ASR9K型號驅動遙測白皮書

目錄

簡介

對象

本白皮書旨在幫助客戶快速瞭解一般的「模型驅動遙測」(MDT)功能以及如何在聚合服務路由器9000(ASR9K)中實施該功能，其中包括一些設計手冊和配置詳細資訊。還包括一些部署注意事項，這對於使用ASR9K部署此功能很有幫助。總之，本白皮書可作為任何使用此功能的工作者的快速參考指南。

雖然遙測作為通用功能引入，但重點是ASR9K實施；也就是說，並非其他cisco平台支援的所有功能都受ASR9K平台支援，某些功能實施可能特定於ASR9K。

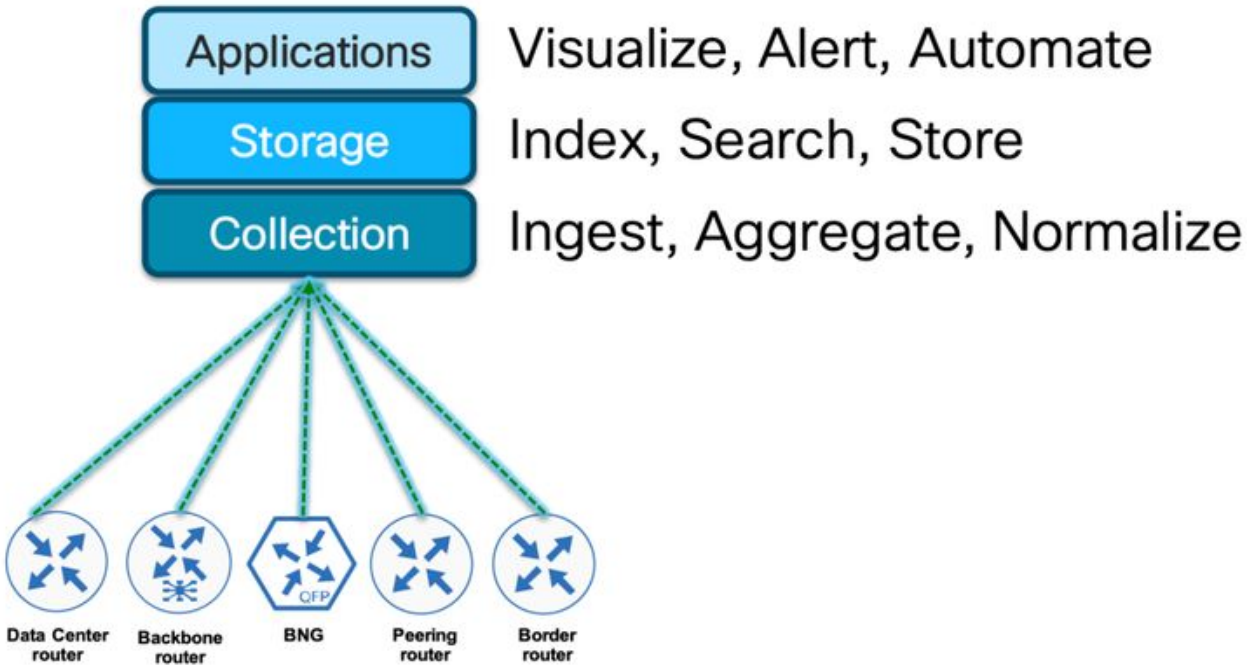
遙測技術簡介

首先，簡單地說，遙測是收集有用運算元**據的過程**。根據維基百科，遙測是一個自動的通訊過程，通過它可以在遙遠或無法到達的點收集測量資料和其他資料，並將資料傳送到接收裝置以進行監控。遙測詞本身源自希臘語根：tele = remote，metron = measure。

對於網路管理，網路運營商長期以來一直依賴簡單網路管理協定(SNMP)。雖然SNMP被廣泛用於網路監控，但它從未用於配置，儘管使用SNMP進行配置的功能始終存在。操作員已編寫自動化指令碼來處理日常配置任務，但指令碼對於這類任務具有挑戰性，並且難以管理。

因此，操作員開始轉向資料模型驅動管理。網路配置基於由netconf等協定推送的YANG資料模型。現在，僅推送配置並不意味著已配置的服務正在運行，因此必須存在一種機制，能夠在配置同時監視服務運算元據。這是oper資料模型的地方；Telemetry使用哪些資訊將資訊推出裝置；有幫助。因此，配置是以YANG資料模型驅動的，所以也必須對服務進行驗證；對於遙測來說，為了具有相同的對象語義。因此，該術語稱為**模型驅動遙測**或流遙測。

cXR（32位IOS XR）自6.1.1版開始引入模型驅動遙測(MDT)，可近即時收集和測量重要資料，為現代網路的大多數運行問題提供快速答案。



高級遙測體系結構

MDT利用網路裝置支援的結構化資料模型，並提供在這些資料模型中定義的關鍵資料。**由於**從網路收集的資料基於標準，並且在整個供應商實施中是統一的，因此遙測可幫助客戶使用一個通用網路管理系統、流程和應用來管理其多供應商網**絡**。

而不是等待從集中管理站（通常是SNMP NMS）檢索（提取）資料；使用MDT，網路裝置主動傳送(**推**)與網路重要功能有關的效能資料，例如資料包轉發資訊、錯誤統計、系統狀態、CPU和記憶體資源等。

