ICS 33.040.50

M 42

|  |
| --- |
|  |

YD

中华人民共和国通信行业标准

XX/T XXXXX—XXXX

|  |
| --- |
|  |

接入网设备支持基于Telemetry接口采集功能的技术要求

Technical Requirements For Access Network Equipments Supporting Data Collection Based On Telemetry Interface

|  |
| --- |
| （送审稿） |
|  |

XXXX-XX-XX发布

XXXX-XX-XX实施

中华人民共和国工业和信息化部发布

目 次

[前　　言 III](#_Toc66886030)

[引　　言 V](#_Toc66886031)

[1 范围 1](#_Toc66886032)

[2 规范性引用文件 1](#_Toc66886033)

[3 术语和定义 1](#_Toc66886034)

[4 缩略语 2](#_Toc66886066)

[5 接入网设备支持基于Telemetry接口采集功能的系统架构 3](#_Toc66886067)

[6 采集功能要求 4](#_Toc66886068)

[6.1 总体要求 4](#_Toc66886069)

[6.2 NETCONF配置及gRPC上报模式 4](#_Toc66886070)

[6.3 NETCONF配置及UDP上报模式 6](#_Toc66886071)

[6.4 gRPC订阅及gRPC上报模式 7](#_Toc66886072)

[7 采集项及性能要求 8](#_Toc66886073)

[7.1 采集项 8](#_Toc66886074)

[7.2 OLT设备采集配置性能要求 11](#_Toc66886075)

[7.3 OLT设备支持的采集上报的性能要求 12](#_Toc66886076)

[附录A （规范性） NETCONF协议数据模型 13](#_Toc66886077)

[A.1 内容层数据模型 13](#_Toc66886078)

[A.1.1 an-telemetry.yang 13](#_Toc66886079)

[A.1.2 an-telemetry-ext.yang 30](#_Toc66886080)

[A.1.3 an-telemetry-types.yang 35](#_Toc66886081)

[A.1.4 an-inet-types.yang 37](#_Toc66886082)

[附录B （规范性） gRPC协议数据模型层数据模型 45](#_Toc66886083)

[B.1 gRPC层数据模型 45](#_Toc66886084)

[B.1.1 grpc-dialin.proto 45](#_Toc66886085)

[B.1.2 grpc-dialout.proto 47](#_Toc66886086)

[B.2 telemetry层数据模型 47](#_Toc66886087)

[B.2.1 telemetry.proto 47](#_Toc66886088)

[B.3 业务数据层数据模型 48](#_Toc66886089)

[B.3.1 an-bb-queue-kpi.proto 48](#_Toc66886090)

[B.3.2 an-bb-service-flow-kpi.proto 49](#_Toc66886091)

[B.3.3 an-ethernet-kpi.proto 50](#_Toc66886092)

[B.3.4 an-epon-onu-transceivers.proto 52](#_Toc66886093)

[B.3.5 an-epon-pm-olt-traffic.proto 53](#_Toc66886094)

[B.3.6 an-epon-pm-olt-transceivers.proto 56](#_Toc66886095)

[B.3.7 an-epon-pm-onu-local-info.proto 57](#_Toc66886096)

[B.3.8 an-epon-pm-onu-remote-info.proto 58](#_Toc66886097)

[B.3.9 an-epon-pm-onu-traffic.proto 58](#_Toc66886098)

[B.3.10 an-gpon-onu-transceivers.proto 59](#_Toc66886099)

[B.3.11 an-gpon-pm-olt-traffic.proto 60](#_Toc66886100)

[B.3.12 an-gpon-pm-olt-transceivers.proto 62](#_Toc66886101)

[B.3.13 an-gpon-pm-onu-line-quality.proto 64](#_Toc66886102)

[B.3.14 an-gpon-pm-onu-local-info.proto 65](#_Toc66886103)

[B.3.15 an-gpon-pm-onu-remote-info.proto 66](#_Toc66886104)

[B.3.16 an-gpon-pm-onu-traffic.proto 68](#_Toc66886105)

[附录C （规范性） sensor path格式 69](#_Toc66886106)

[参考文献 70](#_Toc66886107)

前　　言

本文件按照 GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本文件由中国通信标准化协会提出并归口。

本文件起草单位：中国电信集团有限公司、华为技术有限公司、中国信息通信科技集团有限公司、中兴通讯股份有限公司。

本文件主要起草人：梁真铭、杨子尧、汤健、丰晓东、谢煜。

引　　言

标准化是为了建立最佳秩序、促进共同效益而展开的制定并应用标准的活动。接入网设备支持基于telemetry接口采集功能的技术要求，将在集中统一运营商运维需求的基础上，有效规范和引导设备厂商根据运营商需求研制相关设备，推动接入网的技术发展，指导实际部署和应用。

接入网设备支持基于Telemetry接口采集功能的技术要求

1. 范围

本文件规定了接入网设备支持基于telemetry接口采集功能的应用场景、技术方案和采集项及性能的要求。

本文件适用于公众电信网环境下的PON系统，专用电信网也可参照使用。

1. 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

IETF RFC 768　用户数据报协议（User Datagram Protocol）

IETF RFC 793　传输控制协议（Transmission Control Protocol）

IETF RFC 5246　传输层安全（TLS）协议版本1.2（The Transport Layer Security (TLS) Protocol Version 1.2）

IETF RFC 6241　网络配置协议（Network Configuration Protocol）

IETF RFC 7540　超文本传输协议版本第二版（Hypertext Transfer Protocol Version 2）

IETF RFC 7950　YANG数据建模语言1.1（The YANG 1.1 Data Modeling Language）

1. 术语和定义

下列术语和定义适用于本标准。



telemetry

一项远程从物理设备或虚拟设备上高速采集数据的技术。



采集对象

采集参数所关联的设备中的特定的物理或逻辑实体。



采集项

同一类采集对象中，具有同一类特征的参数集合。



采集数据实例

单个采集对象的单个采集项的采集数据集合。



设备采样间隔

设备获取目标采样数据的最小间隔。该数值是设备针对目标采样数据进行采集的能力指标。



样本间隔 sample interval

设备向采集器上报报文中的采集数据实例的时间间隔，该数值由配置订阅模块配置。例如样本间隔为5秒、设备采样间隔为1秒时，单个采集数据实例由其所对应的5个设备采样数据按照对应规则生成。



绿色报文

对流量进行双速三色限速的结果进行标记，报文分为三种颜色，用DEI进行标记，不超过CIR部分为绿色，标记为0。



黄色报文

对流量进行双速三色限速的结果进行标记，报文分为三种颜色，用DEI进行标记，超过CIR，不超过PIR部分为黄色，标记为1。







YANG

一种用来为NETCONF、NETCONF RPC以及NETCONF告警进行配置、状态数据建模的数据建模语言。

1. 缩略语

下列缩略语适用于本文件。

CPU：中央处理器（Central Processing Unit）

CIR：承诺信息速率(Committed Information Rate)

CRC：循环冗余校验（Cyclic Redundancy Check）

DOW：窗口漂移（Drift of Window）

EPON：以太网无源光网络(Ethernet Passive Optical Network)

FEC：前向纠错（Forward Error Correction）

GEM：GPON封装方式（G-PON Encapsulation Mode）

GPB：谷歌协议缓冲存储区（Google Protocol Buffer）

GPON：吉比特无源光网络（Gigabit-Capable Passive Optical Networks）

gRPC：谷歌远程过程调用（Google Remote Procedure Call）

HEC：混合纠错（Hybrid Error Correction）

HTTP：超文本传输协议（HyperText Transfer Protocol）

IP：互联网协议（Internet Protocol）

LOF：帧丢失（Loss of Frame）

LOS：光信号丢失（Loss of Signal）

NETCONF：网络配置协议（Network Configuration Protocol）

OLT：光线路终端（Optical Line Terminal）

ONU：光网络单元（Optical Network Unit）

PIR：峰值信息速率(Peak Information Rate)

PON：无源光网络（Passive Optical Network）

PPS：每秒数据包数（Packets Per Second）

RPC：远程过程调用（Remote Procedure Call）

SDN：软件定义网络（Software Defined Network）

T-CONT：传输容器（Transmission Container）

TCP：传输控制协议（Transmission Control Protocol）

TLS：传输层安全（Transport Layer Security）

UDP：用户数据报协议（User Datagram Protocol）

XG-PON：10Gbit/s无源光网络（10-Gigabit-capable Passive Optical Network）

XGS-PON：对称10Gbit/s 无源光网络（10-Gigabit-capable symmetric passive optical network）

1. 接入网设备支持基于Telemetry接口采集功能的系统架构

接入网设备（本文件中目前为OLT设备）支持基于telemetry接口的采集功能中，采用主动推送模式，支持结构化数据，具备更高的执行效率及更加实时的亚秒级采集精度，且对设备自身功能和性能影响小，结合SDN应用可为网络问题的快速定位、网络质量优化调整提供重要的大数据分析基础，满足精细化、可视化、智能监控的运维需求。

在接入网设备支持基于telemetry接口采集功能的应用场景中，telemetry采集体系可以分为采集控制器和OLT。采集控制器负责采集配置、接受OLT上报的telemetry采集数据以及数据处理和分析，OLT负责依据采集配置上报telemetry采集数据，如图1所示。

采集控制器

OLT

SNI

ONU

订阅配置

UNI

ODN

telemetry数据上报

图1 OLT支持telemetry采集的系统架构

1. 采集功能要求
   1. 总体要求

接入网设备支持基于telemetry接口的采集功能按照不同的配置和上报方式分为3种模式，分别为：NETCONF配置及gRPC上报模式、NETCONF配置及UDP上报模式、gRPC订阅及gRPC上报模式。接入网设备应至少支持NETCONF配置及gRPC上报模式、NETCONF配置及UDP上报模式中的一种，可选支持gRPC订阅及gRPC上报模式。

* 1. NETCONF配置及gRPC上报模式

采集控制器包含控制模块和采集模块。其中，控制模块应支持通过NETCONF协议，并基于配置数据模型an-telemetry.yang下发数据采集配置；OLT设备应支持作为gRPC的客户端，并基于grpc-dialout.proto、telemetry.proto和本文件表6中涉及的采集数据模型进行数据封装并上报，如图2所示。相关数据模型在本文件附录A、附录B中进行定义。

采集控制器

OLT

控制模块

采集模块

采集配置协议：

NETCONF

数据模型：

an-telemetry.yang

数据上报协议：gRPC

编码：GPB

数据模型：

- grpc-dialout.proto,

- telemetry.proto

- 采集数据模型

图2 NETCONF配置及gRPC上报模式示意图

控制模块下发给OLT设备的配置报文应遵循IETF RFC6241中NETCONF协议要求，其中OLT设备应作为NETCONF服务器端，控制器应作为NETCONF客户端。协议分层及要求如表1所示，相关配置数据模型在本文件附录A中进行定义。

表1 NETCONF配置及gRPC上报模式配置报文的协议分层及要求

| 层次 | 要求 |
| --- | --- |
| 内容层 | 应包含采集配置数据，通过YANG模型an-telemetry.yang进行采集配置数据建模，并通过其包含的sensor-path指定采集项，应符合IETF RFC7950的规定 |
| 操作层 | 应支持NETCONF协议，实现NETCONF协议时，应符合IETF RFC6241的规定 |
| 消息层 | 应支持NETCONF协议，实现NETCONF协议时，应符合IETF RFC6241的规定 |
| 通信协议层 | 应支持NETCONF协议，实现NETCONF协议时，应符合IETF RFC6241的规定 |

