchitektur kann YANG so implementiert werden, dass es mit einer Vielzahl von Tools interagiert.

YANG Modellierung Datenstruktur basiert auf dem Konzept von Modulen & Untermodulen, die eine Hierarchie von Daten in einem Baum wie die Art definiert, die für mehrere Operationen verwendet werden können, einschließlich Konfigurationsaktionen und Benachrichtigungshandling.

Es stehen mehrere Quellen von YANG-Modellen zur Verfügung, von denen die folgenden drei als primär gelten:

- Native Modelle/spezifische Cisco Produkte
- OffeneKonfiguration
- IETF

**Cisco-spezifische Modelle:** Diese werden auch als **natives Modell** bezeichnet und von verschiedenen Geräteherstellern, darunter Cisco, veröffentlicht. z. B. Cisco-IOS-XR-ptp-oper.yang

**OpenConfig-Modelle:** OpenConfig ist eine informelle Arbeitsgruppe von Netzbetreibern. OpenConfig definiert gängige YANG-Modelle, die von allen Anbietern unterstützt werden sollten, um geschäftskritische Funktionen zu konfigurieren. z. B. openconfig-interfaces.yang

**IETF-Modelle:** IETF definiert auch einige gängige YANG-Module, die grundlegende Konfigurationen für Schnittstellen, QoS und andere gängige Datentypen (wie IPv4, IPv6 usw.) beschreiben. z. B. ietf-syslog-types.yang

Cisco unterstützt die verfügbaren OpenConfig-Modelle. Die Anbieter konvergieren zu einer standardisierten Methode der Datenmodellierung, um eine heterogene Umgebung zu unterstützen.

Es gibt **drei Arten** von Yang-Modellen:

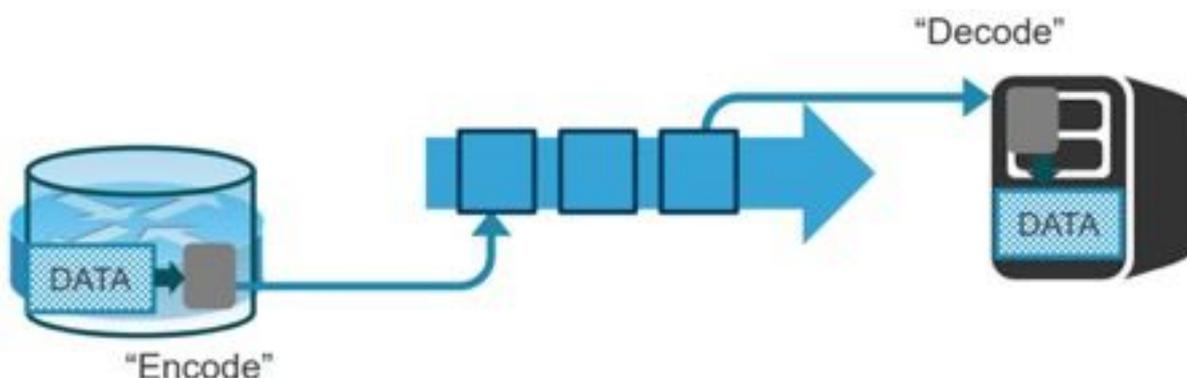
1. Betrieb
2. Konfiguration
3. Aktion

Telemetrie interessiert sich nur für **betriebliche Yang-Modelle**, die als **\*-oper-\*.yang** identifiziert werden können.

YANG ist in RFC 7950 definiert: <https://tools.ietf.org/html/rfc7950>.

## Kodierung

Die Codierung (oder "Serialisierung") übersetzt Daten (Objekte, Status) in ein Format, das über das Netzwerk übertragen werden kann. Wenn der Empfänger die Daten decodiert ("deserialisiert"), hat er eine semantisch identische Kopie der Originaldaten.