為什麼選擇遙測

為了分析和故障排除而收集資料一直是監控網路運行狀況的一個重要方面。有多種機制（例如SNMP、CLI和Syslog）可從網路收集資料。雖然這些方法為網路服務了很長時間，但並不適用於對自動化的要求很高的現代網路，但大規模服務是基本需求。網路運行狀況資訊、流量統計資訊和關鍵基礎設施資訊被傳送到NMS中的遠端站，它們用於增強操作效能並縮短故障排除時間。客戶端輪詢所有網路節點的SNMP等拉取模式效率低下。當要輪詢的客戶端數量較多時，網路節點上的處理負載會增加。相反，推模式能夠連續將資料從網路流出，並通知客戶端。遙測啟用推送模型，提供對監控資料的近即時訪問。

流遙測提供了一種機制，用於從路由器中選擇感興趣的資料，並以標準格式將其傳輸到遠端管理站進行監控。該機制實現了基於即時資料的網路微調，這對網路的無縫運行至關重要。更精細的粒度和更高的遙測資料頻率可實現更好的效能監控，從而可更好地進行故障排除。

它有助於提高網路中服務效率的頻寬利用率、鏈路利用率、風險評估和可擴充性。有了流遙測技術，網路運營商就可以獲得更多接近即時的資料，這有助於改善決策。

從SNMP轉移的需要

SNMP已經存在了三十年，其運行方式沒有改變以滿足現代網路的監控需求。真正的問題在於SNMP的執行速度。

SNMP帶來的三個主要挑戰實際上是其基本操作行為的一部分，因此SNMP提供甚少/沒有改進的空間，遙測技術本身就解決了這三個挑戰。

- **執行速度和即時監控需求**

SNMP使用PULL Model - GetBulk / GetNext操作，這些操作通過從一個列遍歷到另一個列以線性方式工作。此外，如果大型表不能容納在一個資料包中，則需要多個請求。 這是導致SNMP速度減慢的最大瓶頸，而且傳送的資料通常以分鐘為單位超過某個時間因素。這種延遲是現代網路監控要求所無法接受的。

MDT（模型驅動遙測）使用PUSH模型，並且固有不受上面列出的限制，因為它知道以什麼時間間隔向誰傳送什麼資料。它只需一次查詢即可收集資料，並使用預構建的內部模板實現超高速的內部操作，因而能夠以極少的時間交付更多資料。

- **額外的開銷和缺少最佳化選項**

由SNMP提取的資料實際上是以內部資料結構的形式儲存的，需要由節點進行內部轉換。這是網路節點將內部資料結構對映為SNMP格式的幕後額外工作。執行了一些內部最佳化，但是這些最佳化仍然不夠。

另一方面，Telemetry直接提取內部資料結構，並且在將資料傳送出去之前執行最小處理，從而以儘可能最少的時間和精力提供最更新的資料。

- **工作負載的線性性質**

即使我們在同一時間輪詢相同的準確資料，每個額外的輪詢站都會導致節點上的負載增加。 從多個輪詢站並行訪問同一MIB可能導致響應速度較慢和CPU利用率較高。這一點在大型表的情況下尤其明顯，在大型表中，多個站點訪問同一MIB表的不同部分。

另一方面，如果多個目的地需要相同的資料，則遙測需要一次提取資料並複製資料包。 推式模式在速度和規模方面勝過SNMP拉式模式。

使用MDT時，資料收集的方法及其基本原則在下表中列出，並與SNMP技術要點進行比較。

簡易網路管理通訊協定(SNMP)	**模型驅動遙測(MDT)**
非即時資訊	即時資訊
可擴充性差	高度可擴展
拉模	Push-Model
非自動化	自動化就緒/資料模型驅動

流遙測的優點

流式即時遙測資料在以下方面非常有用：

容量規劃/流量優化：當網路中經常監控頻寬利用率和資料包丟棄時，更易於新增或刪除鏈路、重定向流量、修改策略等。使用快速重新路由等技術，網路可以切換到新路徑，並且重新路由的速度比SNMP輪詢間隔機制更快。流傳輸遙測資料有助於為更快的流量提供快速響應時間。

更好的可視性：幫助快速檢測和避免網路中出現問題情況後出現的故障情況。

型號驅動遙測技術規範

下一節將介紹IOS XR型號驅動遙測技術功能及主要元件（也稱為MDT）。

遙測功能

遙測框架被組織成三個獨立且相互關聯的功能塊。

第一個模組是關於數**據表示的**，資料表示是資訊參考分析或測量在板上的組織方式。

第二個塊關於**編碼**。每個取樣間隔，遙測都會將以上測量資料轉換為可線上中進行序列化的格式。當然，另一端的控制器必須能夠解碼資料，以便擁有裝置傳送的原始資料的相同副本。