OLT设备上报给采集模块的采集报文应遵循gRPC协议[1]要求，协议分层及要求如表2所示：

表2 NETCONF配置及gRPC上报模式采集报文的协议分层及要求

| 层次 | | 要求 |
| --- | --- | --- |
| 数据模型层  说明：  数据模型层分为三层，分别在不同的“.proto”文件中定义 | 业务数据层 | 应符合telemetry.proto文件中的sensor\_path字段定义，该字段取值来自不同的业务数据“.proto”文件。0是有意义的数据，应上报。全F值表示因设备原因无法上报的数据值 |
| telemetry层 | 应符合telemetry.proto文件中的定义，文件在本文件附录B中进行定义 |
| RPC层 | 应符合grpc-dialout.proto文件中的定义，文件在本文件附录B中进行定义 |
| gRPC层 | | 应支持双向数据流模式 |
| HTTP2层 | | 应支持HTTP2协议，实现HTTP2协议时，应符合IETF 7540的规定 |
| TLS层 | | 可选支持TLS协议，实现TLS协议时，应符合IETF RFC5246的规定 |
| TCP层 | | 应支持TCP协议，实现TCP协议时，应符合IETF RFC793的规定 |

* 1. NETCONF配置及UDP上报模式

采集控制器包含控制模块和采集模块，其中控制模块应支持通过NETCONF协议，并基于配置数据模型an-telemetry.yang下发数据采集配置；OLT设备应支持作为UDP的客户端，并基于telemetry.proto和本文件表6中涉及的采集数据模型进行数据封装并上报，如图3所示。相关数据模型在本文件附录A、附录B中进行定义。

采集控制器

OLT

控制模块

采集模块

采集配置协议：

NETCONF

数据模型：

an-telemetry.yang

数据上报协议：UDP

编码：GPB

数据模型：

- grpc-dialout.proto,

- telemetry.proto

- 采集数据模型

图3 NETCONF配置及UDP上报模式示意图

控制模块下发给OLT设备的配置报文格式要求同6.1.2。

OLT设备上报给采集模块的采集报文应遵循UDP协议要求，协议分层及要求如表3所示：

表3 NETCONF配置及UDP上报模式采集报文的协议分层及要求

| 层次 | | 要求 |
| --- | --- | --- |
| 数据模型层  说明：  数据模型层分为两层，分别在不同的“.proto”文件中定义 | 业务数据层 | 应符合telemetry.proto文件中的sensor\_path字段定义，该字段取值来自不同的业务数据“.proto”文件。0是有意义的数据，应上报。全F值表示因设备原因无法上报的数据值 |
| telemetry层 | 应符合telemetry.proto文件中的定义，文件在本文件附录B中进行定义 |
| 消息头层 | | 可选支持通过消息头层实现分片及编码格式指示。消息头层中第4比特为S位，第5比特至第8比特为ET位，当S位取值1，ET位取值0时表示GPB编码格式，参考IETF draft-ietf-netconf-udp-notif-01[2] |
| UDP层 | | 应支持UDP协议，实现UDP协议时，应符合IETF RFC768的规定 |

* 1. gRPC订阅及gRPC上报模式

采集控制器应支持作为gRPC客户端下发数据采集订阅；OLT设备应支持作为gRPC的服务器端，并基于grpc-dialin.proto、telemetry.proto和本文件表6中涉及的采集数据模型进行数据封装并上报，如图4所示。相关数据模型在本文件附录A、附录B中进行定义。

采集控制器

OLT

采集配置协议：

gRPC

数据模型：

grpc-dialin.proto

数据上报协议：gRPC

编码：GPB

数据模型：

- grpc-dialin.proto,

- telemetry.proto

- 采集数据模型

图4 gRPC订阅及gRPC上报模式

采集控制器下发给OLT的配置报文应遵循gRPC协议，配置报文协议分层及要求如表4所示：

表4 gRPC订阅及gRPC上报模式配置报文的协议分层及要求

| 层次 | | 要求 |
| --- | --- | --- |
| 数据模型层 | RPC层 | 应符合grpc-dialin.proto文件中的定义，文件在本文件附录B中进行定义 |
| gRPC层 | | 应支持双向数据流模式 |
| HTTP2层 | | 应支持HTTP2协议，实现HTTP2协议时，应符合IETF 7540的规定 |
| TLS层 | | 可选支持TLS协议，实现TLS协议时，应符合IETF RFC5246的规定 |
| TCP层 | | 应支持TCP协议，实现TCP协议时，应符合IETF RFC793的规定 |

采集报文应遵循gRPC协议。采集报文协议分层及要求如表5所示：

表5 gRPC订阅及gRPC上报模式采集报文的协议分层及要求

| 层次 | | 要求 |
| --- | --- | --- |
| 数据模型层  说明：  数据模型层分为三层，分别在不同的“.proto”文件中定义 | 业务数据层 | 应符合telemetry.proto文件中的sensor\_path字段定义，该字段取值来自不同的业务数据“.proto”文件。0是有意义的数据，应上报。全F值表示因设备原因无法上报的数据值 |
| telemetry层 | 应符合telemetry.proto文件中的定义，文件在本文件附录B中进行定义 |
| RPC层 | 应符合grpc-dialin.proto文件中的定义，文件在本文件附录B中进行定义 |
| gRPC层 | | 应支持双向数据流模式 |
| HTTP2层 | | 应支持HTTP2协议，实现HTTP2协议时，应符合IETF 7540的规定 |
| TLS层 | | 可选支持TLS协议，实现TLS协议时，应符合IETF RFC5246的规定 |
| TCP层 | | 应支持TCP协议，实现TCP协议时，应符合IETF RFC793的规定 |

1. 采集项及性能要求
   1. 采集项

OLT设备支持的采集项、及单个采集项对应的采集性能要求如表6所示。采集数据模型在本文件附录B中进行定义。采集配置senor path的格式要求在本文件附录C中定义。

表6 采集项及采集周期要求

| 采集类型 | 采集项 | | 采集内容 | 设备采样能力 | | 样本间隔 | | 数据对应采集模型 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 单采集对象采集 | 全量采集 | 单采集对象采集 | 全量采集 |
| 流量 | 上联口流量 | | 1. 字节数（收发） 2. 数据包数（收发、丢弃、CRC错误、超长、超短、分片、Jabber、错误） 3. 上下行速率（平均、峰值） | ≤1秒 | ≤1秒 | ≤1秒 | ≤1秒 | an-ethernet-kpi.proto |
| PON口流量 | GPON | 1. 字节数（收发） 2. 数据包数（收发、丢弃、CRC错误、超长、超短、错误） 3. 上下行速率（平均、峰值） | ≤1秒 | ≤1秒 | ≤1秒 | ≤1秒 | an-gpon-pm-olt-traffic.proto |
| EPON | an-epon-pm-olt-traffic.proto |
| 队列流量（PON端口、上联端口） | | 1. 转发字节数 2. 数据包数（转发、丢弃、丢弃未识别颜色） 3. 绿色流量（转发字节数、转发丢弃报文数） 4. 黄色流量（转发字节数、转发丢弃报文数） | ≤1秒 | ≤1秒 | ≤1秒 | ≤1秒 | an-bb-queue-kpi.proto |
| ONU流量 | GPON | 1. 下行平均速率 2. 上行平均速率 3. 下行峰值速率 4. 上行峰值速率 | 应≤3秒，宜≤1秒 | 应≤3秒，宜≤1秒 | 应≤3秒，宜≤1秒 | 应≤3秒，宜≤1秒 | an-gpon-pm-onu-traffic.proto |
| EPON | an-epon-pm-onu-traffic.proto |
| 业务流流量（VLAN粒度） | | 1. 下行总包数 2. 下行丢包数（总数、最大、最小） 3. 下行丢包率（最大、最小） 4. 下行字节数 5. 下行平均速率 6. 上行字节数 7. 上行包数（通过、丢弃） | 应≤3秒，宜≤1秒 | 应≤3秒，宜≤1秒 | 应≤3秒，宜≤1秒 | 应≤3秒，宜≤1秒 | an-bb-service-flow-kpi.proto |
| 光链路 | OLT光模块 | GPON | 1. 温度 2. 供电电压 3. 通道1类型 4. 通道1偏置电流 5. 通道1发送功率 6. 通道1空闲时刻的光功率 7. 通道2类型 8. 通道2偏置电流 9. 通道2发送功率 10. 通道2空闲时刻的光功率 11. 光模块类型 12. 光模块子类型 | ≤1分钟 | ≤1分钟 | ≤1分钟 | ≤1分钟 | an-gpon-pm-olt-transceivers.proto |
| EPON | an-epon-pm-olt-transceivers.proto |
| ONU信息采集 | ONU光模块信息（远端a） | GPON | 1. 接收光功率 2. 发送光功率 3. 偏置电流 4. 温度 5. 电压 6. 光模块类型 7. 光模块子类型 | ≤5分钟 | ≤15分钟 | ≤5分钟 | ≤15分钟 | an-gpon-pm-onu-transceivers.proto |
| EPON | an-epon-pm-onu-transceivers.proto |
| ONU状态及下行链路信息（远端a） | GPON | 1. 下行FEC校正字节数 2. 下行FEC校正code words数 3. 下行FEC不可校正 words数 4. 下行接收words总数 5. 下行FEC校正时间 6. GEM HEC错误数 7. 丢弃GEM帧数 8. 内存利用率（可选） 9. CPU利用率（可选） 10. CPU的温度（可选） 11. 发送报文数 12. 接收报文数 13. 接收报文错误数 14. TCONT队列丢包数 15. TCONT队列转发包数 | ≤5分钟 | ≤15分钟 | ≤5分钟 | ≤15分钟 | an-gpon-pm-onu-remote-info.proto |
| EPON | 1. 内存利用率 2. CPU利用率 3. CPU的温度 4. 发送报文数 5. 接收报文数 6. 接收报文错误数 | ≤5分钟 | ≤15分钟 | ≤5分钟 | ≤15分钟 | an-epon-pm-onu-remote-info.proto |
| ONU在线信息（近端 b） | GPON | 1. OLT接收光功率 2. 在线时长 3. 最后一次下线时间 4. 最后一次下线原因（可选） 5. ONU状态 | ≤5分钟 | ≤5分钟 | ≤5分钟 | ≤5分钟 | an-gpon-pm-onu-local-info.proto |
| EPON | an-epon-pm-onu-local-info.proto |
| ONU上行链路信息（近端b） | GPON | 1. LOFi告警次数 2. DOWi告警次数 3. 上行帧定界错误次数 4. 上行帧交叉奇偶校验错误次数 5. 下行帧交叉奇偶校验错误次数 6. 上行FEC解码纠正块数 7. 上行FEC解码未纠正块数 8. 上行接收words总数 9. 上行FEC校正字节数 10. 上行HEC校验错误次数 11. 上行GEM帧数 12. LOSi告警次数 13. DGi告警次数 | ≤5分钟 | ≤5分钟 | ≤5分钟 | ≤5分钟 | an-gpon-pm-onu-line-quality.proto |
| 注：EPON包含EPON、10G-EPON；GPON包含GPON、XG-PON、XGS-PON、GPON and XG-PON、GPON and XGS-PON。 | | | | | | | | |
| 1. 该采集项数据通过ONU本地采样获取，并通过OLT和ONU之间的管理通道提供给OLT。 2. 该采集项数据通过OLT本地采样获取。 | | | | | | | | |

* 1. OLT设备采集配置性能要求

OLT设备支持的采集配置数据模型an-telemetry.yang配置内容的性能要求见下表：

表7 OLT设备采集配置性能要求

| OLT设备采集配置内容 | 性能指标 |
| --- | --- |
| 最大订阅数（subscriptions） | 10 |
| 每个订阅关联sensor-group最大数量 | 5 |
| sensor-group最大数量 | 20 |
| 每个sensor-group配置采集项（sensor-path）最大数量 | 10 |
| 每个订阅关联destination-group最大数量 | 5 |
| destination-group最大数量 | 8 |
| 每个destination-group配置IP地址最大数量 | 5 |