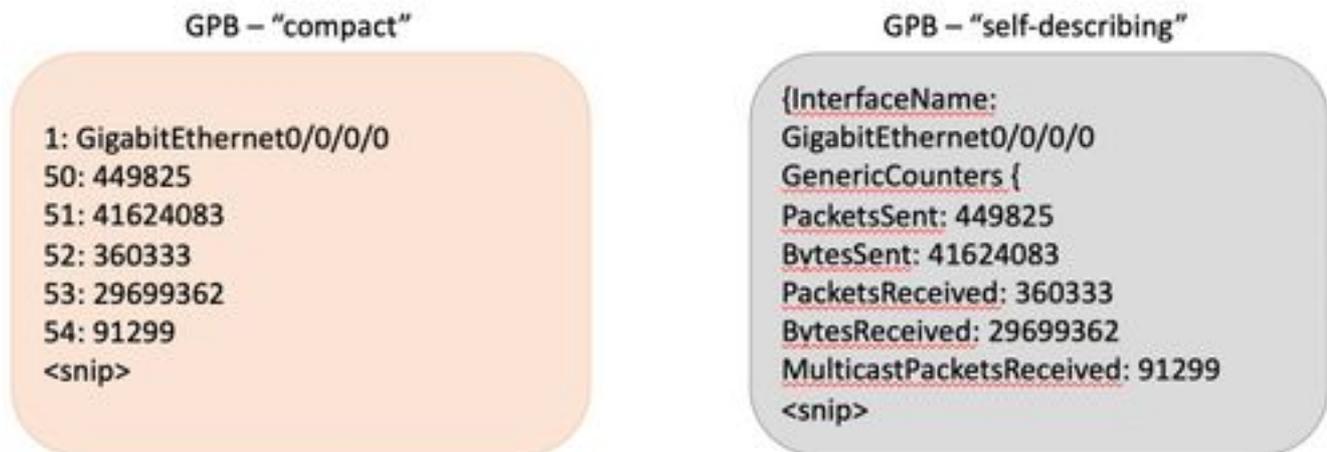


In den frühen Entwicklungsphasen der Telemetrie wurde XML aufgrund seiner tagbasierten Struktur zunächst als Format der ersten Wahl betrachtet. Das Problem bei XML war jedoch die nicht kompakte Codierungsstruktur. GPB (Google Protocol Buffers) wurde schließlich von Cisco eingeführt, da es die Effizienz und Geschwindigkeit von Verschlüsselungsvorgängen verbessert.

Es gibt zwei Varianten von GPB als Verschlüsselungsoptionen für Telemetrie-Streaming:

1. Kompakter GPB
2. Selbstbeschreibendes GPB

Der Hauptunterschied zwischen den beiden GPB-Telemetrieformaten besteht darin, wie sie die Schlüssel innerhalb eines Telemetriestroms darstellen und kodieren.



JSON ist ein weiteres benutzerfreundliches Codierungsschema, das sehr leicht zu verstehen ist und von fast allen Anwendungen decodiert werden kann.

Hinsichtlich der Bereitstellung gibt es nur wenige Vor- und Nachteile eines Codierungsschemas. Ein Vergleich der verschiedenen Codierungsschemata finden Sie im Abschnitt Telemetrie-Designrichtlinien.

## Verkehr

Telemetrie bietet drei mögliche Optionen für Transportprotokolle:

- TCP
- gRPC
- UDP

Telemetrie definiert auch zwei verschiedene Initiierungsmodi, um eine Sitzung zwischen dem Knoten und dem Collector zu starten:

- Dial-out
- Einwahl

Der Unterschied zwischen den beiden Verkehrsträgern besteht nur darin, wie die Transportsitzung aufgebaut wird.

Während der Wählsitzungen initiiert das Gerät die Verbindung, indem es ein SYN-Paket an den vorkonfigurierten Server-Port sendet. Nach dem Herstellen der Verbindung werden Datenströme sofort vom Gerät weggeschoben.

Bei Einwahlsitzungen hört der Router passiv einen TCP-Port ab, der auf eine Serververbindung wartet.

Sobald die Sitzung hergestellt ist, wird der Router jedoch nicht vom Server selbst abgefragt, da das Gerät weiterhin für Daten-Push-Vorgänge verantwortlich ist. In MDT existiert das Konzept der Datenabfrage noch nicht einmal.

TCP ist standardmäßig die vordefinierte Transportmethode für Telemetrie, da sie zuverlässig ist und sehr einfach als Option konfiguriert werden kann.

gRPC ist ein modernes Open-Source-Framework, das in jeder Umgebung ausgeführt werden kann. Sie basiert auf HTTP/2 und bietet eine Reihe erweiterter und umfassender Funktionen.

## Rhythmusbasierte Telemetrie und ereignisbasierte Telemetrie

Die Daten aus dem abonnierten Datensatz werden entweder in einem konfigurierten regelmäßigen Intervall oder nur bei Eintreten eines Ereignisses an das Ziel übertragen. Dieses Verhalten hängt davon ab, ob MDT für die Rhythmusbasierte Telemetrie oder die ereignisbasierte Telemetrie konfiguriert ist.

Die Konfiguration für ereignisbasierte Telemetrie ähnelt der für Rhythmusinformationen, wobei nur das Abtastrate-Intervall als Differenzierungsmerkmal verwendet wird. Wenn Sie den Wert für das Abtastintervall auf Null konfigurieren, wird das Abonnement für ereignisbasierte Telemetrie festgelegt, während Sie das Intervall auf einen beliebigen Wert ungleich null konfigurieren, wird das Abonnement für die Rhythmusbasierte Telemetrie festgelegt.

Es wird empfohlen, ereignisgesteuerte Telemetrie für änderungsbezogene Ereignisse zu verwenden.

## Telemetrie-Designrichtlinien

Wie bereits erläutert, umfasst der Telemetrie-Stack zahlreiche Komponenten. Hier einige Richtlinien, die bei der Implementierung der Telemetrie auf XR-Geräten beachtet werden sollten.

### So wählen Sie ein Codierungsschema aus

Wie bereits erwähnt, werden Daten (Objekte, Status) durch Codierung oder Serialisierung in ein Format übersetzt, das über das Netzwerk übertragen werden kann. Wenn der Empfänger die Daten decodiert oder deserialisiert, hat er eine semantisch identische Kopie der Originaldaten.

Verschiedene Kodierungsoptionen unterscheiden sich hinsichtlich der Leitungseffizienz und der Benutzerfreundlichkeit.