最後一塊是關於**運輸**的這是用於在裝置之間傳輸資料的協定棧。

下表彙總了模型驅動遙測構建塊的主要結構：

功能	**元件**
資料表示	YANG資料模型
編碼	GPB/GPB自我描述
傳輸	TCP/gRPC

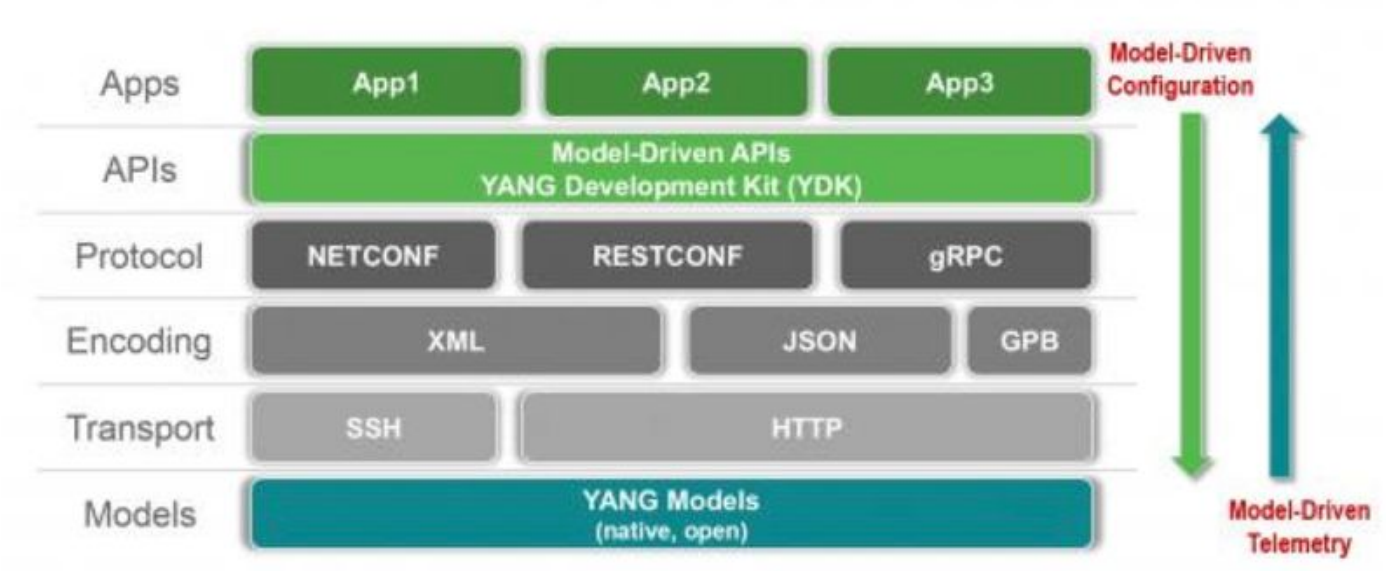
表 3 遙測構建塊

遙測元件

在瞭解遙測和基礎配置部分的工作方式之前，瞭解遙測的不同元件非常重要，以便評估最佳設定。遙測依賴於IOS XR可程式設計性堆疊，在該堆疊中，新的基礎架構框架提供了網路自動化的基本功能。

YANG最近成為資料建模的標準，思科可程式設計性堆疊使用此標準形成結構化資料集，該資料集可編碼並在網路上儘可能快地傳輸。YANG的靈活性為同時用作自動化流程的配置工具帶來了巨大優勢。這些資料模型結合特定的編碼格式和傳輸協定，使MDT成為網路分析的完整解決方案。

對於模型驅動遙測設定，YANG資料模型將成為關鍵元件，以便為收集和分析提供必需的資料流。



IOS XR可程式性堆疊

楊

Yang被定義為「資料建模語言，用於為網路管理協定對配置資料、狀態資料和通知進行建模。」由於YANG與典型程式語言體系結構的分離性，它可以與多種工具進行互動。

YANG建模資料結構是圍繞模組和子模組的概念構建的，它以樹狀的方式定義資料的層次，可用於多種操作，包括配置操作和通知處理。

YANG模型有多種來源可供使用，其中以下三種來源被認為是主要來源：

- 原生型號/思科特定
- OpenConfig
- IETF

思科專用型號：這些也稱為本**機型號**，由包括思科在內的各種裝置供應商發佈。 例如Cisco-IOS-XR-ptp-oper.yang

OpenConfig型號： OpenConfig是一個非正式的網路運營商工作組。OpenConfig定義了所有供應商都應當支援的通用YANG模型，以便配置關鍵任務功能。 例如openconfig-interfaces.yang

IETF型號:IETF還定義了一些常見YANG模組，這些模組描述了介面、QOS的基本配置，並定義了其他常見的資料型別（如Ipv4、IPv6等）。 例如ietf-syslog-types.yang