* 1. OLT设备支持的采集上报的性能要求

1. 数据采集能力要求

OLT设备的采集能力应至少支持以下两种之一：

---单板：2K个采集数据实例/秒

---全局：30K个采集数据实例/秒

1. 数据上报能力要求

OLT设备应能把采集数据实例同时上报给至少两个目的地址，上报能力至少支持以下两种之一：

---单板：500PPS

---全局：7500PPS

附录A  
（规范性）  
NETCONF协议数据模型

1. 内容层数据模型
2. an-telemetry.yang

module an-telemetry {

yang-version "1.1";

namespace "urn:an:yang:an-telemetry";

prefix "an-telemetry";

import an-inet-types {

prefix an-inet;

}

import an-telemetry-types {

prefix an-telemetry-types;

}

description

"Data model which creates the configuration for the telemetry

systems and functions on the device.";

revision "2020-09-22";

grouping telemetry-top {

description

"Top level grouping for telemetry configuration and operational

state data";

container telemetry-system {

description

"Top level configuration and state for the

device's telemetry system.";

container sensor-groups {

description

"Top level container for sensor-groups.";

list sensor-group {

key "sensor-group-id";

description

"List of telemetry sensory groups on the local

system, where a sensor grouping represents a resuable

grouping of multiple paths and exclude filters.";

leaf sensor-group-id {

type leafref {

path "../config/sensor-group-id";

}

description

"Reference to the name or identifier of the

sensor grouping";

}

container config {

description

"Configuration parameters relating to the

telemetry sensor grouping";

uses telemetry-sensor-group-config;

}

container state {

config false;

description

"State information relating to the telemetry

sensor group";

uses telemetry-sensor-group-config;

}

container sensor-paths {

description

"Top level container to hold a set of sensor

paths grouped together";

list sensor-path {

key "path";

description

"List of paths in the model which together

comprise a sensor grouping. Filters for each path

to exclude items are also provided.";

leaf path {

type leafref {

path "../config/path";

}

description

"Reference to the path of interest";

}

container config {

description

"Configuration parameters to configure a set

of data model paths as a sensor grouping";

uses telemetry-sensor-path-config;

}

container state {

config false;

description

"Configuration parameters to configure a set

of data model paths as a sensor grouping";

uses telemetry-sensor-path-config;

}

}

}

}

}

container destination-groups {

description

"Top level container for destination group configuration

and state.";

list destination-group {

key "group-id";

description

"List of destination-groups. Destination groups allow the

reuse of common telemetry destinations across the

telemetry configuration. An operator references a

set of destinations via the configurable

destination-group-identifier.

A destination group may contain one or more telemetry

destinations";

leaf group-id {

type leafref {

path "../config/group-id";

}

description

"Unique identifier for the destination group";

}

container config {

description

"Top level config container for destination groups";

leaf group-id {

type string;

description

"Unique identifier for the destination group";

}

}

container state {

config false;

description

"Top level state container for destination groups";

leaf group-id {

type string;

description

"Unique identifier for destination group";

}

}

container destinations {

description

"The destination container lists the destination

information such as IP address and port of the

telemetry messages from the network element.";

list destination {

key "destination-address destination-port";

description

"List of telemetry stream destinations";

leaf destination-address {

type leafref {

path "../config/destination-address";

}

description

"Reference to the destination address of the

telemetry stream";

}

leaf destination-port {

type leafref {

path "../config/destination-port";

}

description

"Reference to the port number of the stream

destination";

}

container config {

description

"Configuration parameters relating to

telemetry destinations";

uses telemetry-stream-destination-config;

}

container state {

config false;

description

"State information associated with

telemetry destinations";

uses telemetry-stream-destination-config;

}

}

}

}

}

container subscriptions {

description

"This container holds information for both persistent

and dynamic telemetry subscriptions.";

container persistent-subscriptions {

description

"This container holds information relating to persistent

telemetry subscriptions. A persistent telemetry

subscription is configued locally on the device through

configuration, and is persistent across device restarts or

other redundancy changes.";

list persistent-subscription {

key "name";

description

"List of telemetry subscriptions. A telemetry

subscription consists of a set of collection

destinations, stream attributes, and associated paths to

state information in the model (sensor data)";

leaf name {

type leafref {

path "../config/name";

}

description

"Reference to the identifier of the subscription

itself. The id will be the handle to refer to the

subscription once created";

}

container config {

description

"Config parameters relating to the telemetry

subscriptions on the local device";

uses telemetry-subscription-name-config;

uses telemetry-local-source-address-config;

uses telemetry-qos-marking-config;

uses telemetry-stream-protocol-config;

uses telemetry-stream-encoding-config;

}

container state {

config false;

description

"State parameters relating to the telemetry

subscriptions on the local device";

uses telemetry-subscription-name-config;

uses telemetry-subscription-config;

uses telemetry-subscription-state;

uses telemetry-local-source-address-config;

uses telemetry-qos-marking-config;

uses telemetry-stream-protocol-config;

uses telemetry-stream-encoding-config;

}

container sensor-profiles {

description

"A sensor profile is a set of sensor groups or

individual sensor paths which are associated with a

telemetry subscription. This is the source of the

telemetry data for the subscription to send to the

defined collectors.";

list sensor-profile {

key "sensor-group";

description

"List of telemetry sensor groups used

in the subscription";

leaf sensor-group {

type leafref {

path "../config/sensor-group";

}

description

"Reference to the telemetry sensor group name";

}

container config {

description

"Configuration parameters related to the sensor

profile for a subscription";

uses telemetry-sensor-profile-config;

}

container state {

config false;

description

"State information relating to the sensor profile

for a subscription";

uses telemetry-sensor-profile-config;

}

}

}

container destination-groups {

description

"A subscription may specify destination addresses.

If the subscription supplies destination addresses,

the network element will be the initiator of the

telemetry streaming, sending it to the destination(s)

specified.

If the destination set is omitted, the subscription

preconfigures certain elements such as paths and

sample intervals under a specified subscription ID.

In this case, the network element will NOT initiate an

outbound connection for telemetry, but will wait for

an inbound connection from a network management

system.

It is expected that the network management system

connecting to the network element will reference

the preconfigured subscription ID when initiating

a subscription.";

list destination-group {

key "group-id";

description

"Identifier of the previously defined destination

group";

leaf group-id {

type leafref {

path "../config/group-id";

}

description

"The destination group id references a configured

group of destinations for the telemetry stream.";

}

container config {

description

"Configuration parameters related to telemetry

destinations.";

leaf group-id {

type leafref {

path "../../../../../../../destination-groups"

+ "/destination-group/group-id";

}

description

"The destination group id references a reusable

group of destination addresses and ports for

the telemetry stream.";

}

}

container state {

config false;

description

"State information related to telemetry

destinations";

leaf group-id {

type leafref {

path "../../../../../../../destination-groups"

+ "/destination-group/group-id";

}

description

"The destination group id references a reusable

group of destination addresses and ports for

the telemetry stream.";

}

}

}

}

}

}

container dynamic-subscriptions {

description

"This container holds information relating to dynamic

telemetry subscriptions. A dynamic subscription is

typically configured through an RPC channel, and does not

persist across device restarts, or if the RPC channel is

reset or otherwise torn down.";

list dynamic-subscription {

key "id";

config false;

description

"List representation of telemetry subscriptions that

are configured via an inline RPC, otherwise known

as dynamic telemetry subscriptions.";

leaf id {

type leafref {

path "../state/id";

}

description

"Reference to the identifier of the subscription

itself. The id will be the handle to refer to the

subscription once created";

}

container state {

config false;

description

"State information relating to dynamic telemetry

subscriptions.";

uses telemetry-subscription-config;

uses telemetry-stream-destination-config;

uses telemetry-stream-frequency-config;

uses telemetry-heartbeat-config;

uses telemetry-suppress-redundant-config;

uses telemetry-qos-marking-config;

uses telemetry-stream-protocol-config;

uses telemetry-stream-encoding-config;

}

container sensor-paths {

description

"Top level container to hold a set of sensor

paths grouped together";

list sensor-path {

key "path";

description

"List of paths in the model which together

comprise a sensor grouping. Filters for each path

to exclude items are also provided.";

leaf path {

type leafref {

path "../state/path";

}

description

"Reference to the path of interest";

}

container state {

config false;

description

"State information for a dynamic subscription's

paths of interest";

uses telemetry-sensor-path-config;

}

}

}

}

}

}

}

}

// identity statements

// typedef statements

// grouping statements

grouping telemetry-sensor-path-config {

description

"Configuration parameters relating to the

grouping of data model paths comprising a

sensor grouping";

leaf path {

type string;

description

"Path to a section of operational state of interest

(the sensor).";

}

leaf exclude-filter {

type string;

description

"Filter to exclude certain values out of the state

values";

//May not be necessary. Could remove.

}

}

grouping telemetry-heartbeat-config {

description

"Configuration parameters relating to the

heartbeat of the telemetry subscription";

leaf heartbeat-interval {

type uint64;

description

"Maximum time interval in seconds that may pass

between updates from a device to a telemetry collector.

If this interval expires, but there is no updated data to

send (such as if suppress\_updates has been configured), the

device must send a telemetry message to the collector.";

}

}

grouping telemetry-suppress-redundant-config {

description

"Configuration parameters relating to suppression of

redundant upstream updates";

leaf suppress-redundant {

type boolean;

description

"Boolean flag to control suppression of redundant

telemetry updates to the collector platform. If this flag is

set to TRUE, then the collector will only send an update at

the configured interval if a subscribed data value has

changed. Otherwise, the device will not send an update to

the collector until expiration of the heartbeat interval.";

}

}

grouping telemetry-sensor-profile-config {

description

"Configuration parameters relating to the sensor groups

used in the sensor profile";

leaf sensor-group {

type leafref {

path "../../../../../../../sensor-groups/sensor-group"

+ "/config/sensor-group-id";

}

description

"Reference to the sensor group which is used in the profile";

}

uses telemetry-stream-subscription-config;

}

grouping telemetry-stream-subscription-config {

description

"Configuration used when the sensor is a stream based sensor.";

uses telemetry-stream-frequency-config;

uses telemetry-heartbeat-config;

uses telemetry-suppress-redundant-config;

}

grouping telemetry-qos-marking-config {

description

"Config parameters relating to the quality of service

marking on device generated telemetry packets";

leaf originated-qos-marking {

type an-inet:dscp;

description

"DSCP marking of packets generated by the telemetry

subsystem on the network device.";

}

}

grouping telemetry-sensor-group-config {

description

"Config parameters related to the sensor groups

on the device";

leaf sensor-group-id {

type string;

description

"Name or identifier for the sensor group itself.