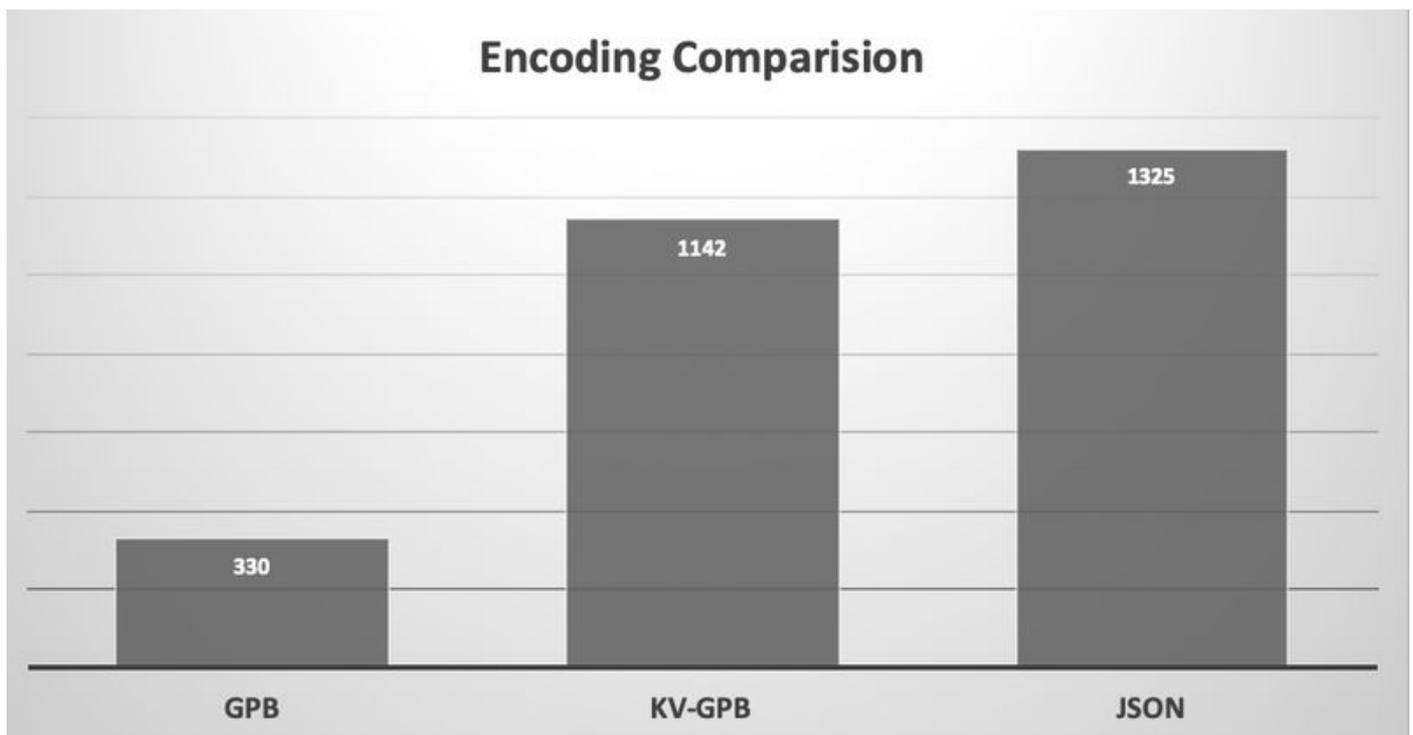
Kodierung	Kurzbeschreibung	Effizienz der Verkabelung	Sonstige Erwägung
GPB (kompakt)	Alle Binärdateien (Ausgenommen Werte, die Zeichenfolgen sind) 2x schneller, betrieblich komplexer (aber nicht im	Hoch	Profiles pro Modell

Vergleich zu SNMP)

GPB - KV (Schlüssel-Wert-Paar)	Zeichenfolgenschlüssel und Binärwerte (mit Ausnahme von Werten, bei denen es sich um Zeichenfolgen handelt) 3x größer, Native Modelle: benötigen weiterhin Heuristik für Schlüsselnamen	Mittel bis niedrig	Einzelne .proto-Datei Dekodierung
JSON	Alles Strings : Schlüssel und Werte	Niedrig	Freundlich. Für Menschen lesbar, anwendungsfreundlich einfach zu analysieren

GPB-KV bietet einen guten und ausgewogenen Mittelpunkt für das Kodierungsschema.