思科支援可用的Openconfg型號。 供應商正在趨向於採用標準化的資料建模方法，以支援多供應商環境。

陽模**型有**三種型別：

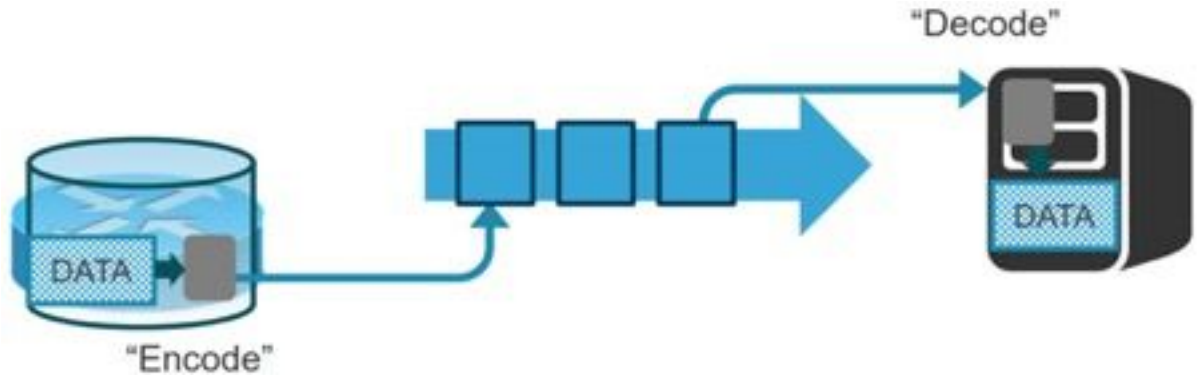
1. 操作
2. 組態
3. 動作

遙測只關**於可識別**為*-oper-*.yang的**Yang操作模型。**

YANG定義在RFC 7950中：https://tools.ietf.org/html/rfc7950。

編碼

編碼（或「序列化」）將資料（對象、狀態）轉換為可通過網路傳輸的格式。當接收器對資料進行解碼（「反序列化」）時，它擁有原始資料的語義相同的副本。



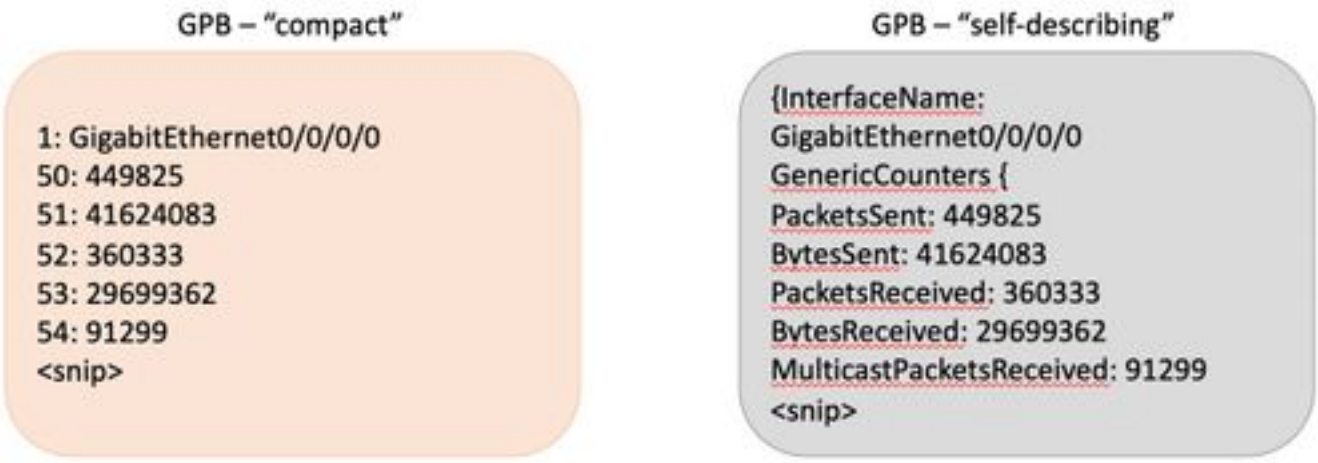
在遙測的早期開發階段，XML由於其基於標籤的結構而最初被認為是首選的編碼格式。然而，XML的問題在於其非緊湊編碼結構。GPB(Google Protocol Buffers)最終被思科採用，因為它提高

了編碼操作的效率和速度。

GPB有兩種形式作為遙測流的編碼選項：

1. 精簡型GPB
2. 自描述GPB

兩種GPB遙測格式之間的主要區別在於它們如何表示和編碼遙測資料流中的金鑰。



JSON是另一個人類友好的編碼模式，非常容易理解，幾乎任何應用程式都能解碼。

從部署角度看，編碼方案的利弊很少。在遙測設計手冊一節中給出了各種編碼方案的比較。

傳輸

遙測為傳輸協定提供了三種可能的選擇：

- TCP
- gRPC
- UDP

遙測還定義了兩種不同的啟動模式，以便啟動節點和收集器之間的會話：

- 撥出
- 撥入

兩種模式之間的區別僅在於傳輸會話的建立方式。

在撥出會話期間，裝置通過向預配置的伺服器埠傳送syn資料包來啟動連線。建立連線後，資料流會立即從裝置推出。

對於撥入作業階段，路由器會被動偵聽等待伺服器連線的tcp連線埠。

但是，建立會話後，路由器不會由伺服器本身輪詢，因為裝置仍負責資料推送操作。在MDT中，資料輪詢的概念甚至不存在。

預設情況下，TCP是遙測預定義的傳輸方法，因為它是可靠的，而且非常容易配置為選項。

gRPC是一個現代開源框架，設計為可在任何環境中運行。它構建在HTTP/2之上，並提供一組增強

和豐富的功能。

基於節奏的遙測與基於事件的遙測

來自預訂的資料集的資料以配置的週期性間隔或者僅在發生事件時才被流到目的地。此行為取決於MDT是配置為基於節奏的遙測還是基於事件的遙測。

基於事件的遙測的配置與基於節奏的遙測類似，僅使用樣本間隔作為區分因素。將取樣間隔值配置為零將設定對基於事件的遙測的訂閱，而將間隔配置為任何非零值將設定對基於事件的遙測的訂閱。