Will be referenced by other configuration specifying a

sensor group";

}

}

grouping telemetry-subscription-config {

description

"Configuration parameters relating to the telemetry

subscription";

leaf id {

type uint64;

description

"System generated identifer of the telemetry

subscription.";

}

}

grouping telemetry-subscription-name-config {

description

"Configuration parameters relating to the configured

name of the telemetry subscription. The name is a user

configured string value which uniquely identifies the

subscription in the configuration database.";

leaf name {

type string;

description

"User configured identifier of the telemetry

subscription. This value is used primarily for

subscriptions configured locally on the network

element.";

}

}

grouping telemetry-subscription-state {

description

"State values for the telemetry subscription";

//TODO add values

}

grouping telemetry-stream-protocol-config {

description

"Configuration parameters relating to the

transport protocol carrying telemetry

data.";

leaf protocol {

type identityref {

base an-telemetry-types:STREAM\_PROTOCOL;

}

description

"Selection of the transport protocol for the telemetry

stream.";

}

}

grouping telemetry-stream-encoding-config {

description

"Configuration parameters relating to the

encoding of telemetry data to and from the

network element. The encoding method controls

specifically the wire format of the telemetry

data, and also controls which RPC framework

may be in use to exchange telemetry data.";

leaf encoding {

type identityref {

base an-telemetry-types:DATA\_ENCODING\_METHOD;

}

description

"Selection of the specific encoding or RPC framework

for telemetry messages to and from the network element.";

}

}

grouping telemetry-stream-destination-config {

description

"Configuration parameters for the stream destinations";

leaf destination-address {

type an-inet:ip-address;

description

"IP address of the telemetry stream destination";

}

leaf destination-port {

type uint16;

description

"Protocol (udp or tcp) port number for the telemetry

stream destination";

}

}

grouping telemetry-stream-frequency-config {

description

"Config parameters for the frequency of updates to

the collector";

leaf sample-interval {

type uint64;

description

" The accuracy of the collected data reported by the device

to the collector";

}

}

grouping telemetry-sensor-specification {

description

"Config related to creating telemetry sensor groups. A sensor

group is a related set of sensor paths and/or filters to

exclude items. A group is assigned a reusable identifer, so

it can be used in multiple telemetry subscriptions.";

list telemetry-sensor-group {

key "telemetry-sensor-group-id";

description

"List of telemetry sensor groups";

leaf telemetry-sensor-group-id {

type string;

description

"The sensor group identifer is a reusable handle which

identifies a single sensor group. It is referenced from

the subscription configuration.";

}

uses telemetry-sensor-paths;

}

}

grouping telemetry-sensor-paths {

description

"This grouping contains these paths to leaves or containers

in the data model which are the sources of telemetry

information.";

list telemetry-sensor-paths {

key "telemetry-sensor-path";

description

"A list of sensor paths and exclude filters which comprise

a sensor grouping";

leaf telemetry-sensor-path {

type string;

description

"The sensor path is a path to a portion of operational

state of interest in the data model";

}

}

}

grouping telemetry-local-source-address-config {

description

"Config relating to the local source address for telemetry

messages";

// TODO: Make this a reference to an interface.

leaf local-source-address {

type an-inet:ip-address;

description

"The IP address which will be the source of packets from

the device to a telemetry collector destination.";

}

}

// data definition statements

uses telemetry-top;

// augment statements

// rpc statements

// notification statements

}

1. an-telemetry-ext.yang

module an-telemetry-ext {

// namespace

namespace "urn:an:yang:an-telemetry-ext";

prefix "an-telemetry-ext";

import an-telemetry {

prefix an-telemetry;

}

description "Data model which creates the condition for the telemetry.";

revision 2020-11-05;

grouping telemetry-sensor-path-condition-op-config {

description

"Parameters for a filter criterion.";

leaf op-field{

type string {

length "1..63";

}

description

"Reference field in the filter criterion. Different paths support different reference fields.";

}

leaf op-type{

type enumeration {

enum eq {

value 0;

description

"Equal.";

}

enum gt {

value 1;

description

"Greater than.";

}

enum ge {

value 2;

description

"Greater than or equal.";

}

enum lt {

value 3;

description

"Less than.";

}

enum le {

value 4;

description

"Less than or equal.";

}

}

description

"Reference to the op-type of the filter criterion.";

}

leaf op-value{

type string {

length "1..63";

}

description

"Comparison value of the filter criterion.For the actual value range, see the definition of op-field in the YANG model.";

}

}

grouping telemetry-sensor-path-filter-condition-config {

description

"Parameters for filter criteria.";

container conditions {

description

"Top level container for filter criteria, including the filter information such as op-field, op-type, and op-value.";

list condition {

key "op-field op-type op-value";

max-elements "2";

description

"List of filter criteria.";

leaf "op-field" {

type leafref {

path "../config/op-field";

}

description

"Reference to the op-field of the filter criterion.";

}

leaf "op-type" {

type leafref {

path "../config/op-type";

}

description

"Reference to the op-type of the filter criterion.";

}

leaf "op-value" {

type leafref {

path "../config/op-value";

}

description

"Comparison value of the filter criterion. Op-value must be an integer and the maximum value range is [0, 4294967295]. For the actual value range, see the definition of op-field in the YANG model.";

}

container config {

description

"Configuration parameters for a filter criterion.";

uses telemetry-sensor-path-condition-op-config;

}

container state {

config false;

description

"State information for a filter criterion.";

uses telemetry-sensor-path-condition-op-config;

}

}

}

}

grouping telemetry-sensor-path-filter-attribute-config {

description

"Configuration parameters relating to the configured name of the telemetry sensor path filter.";

leaf name {

type string {

length "1..8";

pattern "[A-Za-z0-9]+";

}

description "Filter name for the telemetry sensor path filter.";

}

leaf condition-relation{

type enumeration {

enum and {

value 1;

description

"The relationship between filter criteria in a filter of a sampling path is and.";

}

enum or {

value 2;

description

"The relationship between filter criteria in a filter of a sampling path is or.";

}

}

default "and";

description

"Relationship between filter criteria in a filter of a sampling path.";

}

}

grouping telemetry-sensor-path-filter-config {

description

"Condition-based filtering's parameter of the sampling path. If the conditions are met, the data is reported.";

container filters {

description

"Top level container for filters. If the filter criteria are met, sampled data is sent to a collector. A filter and heartbeat interval or redundancy suppression cannot be configured at the same time.";

list filter {

key "name";

max-elements "1";

description

"Filter list of a sampling path.";

leaf name {

type leafref {

path "../config/name";

}

description

"Filter name of a sampling path.";

}

container config {

description

"Configuration parameters for a filter in a sampling path.";

uses telemetry-sensor-path-filter-attribute-config;

}

container state {

config false;

description

"State information for a filter in a sampling path.";

uses telemetry-sensor-path-filter-attribute-config;

}

uses telemetry-sensor-path-filter-condition-config;

}

}

}

augment "/an-telemetry:telemetry-system/an-telemetry:sensor-groups/an-telemetry:sensor-group/an-telemetry:sensor-paths/an-telemetry:sensor-path"{

description

"List of paths in the model which together comprise a sensor grouping. Filters for each path to exclude items are also provided.";

uses telemetry-sensor-path-filter-config;

}

}

1. an-telemetry-types.yang

module an-telemetry-types {

yang-version "1.1";

namespace "urn:an:yang:an-telemetry-types";

prefix "an-telemetry-types";

description

"This module defines type and identities used by the access network

telemetry model.";

revision "2020-09-22";

identity DATA\_ENCODING\_METHOD {

description

"Base identity for supported encoding for configuration and

operational state data";

}

identity ENC\_XML {

base DATA\_ENCODING\_METHOD;

description

"XML encoding";

}

identity ENC\_JSON\_IETF {

base DATA\_ENCODING\_METHOD;

description

"JSON encoded based on IETF draft standard";

reference

"draft-ietf-netmod-yang-json";

}

identity ENC\_PROTO3 {

base DATA\_ENCODING\_METHOD;

description

"Protocol buffers v3";

reference

"https://developers.google.com/protocol-buffers/docs/overview";

}

identity STREAM\_PROTOCOL {

description "Base identity for a telemetry stream protocol";

}

identity STREAM\_SSH {

base "STREAM\_PROTOCOL";

description

"Telemetry stream is carried over a SSH connection";

}

identity STREAM\_GRPC {

base "STREAM\_PROTOCOL";

description

"Telemetry stream is carried over via the gRPC framework";

}

identity STREAM\_UDP {

base "STREAM\_PROTOCOL";

description

"Telemetry stream is carried over via the UDP framework";

}

identity STREAM\_JSON\_RPC {

base "STREAM\_PROTOCOL";

description

"Telemetry stream is carried via the JSON-RPC framework";

}

identity STREAM\_THRIFT\_RPC {

base "STREAM\_PROTOCOL";

description

"Telemetry stream is carried via the Apache Thrift framework";

}

identity STREAM\_WEBSOCKET\_RPC {

base "STREAM\_PROTOCOL";

description

"Telemetry stream is carried by the WebSocket framework";

}

// typedef statements

}

1. an-inet-types.yang

module an-inet-types {

yang-version "1.1";

namespace "urn:an:yang:an-inet-types";

prefix "an-inet";

description

"This module contains a set of Internet address related

types for use in access network modules.

Portions of this code were derived from IETF RFC 6021.";

revision "2020-09-22";

// IPv4 and IPv6 types.

typedef ipv4-address {

type string {

pattern '(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|' +

'25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4]' +

'[0-9]|25[0-5])';

}

description

"An IPv4 address in dotted quad notation using the default

zone.";

}

typedef ipv4-address-zoned {

type string {

pattern '(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|' +

'25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4]' +

'[0-9]|25[0-5])(%[a-zA-Z0-9\_]+)';

}

description

"An IPv4 address in dotted quad notation. This type allows

specification of a zone index to disambiguate identical

address values. For link-local addresses, the index is

typically the interface index or interface name.";

}

typedef ipv6-address {

type string {

pattern

// Must support compression through different lengths

// therefore this regexp is complex.

'(([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|' +

'([0-9a-fA-F]{1,4}:){1,7}:|' +

'([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}|' +

'([0-9a-fA-F]{1,4}:){1,5}(:[0-9a-fA-F]{1,4}){1,2}|' +

'([0-9a-fA-F]{1,4}:){1,4}(:[0-9a-fA-F]{1,4}){1,3}|' +

'([0-9a-fA-F]{1,4}:){1,3}(:[0-9a-fA-F]{1,4}){1,4}|' +

'([0-9a-fA-F]{1,4}:){1,2}(:[0-9a-fA-F]{1,4}){1,5}|' +

'[0-9a-fA-F]{1,4}:((:[0-9a-fA-F]{1,4}){1,6})|' +

':((:[0-9a-fA-F]{1,4}){1,7}|:)' +

')';

}

description

"An IPv6 address represented as either a full address; shortened

or mixed-shortened formats, using the default zone.";

}

typedef ipv6-address-zoned {

type string {

pattern

// Must support compression through different lengths

// therefore this regexp is complex.