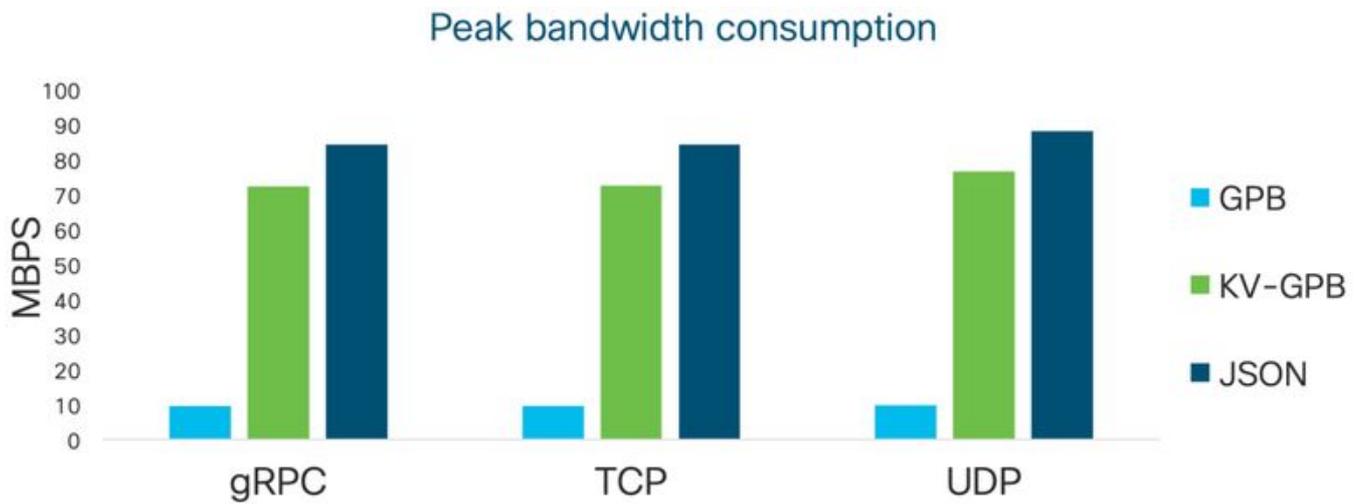
Bezüglich der Nachrichtenlänge für das ausgewählte Codierungsschema ist unten der Vergleich auf dem Kabel dargestellt.



Codierungsvergleich - Nachrichtenlänge in Byte

## Überlegungen zum Design des Transportnetzwerks

Unterschiedliche Kodierungsoptionen stellen unterschiedliche Bandbreitenanforderungen. Bei der Prüfung der Telemetrie muss der Netzbetreiber die erforderliche Bandbreitenbereitstellung entsprechend dem gewählten Kodierungsschema sicherstellen. Nur um eine faire Idee zu haben, Bandbreitennutzung pro Kodierungsschemavergleich.



Vergleich der Netzwerkbandbreite

Cisco empfiehlt die Verwendung von KV-GPB. Es ist ein guter Mittelweg zwischen Effizienz und Komfort.

## Evaluierung der Telemetrie-Konfigurationsoptionen

Bei der Konfiguration der modellbasierten Telemetrie muss der Bediener mit allen verschiedenen Komponenten vertraut sein, die für die Telemetrie erforderlich sind. Basierend auf den oben beschriebenen Optionen für Transport, Kodierung und Streaming-Richtung kann die Kombination ausgewählt werden, die für eine Umgebung am besten geeignet ist.

Die vier Hauptkomponenten sind:

1. Verkehr
2. Kodierung
3. Richtung der Sitzung
4. YANG-Datenmodelle

**Verkehr:** Wie angegeben, kann der Knoten Telemetriedaten entweder über TCP, UDP oder gRPC über HTTP/2 senden.

TCP ist aus Gründen der Einfachheit die bevorzugte Wahl, gRPC bietet jedoch optionale TLS-Funktionen, die als zusätzlicher Vorteil aus Sicherheitssicht in Betracht gezogen werden können.

**Kodierung:** Router können Telemetriedaten in zwei verschiedenen Varianten von Google Protocol Buffers bereitstellen: kompaktes und selbstbeschreibendes GPB.

Compact GPB ist die effizienteste Codierung, erfordert jedoch für jedes gestreamte YANG-Modell eine eindeutige .proto-Datei. Selbstbeschreibendes GPB ist weniger effizient, verwendet aber eine einzige .proto-Datei, um alle YANG-Modelle zu decodieren, da die Schlüssel als Strings in der .proto übergeben werden.

**Sitzungsrichtung:** Bei der Telemetrie-Bereitstellung gibt es zwei Optionen für die Sitzungsiniziation. Der Router kann mit dem Collector verbunden werden oder der Collector mit dem Router.

**YANG** ist der branchenweit akzeptierte Standard für die Datenmodellierung, und der Cisco

Programmability Stack verwendet ihn auch, um strukturierte Datensätze zu erstellen, die so schnell wie möglich über das Netzwerk codiert und übertragen werden können.