建議對更改相關事件使用事件驅動遙測。

遙測設計手冊

如前所述，遙測堆疊中有許多元件，下面是一些在XR裝置上實施遙測時應考慮的准則。

如何選擇編碼方案

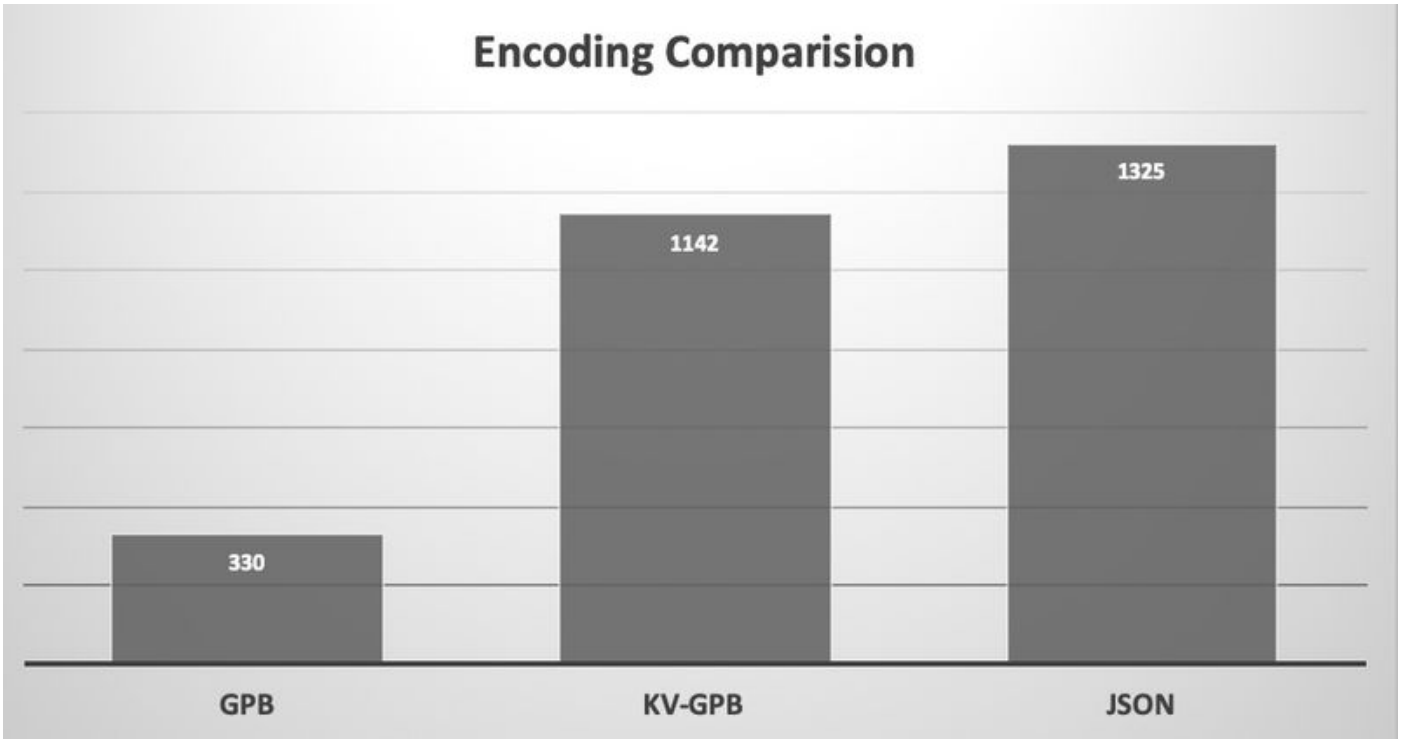
如上所述，編碼或序列化將資料（對象、狀態）轉換為可通過網路傳輸的格式。當接收器對資料進行解碼或反序列化時，它擁有原始資料的語義相同的副本。

各種編碼選項在電線效率和易用性方面各不相同。

編碼	簡要說明	線上的效率	其他考慮因素
GPB（緊湊型）	萬物二進位 （除了作為字串的值） 速度提高2倍，操作更複雜 （但與SNMP無關）	高	每個型號的Proto檔案
GPB - KV（金鑰值對）	字串鍵和二進位制值 （除了字串值） 3倍大， 原生模型：仍需要關鍵名稱的啟發式分析	中到低	單個用於煎煮的.proto
JSON	所有字串：鍵和值	低	很友好。易於閱讀、應 式友好且易於分析