'(([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|' +

'([0-9a-fA-F]{1,4}:){1,7}:|' +

'([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}|' +

'([0-9a-fA-F]{1,4}:){1,5}(:[0-9a-fA-F]{1,4}){1,2}|' +

'([0-9a-fA-F]{1,4}:){1,4}(:[0-9a-fA-F]{1,4}){1,3}|' +

'([0-9a-fA-F]{1,4}:){1,3}(:[0-9a-fA-F]{1,4}){1,4}|' +

'([0-9a-fA-F]{1,4}:){1,2}(:[0-9a-fA-F]{1,4}){1,5}|' +

'[0-9a-fA-F]{1,4}:((:[0-9a-fA-F]{1,4}){1,6})|' +

':((:[0-9a-fA-F]{1,4}){1,7}|:)' +

')(%[a-zA-Z0-9\_]+)';

}

description

"An IPv6 address represented as either a full address; shortened

or mixed-shortened formats. This type allows specification of

a zone index to disambiguate identical address values. For

link-local addresses, the index is typically the interface

index or interface name.";

}

typedef ipv4-prefix {

type string {

pattern '(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|' +

'25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4]' +

'[0-9]|25[0-5])/(([0-9])|([1-2][0-9])|(3[0-2]))';

}

description

"An IPv4 prefix represented in dotted quad notation followed by

a slash and a CIDR mask (0 <= mask <= 32).";

}

typedef ipv6-prefix {

type string {

pattern

'(([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|' +

'([0-9a-fA-F]{1,4}:){1,7}:|' +

'([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}|' +

'([0-9a-fA-F]{1,4}:){1,5}(:[0-9a-fA-F]{1,4}){1,2}|' +

'([0-9a-fA-F]{1,4}:){1,4}(:[0-9a-fA-F]{1,4}){1,3}|' +

'([0-9a-fA-F]{1,4}:){1,3}(:[0-9a-fA-F]{1,4}){1,4}|' +

'([0-9a-fA-F]{1,4}:){1,2}(:[0-9a-fA-F]{1,4}){1,5}|' +

'[0-9a-fA-F]{1,4}:((:[0-9a-fA-F]{1,4}){1,6})|' +

':((:[0-9a-fA-F]{1,4}){1,7}|:)' +

')/(12[0-8]|1[0-1][0-9]|[1-9][0-9]|[0-9])';

}

description

"An IPv6 prefix represented in full, shortened, or mixed

shortened format followed by a slash and CIDR mask

(0 <= mask <= 128).";

}

typedef ip-address {

type union {

type ipv4-address;

type ipv6-address;

}

description

"An IPv4 or IPv6 address with no prefix specified.";

}

typedef ip-prefix {

type union {

type ipv4-prefix;

type ipv6-prefix;

}

description

"An IPv4 or IPv6 prefix.";

}

typedef ip-version {

type enumeration {

enum UNKNOWN {

value 0;

description

"An unknown or unspecified version of the Internet

protocol.";

}

enum IPV4 {

value 4;

description

"The IPv4 protocol as defined in RFC 791.";

}

enum IPV6 {

value 6;

description

"The IPv6 protocol as defined in RFC 2460.";

}

}

description

"This value represents the version of the IP protocol.

Note that integer representation of the enumerated values

are not specified, and are not required to follow the

InetVersion textual convention in SMIv2.";

reference

"RFC 791: Internet Protocol

RFC 2460: Internet Protocol, Version 6 (IPv6) Specification

RFC 4001: Textual Conventions for Internet Network Addresses";

}

typedef domain-name {

type string {

length "1..253";

pattern

'((([a-zA-Z0-9\_]([a-zA-Z0-9\-\_]){0,61})?[a-zA-Z0-9]\.)\*' +

'([a-zA-Z0-9\_]([a-zA-Z0-9\-\_]){0,61})?[a-zA-Z0-9]\.?)' +

'|\.';

}

description

"The domain-name type represents a DNS domain name.

Fully quallified left to the models which utilize this type.

Internet domain names are only loosely specified. Section

3.5 of RFC 1034 recommends a syntax (modified in Section

2.1 of RFC 1123). The pattern above is intended to allow

for current practice in domain name use, and some possible

future expansion. It is designed to hold various types of

domain names, including names used for A or AAAA records

(host names) and other records, such as SRV records. Note

that Internet host names have a stricter syntax (described

in RFC 952) than the DNS recommendations in RFCs 1034 and

1123, and that systems that want to store host names in

schema nodes using the domain-name type are recommended to

adhere to this stricter standard to ensure interoperability.

The encoding of DNS names in the DNS protocol is limited

to 255 characters. Since the encoding consists of labels

prefixed by a length bytes and there is a trailing NULL

byte, only 253 characters can appear in the textual dotted

notation.

Domain-name values use the US-ASCII encoding. Their canonical

format uses lowercase US-ASCII characters. Internationalized

domain names MUST be encoded in punycode as described in RFC

3492";

}

typedef host {

type union {

type ip-address;

type domain-name;

}

description

"The host type represents either an unzoned IP address or a DNS

domain name.";

}

typedef as-number {

type uint32;

description

"A numeric identifier for an autonomous system (AS). An AS is a

single domain, under common administrative control, which forms

a unit of routing policy. Autonomous systems can be assigned a

2-byte identifier, or a 4-byte identifier which may have public

or private scope. Private ASNs are assigned from dedicated

ranges. Public ASNs are assigned from ranges allocated by IANA

to the regional internet registries (RIRs).";

reference

"RFC 1930 Guidelines for creation, selection, and registration

of an Autonomous System (AS)

RFC 4271 A Border Gateway Protocol 4 (BGP-4)";

}

typedef dscp {

type uint8 {

range "0..63";

}

description

"A differentiated services code point (DSCP) marking within the

IP header.";

reference

"RFC 2474 Definition of the Differentiated Services Field

(DS Field) in the IPv4 and IPv6 Headers";

}

typedef ipv6-flow-label {

type uint32 {

range "0..1048575";

}

description

"The IPv6 flow-label is a 20-bit value within the IPv6 header

which is optionally used by the source of the IPv6 packet to

label sets of packets for which special handling may be

required.";

reference

"RFC 2460 Internet Protocol, Version 6 (IPv6) Specification";

}

typedef port-number {

type uint16;

description

"A 16-bit port number used by a transport protocol such as TCP

or UDP.";

reference

"RFC 768 User Datagram Protocol

RFC 793 Transmission Control Protocol";

}

typedef uri {

type string;

description

"An ASCII-encoded Uniform Resource Identifier (URI) as defined

in RFC 3986.";

reference

"RFC 3986 Uniform Resource Identifier (URI): Generic Syntax";

}

typedef url {

type string;

description

"An ASCII-encoded Uniform Resource Locator (URL) as defined

in RFC 3986, section 1.1.3";

reference

"RFC 3986, paragraph 1.1.3";

}

}

附录B  
（规范性）  
gRPC协议数据模型层数据模型

1. gRPC层数据模型
2. grpc-dialin.proto

syntax = "proto3";

package dialin;

service gRPCConfigOper {

rpc Subscribe(SubsArgs) returns(stream SubsReply) {};

rpc Cancel(CancelArgs) returns(CancelReply) {};

}

message Path {

string path = 1;

}

message SubsArgs {

uint64 request\_id = 1;

uint32 encoding = 2;

repeated Destination destination = 3;

//Path to a section of operational state of interest (the sensor).

repeated Path path = 5;

//The accuracy of the collected data reported by the device

//to the collector

uint64 sample\_interval = 6;

//Maximum time interval in seconds that may pass

//between updates from a device to a telemetry collector.

//If this interval expires, but there is no updated data to

//send (such as if suppress\_updates has been configured), the

//device must send a telemetry message to the collector.

uint64 heartbeat\_interval = 7;

//Boolean flag to control suppression of redundant

//telemetry updates to the collector platform. If this flag is

//set to TRUE, then the collector will only send an update at

//the configured interval if a subscribed data value has

//changed. Otherwise, the device will not send an update to

//the collector until expiration of the heartbeat interval.

bool suppress\_redundant = 8;

//range "0..63"

//The dscp type represents a Differentiated Services Code Point

//that may be used for marking packets in a traffic stream.

//In the value set and its semantics, this type is equivalent

//to the Dscp textual convention of the SMIv2.

uint32 originated\_qos\_marking = 9;

}

message Destination {

// IP address of the telemetry stream destination

string destination\_address = 1;

// Protocol (udp or tcp) port number for the telemetry

// stream destination

uint32 destination\_port = 2;

}

message SubsReply {

uint32 subscription\_id = 1;

uint64 request\_id = 2;

string response\_code = 3;

//containing the data described by the Telemetry message

//defined in hw\_telemetry.proto

bytes message = 4;

}

message CancelArgs {

uint64 request\_id = 1;

uint32 subscription\_id = 2;

}

message CancelReply {

uint64 request\_id = 1;

string response\_code = 2;

string message = 3;

}

1. grpc-dialout.proto

syntax = "proto3";

package dialout;

service gRPCDataservice {

rpc dataPublish(stream serviceArgs) returns(stream serviceArgs) {};

}

message serviceArgs {

int64 ReqId = 1;

bytes data = 2;

string errors = 3;

}

1. telemetry层数据模型
2. telemetry.proto

syntax = "proto3"; //proto版本定义为v3版本。

package telemetry; //本包名称为telemetry。

message Telemetry { //telemetry消息结构定义。

string node\_id\_str = 1; //设备的hostname，为设备在网络中的唯一标识，用户可以配置修改，GPB编码时编码为1。

string subscription\_id\_str = 2; //订阅名称，静态配置订阅时的订阅名称，GPB编码时编码为2。

string sensor\_path = 3; //订阅路径，GPB编码时编码为3。

uint64 collection\_id = 4; //标识采样轮次，GPB编码时编码为4。

uint64 collection\_start\_time = 5; //标识采样轮次开始时间，GPB编码时编码为5。

uint64 msg\_timestamp = 6; //生成本消息的时间戳，GPB编码时编码为6。

TelemetryGPBTable data\_gpb = 7; //承载的数据，由TelemetryGPBTable定义，GPB编码时编码为7。

uint64 collection\_end\_time = 8; //标识采样轮次结束时间，GPB编码时编码为8。

uint32 current\_period = 9; //采样精度，单位是毫秒，GPB编码时编码为9。

string except\_desc = 10; //异常描述信息，采样异常时用于上报异常信息，GPB编码时编码为10。

string product\_name = 11; //产品形态名

Encoding encoding = 12; //数据编码。为GPB时，data\_gpb字段有效，否则时data\_str字段有效

enum Encoding {

Encoding\_GPB = 0;

};

}

message TelemetryGPBTable { //TelemetryGPBTable消息结构定义。

repeated TelemetryRowGPB row = 1; //数组定义，标识数据是TelemetryRowGPB结构的重复，GPB编码时编码为1。

}

message TelemetryRowGPB {

uint64 timestamp = 1; //采样当前实例的时间戳，GPB编码时编码为1。

bytes content = 11; //承载的采样实例数据，GPB编码时编码为11，需要结合sensor\_path字段，才可以判断此处会以哪个proto文件编码。

}

1. 业务数据层数据模型
2. an-bb-queue-kpi.proto

syntax = "proto3";

package an\_bb\_queue\_kpi;

message QueueKpiRecords {

message QueueKpiRecord {

//端口名称，格式：gpon.f.s.p，epon.f.s.p，ethernetCsmacd.f.s.p等。

string name = 1 [json\_name = "name"];

//端口下的通道名称。

string channel = 2 [json\_name = "channel"];

//队列索引，范围是0~7。

uint32 index = 3 [json\_name = "index"];

//队列转发的字节数。

uint64 pass\_bytes = 4 [json\_name = "pass-bytes"];

//队列转发的报文数。

uint64 pass\_packets = 5 [json\_name = "pass-packets"];

//队列丢弃的报文数。

uint64 drop\_packets = 6 [json\_name = "drop-packets"];

//队列转发的绿色流量的字节数。硬件不支持时此处填写无效值全F。

uint64 pass\_green\_bytes = 7 [json\_name = "pass-green-bytes"];

//队列转发的绿色流量的报文数。硬件不支持时此处填写无效值全F。

uint64 pass\_green\_packets = 8 [json\_name = "pass-green-packets"];

//队列丢弃的绿色流量的报文数。硬件不支持时此处填写无效值全F。

uint64 drop\_green\_packets = 9 [json\_name = "drop-green-packets"];

//队列转发的黄色流量的字节数。硬件不支持时此处填写无效值全F。

uint64 pass\_yellow\_bytes = 10 [json\_name = "pass-yellow-bytes"];

//队列转发的黄色流量的报文数。硬件不支持时此处填写无效值全F。

uint64 pass\_yellow\_packets = 11 [json\_name = "pass-yellow-packets"];

//队列丢弃的黄色流量的报文数。硬件不支持时此处填写无效值全F。

uint64 drop\_yellow\_packets = 12 [json\_name = "drop-yellow-packets"];

//队列丢弃的未识别颜色的报文数。硬件不支持时此处填写无效值全F。

uint64 drop\_unknown\_color\_packets = 13 [json\_name = "drop-unknown-color-packets"];