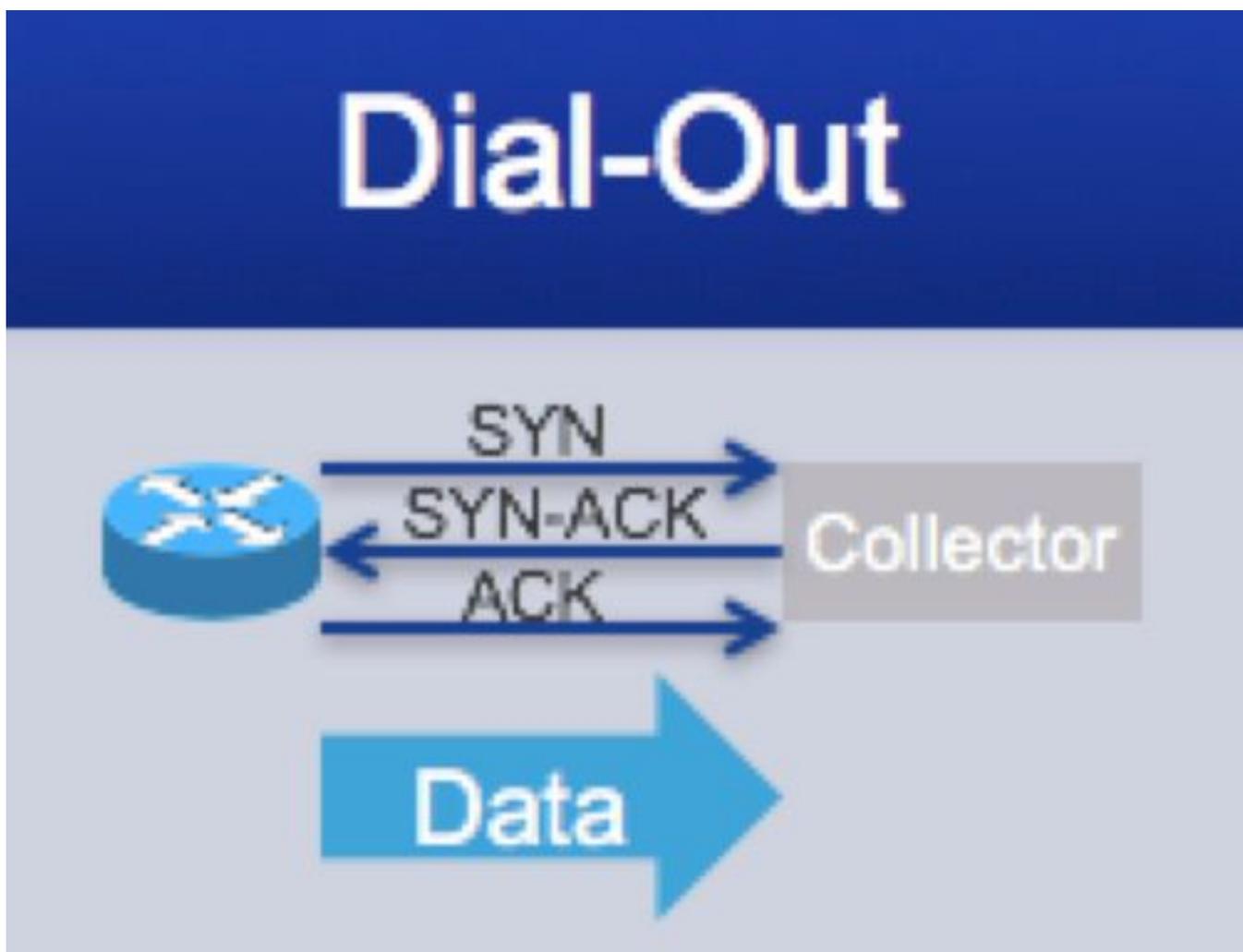
Diese Datenmodelle bilden zusammen mit den oben beschriebenen spezifischen Verschlüsselungsformaten und Transportprotokollen eine vollständige Lösung für Analytics.

## Telemetrie-Konfigurationsbeispiele

### IOS-XR

#### Untergliederung der DFÜ-Konfiguration

Im Dial-Out-Modus muss der Router eine TCP-Sitzung zum Collector initiieren und die von der Sensorgruppe im Abonnement angegebenen Daten senden.



Vom Standpunkt der Konfiguration aus ist die Telemetrie-Konfiguration ein dreistufiger Prozess. Zunächst identifizieren wir die Informationen, die wir streamen möchten, und erfassen sie unter der Konfiguration der Sensorgruppe. Zweitens identifizieren wir das Ziel, zu dem die Informationen gestreamt werden müssen, und erfassen es in der Zielgruppenkonfiguration. Drittens verwenden wir die Informationen, die in den beiden vorherigen Schritten identifiziert wurden, um das tatsächliche Abonnement zu konfigurieren.

1. Sensorgruppen definieren
2. Zielgruppen definieren

### 3. Abonnement definieren

#### Sensorgruppen definieren

Die Konfiguration der Sensorgruppe identifiziert die Informationen, die per Streaming übertragen werden sollen. Die nachfolgende Konfigurationsvorlage enthält die erforderliche Konfiguration zum Konfigurieren von Sensorgruppen.

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

Das nachfolgende Beispiel zeigt ein Beispiel aus der Router-CLI, in dem die Konfiguration der Sensorgruppe veranschaulicht wird:

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

Es können mehrere Sensorpfade als Teil derselben SensorGroup-Definition vorhanden sein:

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

#### Zielgruppen definieren

Die Zielgruppenkonfiguration gibt das Ziel an, an das die Informationen übertragen werden sollen.

Es verfügt über drei Hauptparameter

1. Richtung der Sitzung
2. Zu verwendende Kodierung
3. Zu verwendendes Transportprotokoll

Hier ein Beispiel:

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

## Abonnement definieren

Das Abonnement verknüpft die Informationen zur Sensorgruppe und zur Zielgruppe als letzten Teil der Konfiguration. Das Beispielintervall wird als Teil des Abonnements definiert.