GPB-KV為編碼模式提供了一個良好的平衡中點。

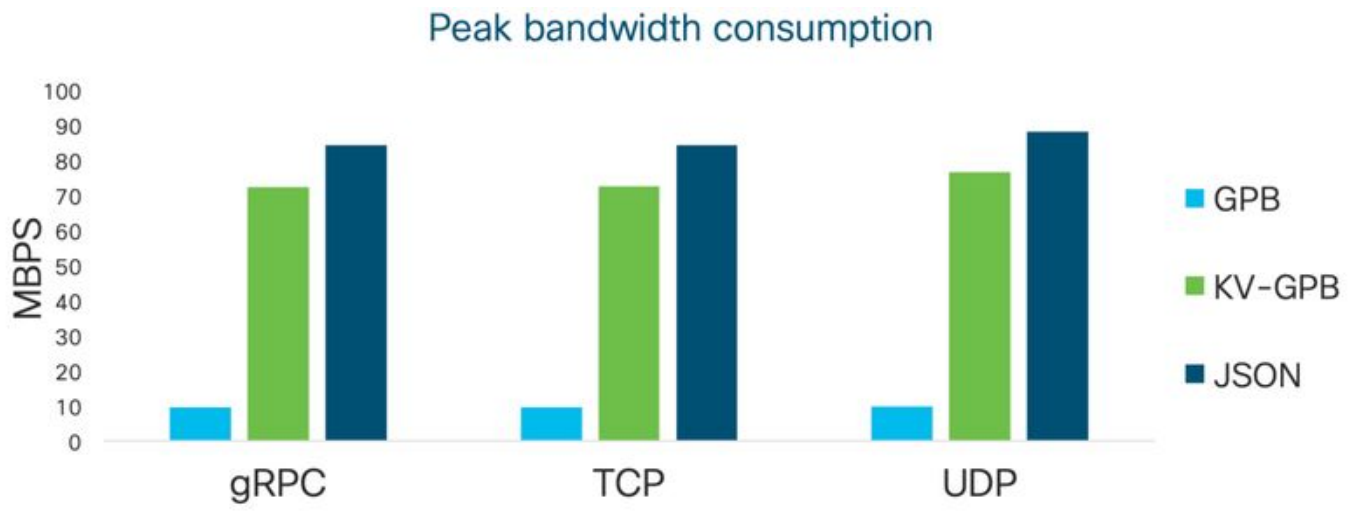
關於所選編碼方案的消息長度，下面是線路上的比較。



編碼比較 — 消息長度（以位元組為單位）

傳輸網路設計注意事項

不同的編碼選項對頻寬的要求不同。在考慮遙測時，網路運營商需要根據選擇的編碼方案進行充分的頻寬調配。只需有一個合理的想法，即低於每個編碼模式所消耗的頻寬。



網路頻寬比較

思科建議使用KV-GPB。它是效率與便利之間的良好中點。

評估遙測配置選項

配置模型驅動遙測時，操作員應瞭解遙測涉及的所有不同元件。基於上述可用於傳輸、編碼和流方向的選項，您可以進一步選擇更適合環境的組合。

四個關鍵元件是

1. 傳輸

2. 編碼
3. 作業階段方向
4. YANG資料模型

傳輸： 如上所述，節點可以使用TCP、UDP或gRPC over HTTP/2傳輸遙測資料。

雖然TCP是簡便性的首選方案，但gRPC提供可選的TLS功能，從安全形度考慮，該功能可視為一項額外優勢。

編碼： 路由器可以交付兩種不同型別的Google協定緩衝區的遙測資料：緊湊和自描述GPB。

精簡型GPB是最有效的編碼，但每個流傳輸的YANG模型都需要一個唯一的.proto。自描述GPB效率較低，但它使用單個.proto檔案來解碼所有YANG模型，因為這些金鑰在.proto中作為字串傳遞。

會話方向： 在遙測部署中啟動會話有兩種選項。路由器可以「撥出」到收集器，或收集器可以「撥入」到路由器。

YANG是業界公認的資料建模標準，思科可程式設計性堆疊還使用該標準形成結構化資料集，該資料集可通過網路儘快編碼和攜帶。

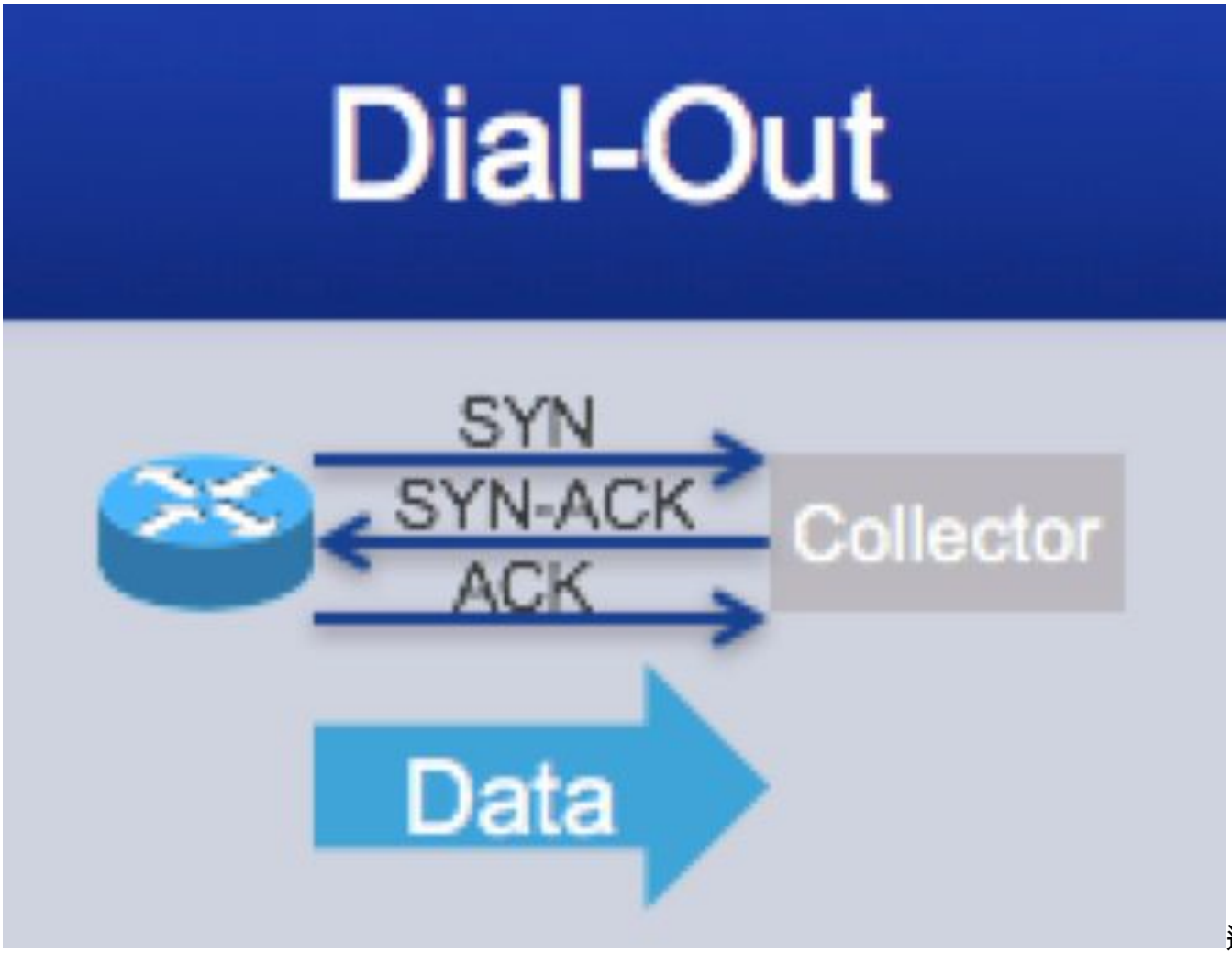
這些資料模型加上上文討論的特定編碼格式和傳輸協定，使模型驅動遙測(MDT)成為分析的完整解決方案。