}

repeated QueueKpiRecord queue\_kpi\_record = 1 [json\_name = "queue-kpi-record"];

}

1. an-bb-service-flow-kpi.proto

syntax = "proto3";

package an\_bb\_service\_flow\_kpi;

message ServiceFlowKpiRecords {

message ServiceFlowKpiRecord {

//vlan id

string name = 1 [json\_name = "name"];

//下行方向业务流丢包总数。

uint64 downstream\_flow\_drop\_cnt = 2 [json\_name = "downstream-flow-drop-cnt"];

//下行方向入业务流通过的总包数。

uint64 downstream\_flow\_pass\_cnt = 3 [json\_name = "downstream-flow-pass-cnt"];

//下行方向业务流秒级丢包最大数。

uint32 downstream\_flow\_drop\_max = 4 [json\_name = "downstream-flow-drop-max"];

//下行方向业务流秒级丢包最小数。

uint32 downstream\_flow\_drop\_min = 5 [json\_name = "downstream-flow-drop-min"];

//下行方向业务流秒级丢包最大丢包率。与最大丢包无时间对应关系。单位10^-5。

uint32 downstream\_flow\_drop\_rate\_max = 6 [json\_name = "downstream-flow-drop-rate-max"];

//下行方向业务流秒级丢包最小丢包率。与最小丢包无时间对应关系。单位10^-5。

uint32 downstream\_flow\_drop\_rate\_min = 7 [json\_name = "downstream-flow-drop-rate-min"];

//下行方向业务流发生丢包的秒数的累加。

uint32 downstream\_flow\_drop\_seconds\_cnt = 8 [json\_name = "downstream-flow-drop-seconds-cnt"];

//下行方向业务流通过的字节数。

uint64 downstream\_flow\_pass\_bytes = 9 [json\_name = "downstream-flow-pass-bytes"];

//统计周期内，下行方向平均速率。单位kbps。

uint32 downstream\_mfr\_avg = 10 [json\_name = "downstream-mfr-avg"];

//上行方向通过的字节数

uint64 upstream\_pass\_bytes = 11 [json\_name = "upstream-pass-bytes"];

//上行方向通过的总包数

uint64 upstream\_pass\_cnt = 12 [json\_name = "upstream-pass-cnt"];

//上行方向丢包总数

uint64 upstream\_drop\_cnt = 13 [json\_name = "upstream-drop-cnt"];

}

repeated ServiceFlowKpiRecord service\_flow\_kpi\_record = 1 [json\_name = "service-flow-kpi-record"];

}

1. an-ethernet-kpi.proto

syntax = "proto3";

package an\_ethernet\_kpi;

message EthernetPortKpiRecords {

message EthernetPortKpiRecord {

//ethernetCsmacd.f.s.p

string name = 1 [json\_name = "name"];

//以太接口发送的字节数

uint64 port\_tx\_bytes = 2 [json\_name = "port-tx-bytes"];

//以太接口接收的字节数

uint64 port\_rx\_bytes = 3 [json\_name = "port-rx-bytes"];

//以太接口发送的帧数

uint64 port\_tx\_packets = 4 [json\_name = "port-tx-packets"];

//以太接口接收的帧数

uint64 port\_rx\_packets = 5 [json\_name = "port-rx-packets"];

//以太接口发送方向丢弃的帧数，一般是MAC层校验错误的帧数

uint64 port\_tx\_discard\_packets = 6 [json\_name = "port-tx-discard-packets"];

//以太接口接收方向丢弃的帧数，一般是MAC层校验错误的帧数

uint64 port\_rx\_discard\_packets = 7 [json\_name = "port-rx-discard-packets"];

//以太接口接收的对齐错误的帧数

uint64 port\_rx\_alignment\_error\_packets = 8 [json\_name = "port-rx-alignment-error-packets"];

//以太接口发送方向的CRC错误帧数

uint64 port\_tx\_crc\_error\_packets = 9 [json\_name = "port-tx-crc-error-packets"];

//以太接口接收方向的CRC错误帧数

uint64 port\_rx\_crc\_error\_packets = 10 [json\_name = "port-rx-crc-error-packets"];

//以太接口发送的超长帧数

uint64 port\_tx\_oversized\_packets = 11 [json\_name = "port-tx-oversized-packets"];

//以太接口接收的超长帧数

uint64 port\_rx\_oversized\_packets = 12 [json\_name = "port-rx-oversized-packets"];

//以太接口发送的超短帧数 （帧长小于64并且CRC正确的包数)

uint64 port\_tx\_undersized\_packets = 13 [json\_name = "port-tx-undersized-packets"];

//以太接口接收的超短帧数

uint64 port\_rx\_undersized\_packets = 14 [json\_name = "port-rx-undersized-packets"];

//以太接口发送的分片帧数 (帧长小于64并且CRC错误的包数)

uint64 port\_tx\_fragment\_packets = 15 [json\_name = "port-tx-fragment-packets"];

//以太接口接收的分片帧数

uint64 port\_rx\_fragment\_packets = 16 [json\_name = "port-rx-fragment-packets"];

//以太接口发送的Jabber帧数

uint64 port\_tx\_jabber\_packets = 17 [json\_name = "port-tx-jabber-packets"];

//以太接口接收的Jabber帧数

uint64 port\_rx\_jabber\_packets = 18 [json\_name = "port-rx-jabber-packets"];

//以太接口发送的错误帧数

uint64 port\_tx\_error\_packets = 19 [json\_name = "port-tx-error-packets"];

//以太接口接收的错误帧数

uint64 port\_rx\_error\_packets = 20 [json\_name = "port-rx-error-packets"];

//以太端口下行平均速率，单位kbps

uint64 port\_tx\_rate = 21 [json\_name = "port-tx-rate"];

//以太端口上行平均速率，单位kbps

uint64 port\_rx\_rate = 22 [json\_name = "port-rx-rate"];

//以太端口下行秒级峰值速率，单位kbps

uint64 port\_tx\_peak\_rate = 23 [json\_name = "port-tx-peak-rate"];

//以太端口上行秒级峰值速率，单位kbps

uint64 port\_rx\_peak\_rate = 24 [json\_name = "port-rx-peak-rate"];

}

repeated EthernetPortKpiRecord ethernet\_port\_kpi\_record = 1 [json\_name = "ethernet-port-kpi-record"];

}

1. an-epon-onu-transceivers.proto

syntax = "proto3";

package an\_epon\_onu\_transceivers;

message EponOnuTransceivers {

message EponOnuTransceiver {

//ONU名称，格式：v-ani.f.s.p.onuid

string name = 1 [json\_name = "name"];

//接收光功率，单位0.01dBm

int32 optical\_unit\_rx\_power = 2 [json\_name = "optical-unit-rx-power"];

//发送光功率，单位0.01dBm

int32 optical\_unit\_tx\_power = 3 [json\_name = "optical-unit-tx-power"];

//端口的偏置电流， 单位：0.01mA。

int32 optical\_unit\_laser\_bias\_current = 4 [json\_name = "optical-unit-laser-bias-current"];

//光模块当前的温度，单位：0.01℃。

int32 optical\_unit\_temperature = 5 [json\_name = "optical-unit-temperature"];

//光模块当前的电压，单位是：0.01V。

int32 optical\_unit\_voltage = 6 [json\_name = "optical-unit-voltage"];

//光模块类型，0:unknown,1:epon,2:10g-epon-s,3:10g-epon-a,-1:invalid

int32 module\_type = 7 [json\_name = "module-type"];

//光模块子类型

string module\_sub\_type = 8 [json\_name = "module-sub-type"];

}

repeated EponOnuTransceiver epon\_onu\_transceiver = 1 [json\_name = "epon-onu-transceiver"];

}

1. an-epon-pm-olt-traffic.proto

syntax = "proto3";

package an\_epon\_pm\_olt\_traffic;

message EponPmOltTraffics {

message EponPmOltTraffic {

//端口ID名称，格式：epon.f.s.p

string name = 1 [json\_name = "name"];

//EPON端口发送的以太帧字节数

uint64 port\_tx\_bytes = 2 [json\_name = "port-tx-bytes"];

//EPON端口接收的以太帧字节数

uint64 port\_rx\_bytes = 3 [json\_name = "port-rx-bytes"];

//EPON端口发送的以太帧数

uint64 port\_tx\_pkt = 4 [json\_name = "port-tx-pkt"];

//EPON端口接收的以太帧数

uint64 port\_rx\_pkt = 5 [json\_name = "port-rx-pkt"];

//EPON端口下行队列丢包统计

uint64 port\_tx\_discard\_pkt = 6 [json\_name = "port-tx-discard-pkt"];

//EPON端口口接收方向丢弃的以太帧数，一般是MAC层校验错误的帧数

uint64 port\_rx\_discard\_pkt = 7 [json\_name = "port-rx-discard-pkt"];

//EPON端口接收方向的以太帧CRC错误帧数

uint64 port\_rx\_crc\_error\_pkt = 8 [json\_name = "port-rx-crc-error-pkt"];

//EPON端口接收方向丢弃的以太帧超长帧数

uint64 port\_rx\_oversized\_discard\_pkt = 9 [json\_name = "port-rx-oversized-discard-pkt"];

//EPON端口接收方向丢弃的以太帧超短帧数

uint64 port\_rx\_undersized\_discard\_pkt = 10 [json\_name = "port-rx-undersized-discard-pkt"];

//EPON端口接收的以太帧错误帧数

uint64 port\_rx\_error\_pkt = 11 [json\_name = "port-rx-error-pkt"];

//EPON端口下行平均速率，单位kbps

uint32 port\_tx\_rate = 12 [json\_name = "port-tx-rate"];

//EPON端口上行平均速率，单位kbps

uint32 port\_rx\_rate = 13 [json\_name = "port-rx-rate"];

//EPON端口下行秒级峰值速率，单位kbps

uint32 port\_tx\_peak\_rate = 14 [json\_name = "port-tx-peak-rate"];

//EPON端口上行秒级峰值速率，单位kbps

uint32 port\_rx\_peak\_rate = 15 [json\_name = "port-rx-peak-rate"];

}

repeated EponPmOltTraffic pm\_olt\_traffic = 1 [json\_name = "epon-pm-olt-traffic"];

}

message EponPmOltChannelTraffics {

message EponPmOltChannelTraffic {

//端口ID名称，格式：epon.f.s.p

string name = 1 [json\_name = "name"];

//通道号

uint32 channel = 2 [json\_name = "channel"];

//EPON端口发送的以太帧字节数

uint64 port\_tx\_bytes = 3 [json\_name = "port-tx-bytes"];

//EPON端口接收的以太帧字节数

uint64 port\_rx\_bytes = 4 [json\_name = "port-rx-bytes"];

//EPON端口发送的以太帧数

uint64 port\_tx\_pkt = 5 [json\_name = "port-tx-pkt"];

//EPON端口接收的以太帧数

uint64 port\_rx\_pkt = 6 [json\_name = "port-rx-pkt"];

//EPON端口下行队列丢包统计

uint64 port\_tx\_discard\_pkt = 7 [json\_name = "port-tx-discard-pkt"];

//EPON端口口接收方向丢弃的以太帧数，一般是MAC层校验错误的帧数

uint64 port\_rx\_discard\_pkt = 8 [json\_name = "port-rx-discard-pkt"];

//EPON端口接收方向的以太帧CRC错误帧数

uint64 port\_rx\_crc\_error\_pkt = 9 [json\_name = "port-rx-crc-error-pkt"];

//EPON端口接收方向丢弃的以太帧超长帧数

uint64 port\_rx\_oversized\_discard\_pkt = 10 [json\_name = "port-rx-oversized-discard-pkt"];