Hier ein Beispiel:

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

## Vollständiges Konfigurationsbeispiel

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

## Vorteile bei der Hinauswahl

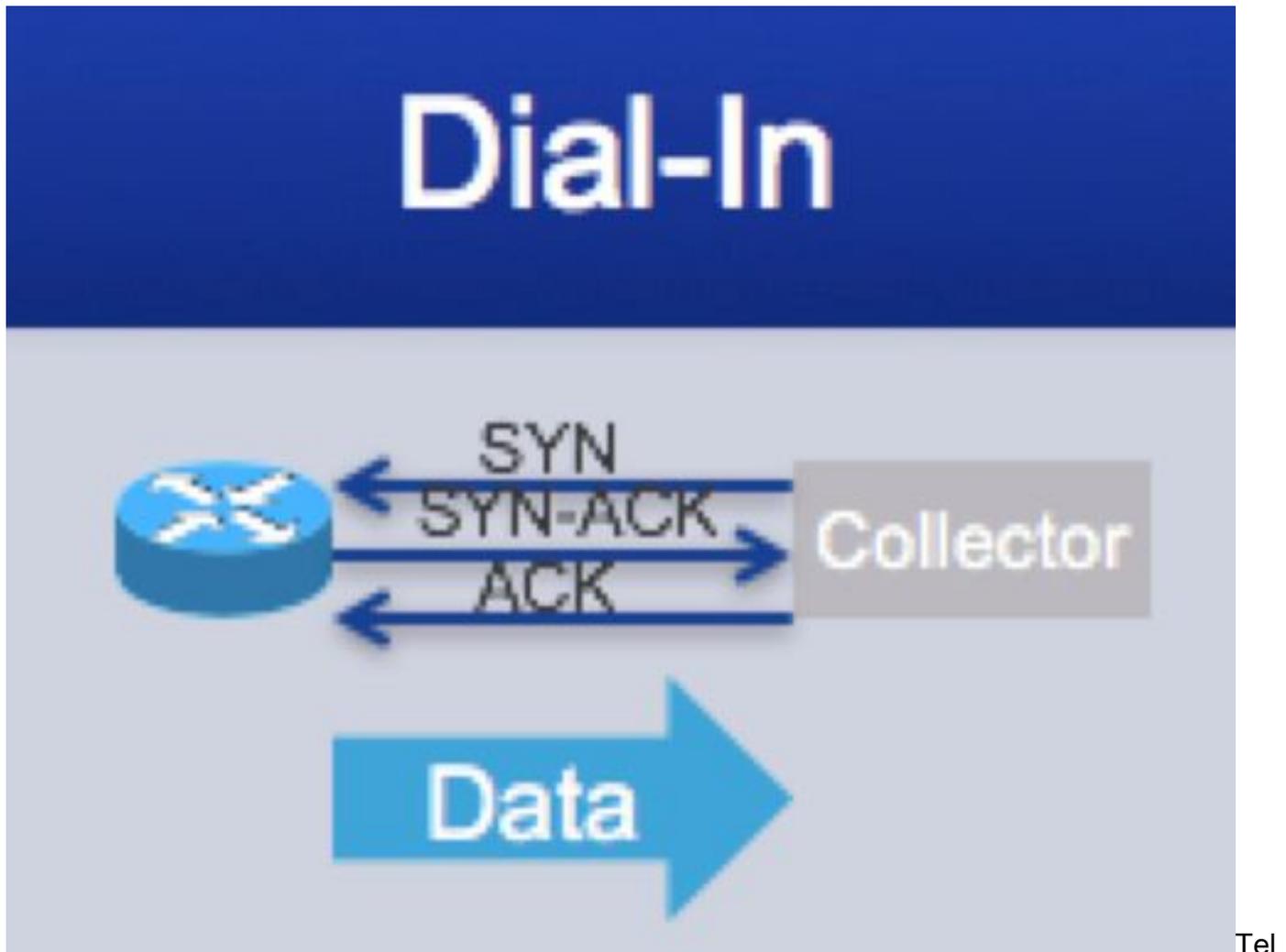
- Größere Flexibilität bei den Transportoptionen.
- Es müssen keine Ports für eingehenden Management-Datenverkehr geöffnet werden.

- Anycast und Lastenausgleich.

## Untergliederung der Einwahlkonfiguration

Im Einwahlmodus meldet sich ein MDT-Collector/Receiver/Orchestrator am Router an und meldet sich dynamisch für einen oder mehrere Sensorpfade oder Abonnements an. Der Router fungiert als Server und der Client als Empfänger.

Es wird nur eine Sitzung gebildet, und der Router streamt Telemetriedaten über dieselbe Sitzung. Dieses dynamische Abonnement wird beendet, wenn der Empfänger das Abonnement kündigt oder die Sitzung beendet wird.



emetrie-Einwahl

Da sich der Collector beim Router einwählt, muss in der Konfiguration nicht jedes MDT-Ziel angegeben werden. Aktivieren Sie einfach den gRPC-Dienst auf dem Router, schließen Sie Ihren Client an, und aktivieren Sie dynamisch das gewünschte Telemetrie-Abonnement.

Aus konfiguratorischer Sicht ist die Telemetrie-Konfiguration ein dreistufiger Prozess, ähnlich dem oben beschriebenen. Zunächst müssen wir gRPC aktivieren. Zweitens identifizieren wir das Ziel, zu dem die Informationen gestreamt werden müssen, und erfassen sie in der Konfiguration der Sensorgruppe. Drittens verwenden wir die Informationen, die in den beiden vorherigen Schritten identifiziert wurden, um das tatsächliche Abonnement zu konfigurieren.

1. gRPC aktivieren

2. Sensorgruppen definieren

3. Abonnement definieren

[Zum Erweitern klicken](#)

Der Einwahlmodus wird nur mit gRPC unterstützt.

Der Einwahlmodus wird nur mit gRPC unterstützt.