遙測配置示例

IOS-XR

撥出配置中斷

在撥出模式下，路由器負責向收集器發起TCP會話，並傳送訂閱中感測器組指定的資料。



遙測撥出從配置的角度來看，遙測配置是一個三步過程。首先，在感測器組配置下，識別並捕獲要傳輸的資訊。其次，我們確定資訊必須流式傳輸到的目標，並在目標組配置下捕獲該資訊。第三，使用前兩步中確定的資訊配置實際的訂閱。

1. 定義感測器組
2. 定義目標組
3. 定義訂閱

定義感測器組

感測器組配置標識要流式傳輸的資訊。以下配置模板提供了配置感測器組所需的配置。

```
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#
```

以下示例顯示路由器CLI上的實際示例，其中顯示感測器組配置的實際示例：

```
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#
```

可以將多個感測器路徑作為同一SensorGroup定義的一部分：

```
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#
```

定義目標組

目標組配置標識將資訊流傳送到的目標。

它有三個關鍵引數

1. 作業階段方向
2. 要使用的編碼
3. 要使用的傳輸協定

以下是範例：

```
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#
```

定義預訂

訂閱將感測器組和目標組資訊繫結在一起，作為配置的最後一部分。示例間隔被定義為預訂的一部分。

以下是範例：

```
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#
```

完整配置示例

```
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#
```

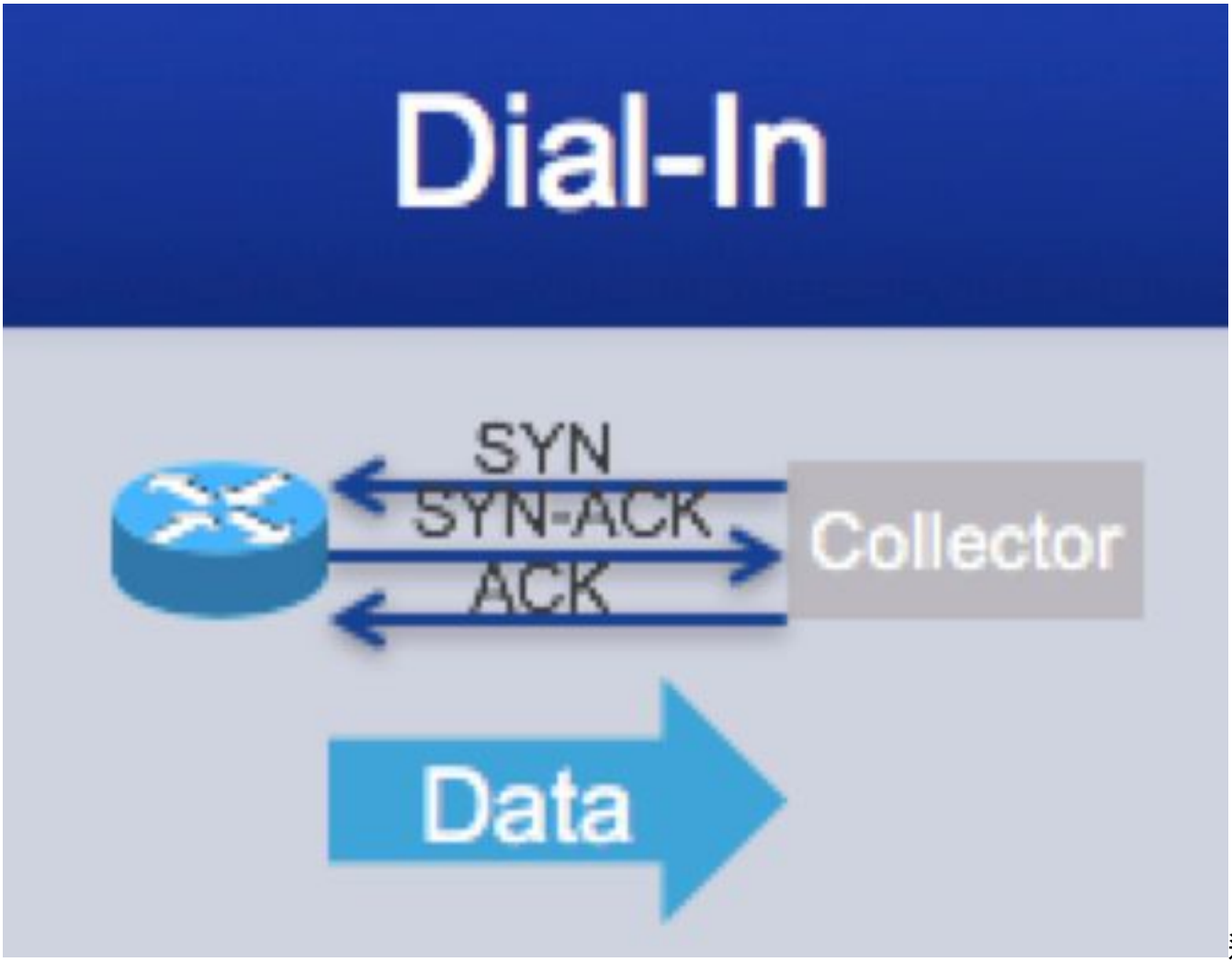
撥出的優點

- 傳輸選項具有更大的靈活性。
- 無需為入站管理流量開啟埠。
- 任播和負載平衡。

撥入組態中斷

在撥入模式下，MDT收集器/接收器/協調器撥入路由器並動態訂閱一個或多個感測器路徑或訂閱。路由器充當伺服器，客戶端充當接收器。

僅形成一個會話，路由器通過同一個會話傳輸遙測資料。當接收方取消訂閱或會話終止時，此動態訂閱終止。



遙測撥入

由於收集器「撥入」路由器，因此無需在配置中指定每個MDT目標。只需在路由器上啟用gRPC服務，連線您的客戶端，並動態啟用所需的遙測訂閱。

從配置的角度來看，遙測配置是一個類似於上述過程的三步過程。首先，我們需要啟用gRPC。其次，我們確定資訊必須流式傳輸到的目標，並在感測器組配置下捕獲該資訊。第三，使用前兩步中確定的資訊配置實際的訂閱。