//EPON端口接收方向丢弃的以太帧超短帧数

uint64 port\_rx\_undersized\_discard\_pkt = 11 [json\_name = "port-rx-undersized-discard-pkt"];

//EPON端口接收的以太帧错误帧数

uint64 port\_rx\_error\_pkt = 12 [json\_name = "port-rx-error-pkt"];

//EPON端口下行平均速率，单位kbps

uint32 port\_tx\_rate = 13 [json\_name = "port-tx-rate"];

//EPON端口上行平均速率，单位kbps

uint32 port\_rx\_rate = 14 [json\_name = "port-rx-rate"];

//EPON端口下行秒级峰值速率，单位kbps

uint32 port\_tx\_peak\_rate = 15 [json\_name = "port-tx-peak-rate"];

//EPON端口上行秒级峰值速率，单位kbps

uint32 port\_rx\_peak\_rate = 16 [json\_name = "port-rx-peak-rate"];

}

repeated EponPmOltChannelTraffic pm\_olt\_channel\_traffic =1 [json\_name = "epon-pm-olt-channel-traffic"];

}

1. an-epon-pm-olt-transceivers.proto

syntax = "proto3";

package an\_epon\_pm\_olt\_transceivers;

message EponPmOltTransceivers {

message EponPmOltTransceiver {

//端口ID名称，格式：epon.f.s.p

string name = 1 [json\_name = "name"];

//光模块温度，单位0.01℃

int32 temperature = 2 [json\_name = "temperature"];

//光模块供电电压， 单位0.01V

int32 supply\_voltage = 3 [json\_name = "supply-voltage"];

//通道1类型

int32 channel\_1\_type = 4 [json\_name = "channel-0-type"];

//通道1发送偏置电流，单位0.01mA

int32 channel\_1\_tx\_bias = 5 [json\_name = "channel-0-tx-bias"];

//通道1发送功率，单位0.01dBm，

int32 channel\_1\_tx\_power = 6 [json\_name = "channel-0-tx-power"];

//通道1空闲时刻的光功率，单位0.01dBm

int32 channel\_1\_idle\_rssi = 7 [json\_name = "channel-0-idle-rssi"];

//通道2类型

int32 channel\_2\_type = 8 [json\_name = "channel-1-type"];

//通道2发送偏置电流，单位0.01mA

int32 channel\_2\_tx\_bias = 9 [json\_name = "channel-1-tx-bias"];

//通道2发送功率，单位0.01dBm

int32 channel\_2\_tx\_power = 10 [json\_name = "channel-1-tx-power"];

//通道2空闲时刻的光功率，单位0.01dBm

int32 channel\_2\_idle\_rssi = 11 [json\_name = "channel-1-idle-rssi"];

//光模块类型,0:unknown,1:epon,2:10g-epon-s,3:10g-epon-a,-1:invalid

int32 module\_type = 12 [json\_name = "module-type"];

string module\_sub\_type = 13 [json\_name = "module-sub-type"];

}

repeated EponPmOltTransceiver epon\_pm\_olt\_transceiver = 1 [json\_name = "epon-pm-olt-transceiver"];

}

1. an-epon-pm-onu-local-info.proto

syntax = "proto3";

package an\_epon\_pm\_onu\_local\_info;

message EponPmOnuLocalInfos {

message EponPmOnuLocalInfo {

//ONU名称，格式：v-ani.f.s.p.onuid

string name = 1 [json\_name = "name"];

//OLT光模块接收到ONU的光功率，单位：0.01dBm

int32 olt\_rx\_power = 2 [json\_name = "olt-rx-power"];

//在线时长，单位秒

uint32 online\_duration = 3 [json\_name = "online-duration"];

//最后一次下线时间，从1970年开始的秒数

uint32 last\_down\_time = 4 [json\_name = "last-down-time"];

//最后一次下线原因

//The value (0) indicates that the reason is unknown，

//The value (1) indicates that the reason is LOS(Loss of signal)，

//The value (2) indicates that the reason is LOSi(Loss of signal for ONUi) or LOBi (Loss of burst for ONUi)，

//The value (3) indicates that the reason is LOFI(Loss of frame of ONUi)，

//The value (4) indicates that the reason is SFI(Signal fail of ONUi)，

//The value (5) indicates that the reason is LOAI(Loss of acknowledge with ONUi)，

//The value (6) indicates that the reason is LOAMI(Loss of PLOAM for ONUi)，

//The value (7) indicates that the reason is deactive ONT fails，

//The value (8) indicates that the reason is deactive ONT success，

//The value (9) indicates that the reason is reset ONT，

//The value (10) indicates that the reason is re-register ONT，

//The value (11) indicates that the reason is pop up fail，

//The value (13) indicates that the reason is dying-gasp，

//The value (15) indicates that the reason is LOKI(Loss of key synch with ONUi)，

//The value (18) indicates that the reason is deactived ONT due to the ring，

//The value (30) indicates that the reason is shut down ONT optical module，

//The value (255) indicates that the query fails.

uint32 last\_down\_cause = 5 [json\_name = "last-down-cause"];

//ONU状态，1-online，2-offline，255-invalid

uint32 onu\_status = 6 [json\_name = "onu-status"];

}

repeated EponPmOnuLocalInfo epon\_pm\_onu\_local\_info = 1 [json\_name = "epon-pm-onu-local-info"];

}

1. an-epon-pm-onu-remote-info.proto

syntax = "proto3";

package an\_epon\_pm\_onu\_remote\_info;

message EponPmOnuRemoteInfos {

message EponPmOnuRemoteInfo {

//ONU名称，格式：v-ani.f.s.p.onuid

string name = 1 [json\_name = "name"];

//内存利用率，单位%

uint32 mem\_occup = 2 [json\_name = "mem-occup"];

//CPU利用率，单位%

uint32 cpu\_occup = 3 [json\_name = "cpu-occup"];

//CPU的温度，单位℃

int32 cpu\_temp = 4 [json\_name = "cpu-temp"];

//ONU PON口发送报文个数

uint64 onu\_pon\_send\_packets = 5 [json\_name = "onu-pon-send-packets"];

//ONU PON口接收报文个数

uint64 onu\_pon\_recv\_packets = 6 [json\_name = "onu-pon-recv-packets"];

//ONU PON口接收报文错误个数

uint64 onu\_pon\_recv\_errors\_packets = 7 [json\_name = "onu-pon-recv-errors-packets"];

}

repeated EponPmOnuRemoteInfo epon\_pm\_onu\_remote\_info = 1 [json\_name = "epon-pm-onu-remote-info"];

}

1. an-epon-pm-onu-traffic.proto

syntax = "proto3";

package an\_epon\_pm\_onu\_traffic;

message EponPmOnuTraffics {

message EponPmOnuTraffic {

//ONU名称，格式：v-ani.f.s.p.onuid

string name = 1 [json\_name = "name"];

//ONU下行平均速率，单位kbps

uint32 tx\_rate = 2 [json\_name = "tx-rate"];

//ONU上行平均速率，单位kbps

uint32 rx\_rate = 3 [json\_name = "rx-rate"];

//ONU下行秒级峰值速率，单位kbps

uint32 tx\_peak\_rate = 4 [json\_name = "tx-peak-rate"];

//ONU上行秒级峰值速率，单位kbps

uint32 rx\_peak\_rate = 5 [json\_name = "rx-peak-rate"];

}

repeated EponPmOnuTraffic epon\_pm\_onu\_traffic = 1 [json\_name = "epon-pm-onu-traffic"];

}

1. an-gpon-onu-transceivers.proto

syntax = "proto3";

package an\_gpon\_onu\_transceivers;

message GponOnuTransceivers {

message GponOnuTransceiver {

//ONU名称，格式：v-ani.f.s.p.onuid

string name = 1 [json\_name = "name"];

//接收光功率，单位0.01dBm

int32 optical\_unit\_rx\_power = 2 [json\_name = "optical-unit-rx-power"];

//发送光功率，单位0.01dBm

int32 optical\_unit\_tx\_power = 3 [json\_name = "optical-unit-tx-power"];

//端口的偏置电流， 单位：0.01mA。

int32 optical\_unit\_laser\_bias\_current = 4 [json\_name = "optical-unit-laser-bias-current"];

//光模块当前的温度，单位：0.01℃。

int32 optical\_unit\_temperature = 5 [json\_name = "optical-unit-temperature"];

//光模块当前的电压，单位是：0.01V。

int32 optical\_unit\_voltage = 6 [json\_name = "optical-unit-voltage"];

//光模块类型,0:unknown,1:gpon,2:xg-pon,3:xgs-pon,4:gpon-and-xgpon,5:gpon-and-xgspon,-1:invalid

int32 module\_type = 7 [json\_name = "module-type"];

//光模块子类型

string module\_sub\_type = 8 [json\_name = "module-sub-type"];

}

repeated GponOnuTransceiver gpon\_onu\_transceiver = 1 [json\_name = "gpon-onu-transceiver"];

}

1. an-gpon-pm-olt-traffic.proto

syntax = "proto3";

package an\_gpon\_pm\_olt\_traffic;

message GponPmOltTraffics {

message GponPmOltTraffic {

//端口ID名称，格式：gpon.f.s.p

string name = 1 [json\_name = "name"];

//GPON端口发送的以太帧字节数

uint64 port\_tx\_bytes = 2 [json\_name = "port-tx-bytes"];

//GPON端口接收的以太帧字节数

uint64 port\_rx\_bytes = 3 [json\_name = "port-rx-bytes"];

//GPON端口发送的以太帧数

uint64 port\_tx\_pkt = 4 [json\_name = "port-tx-pkt"];

//GPON端口接收的以太帧数

uint64 port\_rx\_pkt = 5 [json\_name = "port-rx-pkt"];

//GPON端口下行队列丢包统计

uint64 port\_tx\_discard\_pkt = 6 [json\_name = "port-tx-discard-pkt"];

//GPON端口口接收方向丢弃的以太帧数，一般是MAC层校验错误的帧数

uint64 port\_rx\_discard\_pkt = 7 [json\_name = "port-rx-discard-pkt"];

//GPON端口接收方向的以太帧CRC错误帧数

uint64 port\_rx\_crc\_error\_pkt = 8 [json\_name = "port-rx-crc-error-pkt"];

//GPON端口接收方向丢弃的以太帧超长帧数

uint64 port\_rx\_oversized\_discard\_pkt = 9 [json\_name = "port-rx-oversized-discard-pkt"];

//GPON端口接收方向丢弃的以太帧超短帧数

uint64 port\_rx\_undersized\_discard\_pkt = 10 [json\_name = "port-rx-undersized-discard-pkt"];

//GPON端口接收的以太帧错误帧数

uint64 port\_rx\_error\_pkt = 11 [json\_name = "port-rx-error-pkt"];

//GPON端口下行平均速率，单位kbps

uint32 port\_tx\_rate = 12 [json\_name = "port-tx-rate"];

//GPON端口上行平均速率，单位kbps

uint32 port\_rx\_rate = 13 [json\_name = "port-rx-rate"];

//GPON端口下行秒级峰值速率，单位kbps

uint32 port\_tx\_peak\_rate = 14 [json\_name = "port-tx-peak-rate"];

//GPON端口上行秒级峰值速率，单位kbps

uint32 port\_rx\_peak\_rate = 15 [json\_name = "port-rx-peak-rate"];

}

repeated gponPmOltTraffic pm\_olt\_traffic = 1 [json\_name = "gpon-pm-olt-traffic"];

}

message GponPmOltChannelTraffics {

message GponPmOltChannelTraffic {

//端口ID名称，格式：gpon.f.s.p

string name = 1 [json\_name = "name"];

//通道号

uint32 channel = 2 [json\_name = "channel"];

//GPON端口发送的以太帧字节数

uint64 port\_tx\_bytes = 3 [json\_name = "port-tx-bytes"];