## gRPC aktivieren

Zunächst muss der gRPC-Server auf dem Router aktiviert werden, damit eingehende Verbindungen vom Collector akzeptiert werden.

[Zum Erweitern klicken](#)

- Der Bereich <Portnummer> liegt zwischen 57344 und 57999. Wenn eine Portnummer nicht verfügbar ist, wird ein Fehler angezeigt.

Der Bereich <Portnummer> liegt zwischen 57344 und 57999. Wenn eine Portnummer nicht verfügbar ist, wird ein Fehler angezeigt.

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

## Sensorgruppen definieren

Die Konfiguration der Sensorgruppe identifiziert die Informationen, die per Streaming übertragen werden sollen. Die nachfolgende Konfigurationsvorlage enthält die erforderliche Konfiguration zum Konfigurieren von Sensorgruppen.

Das nachfolgende Beispiel zeigt ein Beispiel aus der Router-CLI, wobei es sich um ein Beispiel für die Konfiguration einer Sensorgruppe handelt.

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

## Abonnement definieren

Das Abonnement bindet die Sensorgruppe und den gRPC als letzten Teil der Konfiguration zusammen. Das Beispielintervall wird als Teil des Abonnements definiert.

Die nachfolgende Konfigurationsvorlage enthält die zum Konfigurieren des Abonnements erforderliche Konfiguration.

Das folgende Beispiel zeigt ein Beispiel aus der Router-CLI, in dem Sie ein Abonnement erstellen,

die Sensorgruppe und die Zielgruppe zusammenbinden und die Abtastrate definieren.

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

## Vollständige Konfigurationsvorlage und Beispiel

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

## Vorteile der Einwahl

- Ein Kanal für Konfiguration und Streaming
- Überwachungsport am Router/Gerät
- Übergangsverbindung
- Derzeit nur gRPC/gNMI verfügbar

## Ereignisgesteuerte Telemetrie

Bei der ereignisgesteuerten Telemetrie werden die Daten des abonnierten Datensatzes nur dann per Streaming übertragen, wenn ein Ereignis eintritt.

## Ereignisgesteuerte Telemetriekonfiguration

Die Konfiguration für ereignisbasierte Telemetrie ähnelt der für Rhythmustelemetrie. Der einzige Unterschied bei der Konfiguration von ereignisbasierter Telemetrie besteht in der Konfiguration des Abtastintervalls. Wenn Sie den Wert für das Abtastintervall auf Null konfigurieren, wird das Abonnement für ereignisbasierte Telemetrie festgelegt.

## Vollständige Konfigurationsvorlage und Beispiel für DIAL-OUT

```
RP/0/RP0/CPU0:XR#
RP/0/RP0/CPU0:XR#conf
RP/0/RP0/CPU0:XR(config)#
RP/0/RP0/CPU0:XR(config)#telemetry model-driven
RP/0/RP0/CPU0:XR(config-model-driven)#sensor-group <Sensor-Group-Name>
```

```

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

```

Das nachfolgende Beispiel zeigt ein Beispiel aus der Router-CLI.

```

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

```

## Vollständige Konfigurationsvorlage und Beispiel für DIAL-IN

```

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

```

Das nachfolgende Beispiel zeigt ein Beispiel aus der Router-CLI.

```

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

```

```
RP/0/RP0/CPU0:XR(config-model-driven-sub)# commit
RP/0/RP0/CPU0:XR(config-model-driven-sub)# end
RP/0/RP0/CPU0:XR#
```

## Telemetrie mit SHOW-Befehlen validieren

Vom Router-Standpunkt aus können wir die für jede Sensorgruppe, Zielgruppe und Abonnement konfigurierten Parameter überprüfen.

```
// ALL CONFIGURED SUBSCRIPTIONS
```

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

```
Subscription: Subscription101          State: ACTIVE
-----
Sensor groups:
Id          Interval (ms)      State
SensorGroup101  30000             Resolved

Destination Groups:
Id          Encoding          Transport  State  Port  IP
DestGroup101  self-describing-gpb tcp        Active  5432  172.16.128.3
```

```
// DETAILS ON A PARTICULAR SUBSCRIPTION
```

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

```
Subscription: Subscription101
-----
State:          ACTIVE
Sensor groups:
Id: SensorGroup101
  Sample Interval:      30000 ms
  Sensor Path:          Cisco-IOS-XR-infra-statsd-oper:infra-
statistics/interfaces/interface/latest/generic-counters
  Sensor Path State:    Resolved

Destination Groups:
Group Id: DestGroup101
  Destination IP:       172.16.128.3
  Destination Port:     5432
  Encoding:             self-describing-gpb
  Transport:            tcp
  State:                Active
  Total bytes sent:     4893
  Total packets sent:   1
  Last Sent time:      2019-11-01 10:04:11.2378949664 +0000

Collection Groups:
-----
Id: 1
  Sample Interval:      30000 ms
  Encoding:             self-describing-gpb
  Num of collection:    5
  Collection time:      Min:      6 ms Max:      29 ms
  Total time:          Min:      6 ms Avg:     12 ms Max:     29 ms
  Total Deferred:      0
  Total Send Errors:   0
  Total Send Drops:    0
  Total Other Errors:  0
```