1. 啟用gRPC
2. 定義感測器組
3. 定義訂閱

按一下以展開

只有gRPC支援撥入模式

只有gRPC支援撥入模式

啟用gRPC

首先，我們需要在路由器上啟用gRPC伺服器以接受來自收集器的傳入連線。

按一下以展開

- <port-number>的範圍是從57344到57999。如果埠號不可用，則顯示錯誤。

<port-number>的範圍是從57344到57999。如果埠號不可用，則顯示錯誤。

```
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#
```

定義感測器組

感測器組配置標識要流式傳輸的資訊。以下配置模板提供了配置感測器組所需的配置。

以下示例顯示了路由器CLI上的實際示例，其中顯示了感測器組配置的實際示例

```
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#
```

定義預訂

訂閱將感測器組和gRPC繫結在一起，作為配置的最後一部分。示例間隔被定義為預訂的一部分。

以下配置模板提供配置訂閱所需的配置。

以下示例顯示了來自路由器CLI的實際示例，在此示例中，我們將建立訂閱並將感測器組和目標組繫結在一起，同時定義取樣速率。

```
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#
```

完成配置模板和示例

```
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#
```

撥入的優勢

- 用於配置和串流的單一通道
- 路由器/裝置上的偵聽埠
- 臨時連線
- 當前僅可使用gRPC/gNMI

事件驅動遙測

在事件驅動遙測中，只有在發生事件時，才會從預訂的資料集中流式傳輸資料。

事件驅動遙測配置

基於事件的遙測的配置與基於節奏的遙測類似，事件驅動遙測的配置唯一區別是樣本間隔。將示例間隔值配置為零將設定對基於事件的遙測的訂閱。

完成配置模板和撥出示例

```
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#
```

以下示例顯示了路由器CLI的實際示例。

```
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#
```

撥入的完整配置模板和示例

```
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#
```

以下示例顯示了路由器CLI的實際示例。

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

使用SHOW命令驗證遙測

從路由器的角度來說，我們可以檢驗為每個感測器組、目標組和訂閱配置的引數

```
// 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 10:06:11.2499006664 +0000
    Sensor Path:          Cisco-IOS-XR-infra-statsd-oper:infra-
statistics/interfaces/interface/latest/generic-counters

RP/0/RP0/CPU0:XR#

// ALL CONFIGURED DESTINATIONS
RP/0/RP0/CPU0:XR#show telemetry model-driven destination
  Group Id        IP              Port    Encoding            Transport   State
  ------------------------------------------------------------------------------
  DestGroup101    172.16.128.3    5432    self-describing-gpb tcp         Active
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#
```

遙測收集堆疊

除了路由器配置之外，基於遙測的解決方案還需要收集器、資料庫和監控/分析軟體等多個元件。這些元件可以單獨配置，也可以作為單個綜合產品的一部分。

- 應安裝解碼器以接收入站資料包並將它們傳遞至其它儲存位置。
- 需要時序資料庫（也稱為TSDB）才能儲存流資訊。
- 還需要一個圖形工具來視覺化從內部資料庫獲取的資料。

它超出了詳細描述集合堆疊的範圍。Cisco Crossworks健康狀況洞察允許零接觸遙測，自動為裝置調配遙測配置，並在時序資料庫(TSDB)中建立表/架構。它簡化了收集和清理資料的運營和網路管理開銷，從而使運營商能夠專注於其業務目標。使用通用收集器通過SNMP、CLI和模型驅動遙測收集網路裝置資料，可以避免資料重複，還可以減少裝置和網路的負載。

網路中遙測的部署注意事項

以下是分析網路中遙測部署時需要考慮的各種因素。

擴展

遙測可以傳輸大量資料，建議仔細考慮可擴充性問題。

僅流式傳輸所需資料

每個Yang模型將具有多個葉節點。建議具體說明所需的資訊及不需要的資訊。建議深入瞭解Yang模型並確定遙測用例所需的資料路徑。

考慮流資料量

流傳輸遙測資料的總量需要考慮以下幾點：

1. 網路中的頻寬分配
2. Qos
3. 其他應用程式/軟體(如 收集器軟體時間系列資料庫分析/視覺化軟體
4. 編碼效率（在「遙測設計手冊」一節中討論）直接影響流傳輸的資料量。如果可行，建議使用緊湊型GPB。
5. 收集間隔直接影響多個方面，包括但不限於以下方面。 網路中的頻寬利用率資料庫儲存要求流式傳輸資料的裝置的效能

建議根據應用要求評估集合的頻率。

總之，建議考慮在認為可行的情況下在源或目標位置過濾不需要的資料。我們確實可以選擇過濾不想要的資料。過濾可在兩個級別上執行： -

1. 在源位置 — 裝置流式傳輸資料。
2. 在目標位置 — 收集器正在收集和規範化資料。（在收集器處過濾不在本檔案的範圍之內）

以下示例顯示通過應用萬用字元僅對感測器路徑內的Hunded Gig介面過濾資料。

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

參考資料

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

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。