Last Collection Start:2019-11-01 10:06:11.2499000664 +0000  
Last Collection End: 2019-11-01 10:06:11.2499000664 +0000  
Sensor Path: Cisco-IOS-XR-infra-statsd-oper:infra-  
statistics/interfaces/interface/latest/generic-counters

RP/0/RP0/CPU0:XR#

// ALL CONFIGURED DESTINATIONS

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

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

RP/0/RP0/CPU0:XR#

// PARTICULAR DESTINATION

RP/0/RP0/CPU0:XR#show telemetry model-driven destination DestGroup101

Destination Group: **DestGroup101**

-----

Destination IP: 172.16.128.3  
Destination Port: 5432  
State: Active  
Encoding: self-describing-gpb  
Transport: tcp  
**Total bytes sent: 83181**  
**Total packets sent: 17**  
Last Sent time: 2019-11-01 10:12:11.2859133664 +0000

Collection Groups:

-----

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

RP/0/RP0/CPU0:XR#

// ALL CONFIGURED SENSOR GROUPS

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

Sensor Group Id:**SensorGroup101**

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

// PARTICULAR SENSOR GROUPS

RP/0/RP0/CPU0:XR#show telemetry model-driven sensor-group SensorGroup101

Sensor Group Id:**SensorGroup101**

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

RP/0/RP0/CPU0:XR#

# Telemetrie-Erfassungstapel

Neben der Router-Konfiguration waren für eine Telemetrie-basierte Lösung verschiedene Komponenten wie Collector, Datenbank und Monitoring-/Analysesoftware erforderlich. Diese Komponenten können entweder separat konfiguriert werden oder Teil eines umfassenden Einzelprodukts sein.

- Ein Decoder sollte installiert werden, um eingehende Pakete zu übernehmen und zur weiteren Speicherung weiterzuleiten.
- Eine Zeitreihendatenbank (TSDB) ist erforderlich, um gestreamte Informationen zu speichern.
- Ein grafisches Tool wird auch benötigt, um Daten aus der internen Datenbank zu visualisieren.

Eine detaillierte Beschreibung des Auflistungspakets geht über den Rahmen hinaus. Cisco Crossworks Health Insights ermöglicht die Telemetrie ohne Benutzereingriffe. Dabei werden Geräte automatisch mit Telemetrikonfiguration und Tabellen/Schemata in einer TSDB (Time Series Database) erstellt. Der Betriebs- und Netzwerkmanagement-Aufwand für die Erfassung und Bereinigung von Daten wird optimiert, sodass sich die Betreiber auf ihre Geschäftsziele konzentrieren können. Durch die Verwendung eines gemeinsamen Collectors zur Erfassung von Netzwerkgerätedaten über SNMP, CLI und modellgestützte Telemetrie können Datenduplikate vermieden und Geräte sowie das Netzwerk entlastet werden.

## Überlegungen zur Bereitstellung von Telemetrie in einem Netzwerk

Bei der Analyse der Telemetrie-Bereitstellung in einem Netzwerk müssen verschiedene Aspekte berücksichtigt werden.

### Skalierung

Telemetrie kann große Datenmengen übertragen. Daher wird eine sorgfältige Prüfung der Skalierbarkeit empfohlen.

### Nur die erforderlichen Daten streamen

Jedes Yang-Modell wird mehrere Leaf-Knoten haben. Es wird empfohlen, die erforderlichen und nicht erforderlichen Informationen genau anzugeben. Es wird empfohlen, die Yang-Modelle zu untersuchen und den Datenpfad zu identifizieren, der für die Telemetrie-Anwendungsfälle erforderlich ist.

### Die Menge der Streaming-Daten berücksichtigen

Für die Gesamtzahl der gestreamten Telemetriedaten müssen folgende Punkte berücksichtigt werden:

1. Bandbreitenzuweisung im Netzwerk
2. QoS
3. Leistung zusätzlicher Anwendungen/Software wie Collector-SoftwareDatenbank der ZeitreiheAnalyse-/Visualisierungssoftware

4. Die Encoding-Effizienz (im Abschnitt "Telemetrie-Designrichtlinien" beschrieben) wirkt sich direkt auf die Menge der gestreamten Daten aus. Ein kompaktes GPB wird nach Möglichkeit empfohlen.
5. Das Erfassungsintervall wirkt sich direkt auf mehrere Aspekte aus, einschließlich, jedoch nicht beschränkt auf die folgenden. Bandbreitennutzung im NetzwerkSpeicheranforderung in der Datenbankleistung des Geräts, das Daten streamt

Es wird empfohlen, die Häufigkeit der Erfassung anhand der Anwendungsanforderungen zu evaluieren.

Insgesamt wird empfohlen, das Filtern unerwünschter Daten an der Quelle oder am Ziel als machbar zu betrachten. Wir haben die Möglichkeit, unerwünschte Daten zu filtern. Die Filterung kann auf zwei Ebenen durchgeführt werden: -

1. Bei der Quelle: Die Geräte, die die Daten streamen.
2. Am Ziel - Der Collector sammelt und normalisiert Daten. (Die Filterung am Collector geht über den Umfang dieses Dokuments hinaus.)

Das folgende Beispiel zeigt, dass Daten nur für Hundered Gig-Schnittstellen innerhalb der Sensorpfade gefiltert werden, indem Platzhalter verwendet werden.

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

## Referenzen

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

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

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

## Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.