//GPON端口接收的以太帧字节数

uint64 port\_rx\_bytes = 4 [json\_name = "port-rx-bytes"];

//GPON端口发送的以太帧数

uint64 port\_tx\_pkt = 5 [json\_name = "port-tx-pkt"];

//GPON端口接收的以太帧数

uint64 port\_rx\_pkt = 6 [json\_name = "port-rx-pkt"];

//GPON端口下行队列丢包统计

uint64 port\_tx\_discard\_pkt = 7 [json\_name = "port-tx-discard-pkt"];

//GPON端口口接收方向丢弃的以太帧数，一般是MAC层校验错误的帧数

uint64 port\_rx\_discard\_pkt = 8 [json\_name = "port-rx-discard-pkt"];

//GPON端口接收方向的以太帧CRC错误帧数

uint64 port\_rx\_crc\_error\_pkt = 9 [json\_name = "port-rx-crc-error-pkt"];

//GPON端口接收方向丢弃的以太帧超长帧数

uint64 port\_rx\_oversized\_discard\_pkt = 10 [json\_name = "port-rx-oversized-discard-pkt"];

//GPON端口接收方向丢弃的以太帧超短帧数

uint64 port\_rx\_undersized\_discard\_pkt = 11 [json\_name = "port-rx-undersized-discard-pkt"];

//GPON端口接收的以太帧错误帧数

uint64 port\_rx\_error\_pkt = 12 [json\_name = "port-rx-error-pkt"];

//GPON端口下行平均速率，单位kbps

uint32 port\_tx\_rate = 13 [json\_name = "port-tx-rate"];

//GPON端口上行平均速率，单位kbps

uint32 port\_rx\_rate = 14 [json\_name = "port-rx-rate"];

//GPON端口下行秒级峰值速率，单位kbps

uint32 port\_tx\_peak\_rate = 15 [json\_name = "port-tx-peak-rate"];

//GPON端口上行秒级峰值速率，单位kbps

uint32 port\_rx\_peak\_rate = 16 [json\_name = "port-rx-peak-rate"];

}

repeated GponPmOltChannelTraffic pm\_olt\_channel\_traffic =1 [json\_name = "gpon-pm-olt-channel-traffic"];

}

1. an-gpon-pm-olt-transceivers.proto

syntax = "proto3";

package an\_gpon\_olt\_transceivers;

message GponOltTransceivers {

message GponOltTransceiver {

//端口ID名称，格式：Gpon.f.s.p

string name = 1 [json\_name = "name"];

//光模块温度，单位0.01℃

int32 temperature = 2 [json\_name = "temperature"];

//光模块供电电压， 单位0.01V

int32 supply\_voltage = 3 [json\_name = "supply-voltage"];

//通道1类型

int32 channel\_1\_type = 4 [json\_name = "channel-0-type"];

//通道1发送偏置电流，单位0.01mA

int32 channel\_1\_tx\_bias = 5 [json\_name = "channel-0-tx-bias"];

//通道1发送功率，单位0.01dBm，

int32 channel\_1\_tx\_power = 6 [json\_name = "channel-0-tx-power"];

//通道1空闲时刻的光功率，单位0.01dBm

int32 channel\_1\_idle\_rssi = 7 [json\_name = "channel-0-idle-rssi"];

//通道2类型

int32 channel\_2\_type = 8 [json\_name = "channel-1-type"];

//通道2发送偏置电流，单位0.01mA

int32 channel\_2\_tx\_bias = 9 [json\_name = "channel-1-tx-bias"];

//通道2发送功率，单位0.01dBm

int32 channel\_2\_tx\_power = 10 [json\_name = "channel-1-tx-power"];

//通道2空闲时刻的光功率，单位0.01dBm

int32 channel\_2\_idle\_rssi = 11 [json\_name = "channel-1-idle-rssi"];

//光模块类型,0:unknown,1:gpon,2:xg-pon,3:xgs-pon,4:gpon-and-xgpon,5:gpon-and-xgspon,-1:invalid

int32 module\_type = 12 [json\_name = "module-type"];

string module\_sub\_type = 13 [json\_name = "module-sub-type"];

}

repeated GponOltTransceiver gpon\_olt\_transceiver = 1 [json\_name = "gpon-pm-olt-transceiver"];

}

1. an-gpon-pm-onu-line-quality.proto

syntax = "proto3";

package an\_gpon\_pm\_onu\_line\_quality;

message GponPmOnuLineQualitys {

message GponPmOnuLineQuality {

//ONU名称，格式：v\_ani.f.s.p.onuid

string name = 1 [json\_name = "name"];

//OLT针对该ONU的帧丢失指示（LOFi，Loss of Frame of one specific ONU）告警次数。GPON ONU有效，XGPON ONU不支持

uint32 lofi\_alarm\_count = 2 [json\_name = "lofi-alarm-count"];

//ONU上线后，上行帧定界错误次数。GPON ONU有效，XGPON ONU不支持

uint32 dowi\_alarm\_count = 3 [json\_name = "dowi-alarm-count"];

//ONU上线后，上行帧定界错误次数。GPON ONU有效，XGPON ONU不支持

uint32 upstream\_delimiter\_error\_count = 4 [json\_name = "upstream-delimiter-error-count"];

//OLT启动后，上行帧比特交叉奇偶校验错误次数。GPON/XGPON ONU有效。

uint64 upstream\_bip\_error\_count = 5 [json\_name = "upstream-bip-error-count"];

//下行帧比特交叉奇偶校验错误次数。GPON ONU有效，XGPON ONU不支持。该Counter虽然由ONU侧检测，但GPON协议中通过

//REI PLOAM消息上传后，实际数据由OLT从REI消息中的参数值进行累计，而ONU侧无对应OMCI MIB进行统计。因此将该参数

//纳入近端数据。

uint64 downstream\_bip\_error\_count = 6 [json\_name = "downstream-bip-error-count"];

//OLT启动后，上行前向纠错（FEC，Forward Error Correction）解码纠正块数。GPON/XGPON ONU有效。

uint64 upstream\_fec\_block = 7 [json\_name = "upstream-fec-block"];

//OLT启动后，上行FEC解码未纠正块数。GPON/XGPON ONU有效。

uint64 upstream\_fec\_error\_block = 8 [json\_name = "upstream-fec-error-block"];

//上行接收code words总数,GPON/XGPON ONU有效

uint64 upstream\_fec\_total\_block = 9 [json\_name = "upstream-fec-total-block"];

//上行FEC校正字节数。GPON/XGPON ONU有效

uint64 upstream\_fec\_byte = 10 [json\_name = "upstream-fec-byte"];

//OLT启动后，上行信头差错控制码（HEC，Header Error Code）校验错误次数。GPON/XGPON ONU有效

uint64 upstream\_hec\_error\_count = 11 [json\_name = "upstream-hec-error-count"];

//上行GEM帧数，GPON/XGPON ONU有效

uint64 upstream\_gem\_count = 12 [json\_name = "upstream-gem-count"];

//ONU LOSi告警次数，GPON/XGPON ONU有效

uint32 losi\_alarm\_count = 13 [json\_name = "losi-alarm-count"];

//ONU DGi告警次数，GPON/XGPON ONU有效

uint32 dgi\_alarm\_count = 14 [json\_name = "dgi-alarm-count"];

}

repeated GponPmOnuLineQuality gpon\_pm\_onu\_line\_quality = 1 [json\_name = "gpon-pm-onu-line-quality"];

}

1. an-gpon-pm-onu-local-info.proto

syntax = "proto3";

package an\_gpon\_pm\_onu\_local\_info;

message GponPmOnuLocalInfos {

message GponPmOnuLocalInfo {

//ONU名称，格式：v\_ani.f.s.p.onuid

string name = 1 [json\_name = "name"];

//OLT光模块接收到ONU的光功率，单位：0.01dBm

int32 olt\_rx\_power = 2 [json\_name = "olt-rx-power"];

//在线时长，单位秒

uint32 online\_duration = 3 [json\_name = "online-duration"];

//最后一次下线时间，从1970年开始的秒数

uint32 last\_down\_time = 4 [json\_name = "last-down-time"];

//最后一次下线原因

//The value (0) indicates that the reason is unknown，

//The value (1) indicates that the reason is LOS(Loss of signal)，

//The value (2) indicates that the reason is LOSi(Loss of signal for ONUi) or LOBi (Loss of burst for ONUi)，

//The value (3) indicates that the reason is LOFI(Loss of frame of ONUi)，

//The value (4) indicates that the reason is SFI(Signal fail of ONUi)，

//The value (5) indicates that the reason is LOAI(Loss of acknowledge with ONUi)，

//The value (6) indicates that the reason is LOAMI(Loss of PLOAM for ONUi)，

//The value (7) indicates that the reason is deactive ONT fails，

//The value (8) indicates that the reason is deactive ONT success，

//The value (9) indicates that the reason is reset ONT，

//The value (10) indicates that the reason is re-register ONT，

//The value (11) indicates that the reason is pop up fail，

//The value (13) indicates that the reason is dying-gasp，

//The value (15) indicates that the reason is LOKI(Loss of key synch with ONUi)，

//The value (18) indicates that the reason is deactived ONT due to the ring，

//The value (30) indicates that the reason is shut down ONT optical module，

//The value (255) indicates that the query fails.

uint32 last\_down\_cause = 5 [json\_name = "last-down-cause"];

//ONU状态，1-online，2-offline，255-invalid

uint32 onu\_status = 6 [json\_name = "onu-status"];

}

repeated GponPmOnuLocalInfo gpon\_pm\_onu\_local\_info = 1 [json\_name = "gpon-pm-onu-local-info"];

}

1. an-gpon-pm-onu-remote-info.proto

syntax = "proto3";

package an\_gpon\_pm\_onu\_remote\_info;

message GponPmOnuRemoteInfos {

message GponPmOnuRemoteInfo {

//ONU名称，格式：v\_ani.f.s.p.onuid

string name = 1 [json\_name = "name"];

//下行FEC校正字节数

uint64 ds\_fec\_corrected\_bytes = 2 [json\_name = "ds-fec-corrected-bytes"];

//下行FEC校正code words数

uint64 ds\_fec\_corrected\_words = 3 [json\_name = "ds-fec-corrected-words"];

//下行FEC不可校正code words数

uint64 ds\_fec\_uncorrected\_words = 4 [json\_name = "ds-fec-uncorrected-words"];

//下行接收code words总数

uint64 ds\_total\_rx\_code\_words = 5 [json\_name = "ds-total-rx-code-words"];

//下行FEC校正时间

uint32 ds\_fec\_seconds = 6 [json\_name = "ds-fec-seconds"];

//ONU接收到GEM HEC错误数

uint64 xgpon\_gem\_hec\_error\_count = 7 [json\_name = "xgpon-gem-hec-error-count"];

//丢弃的XG-PON GEM帧数

uint64 xgpon\_gem\_key\_error\_count = 8 [json\_name = "xgpon-gem-key-error-count"];

//内存利用率，单位%

uint32 mem\_occup = 9 [json\_name = "mem-occup"];

//CPU利用率，单位%

uint32 cpu\_occup = 10 [json\_name = "cpu-occup"];

//CPU的温度，单位℃

int32 cpu\_temp = 11 [json\_name = "cpu-temp"];

//ONU PON口发送报文个数

uint64 onu\_pon\_send\_packets = 12 [json\_name = "onu-pon-send-packets"];

//ONU PON口接收报文个数

uint64 onu\_pon\_recv\_packets = 13 [json\_name = "onu-pon-recv-packets"];

//ONU PON口接收报文错误个数

uint64 onu\_pon\_recv\_errors\_packets = 14 [json\_name = "onu-pon-recv-errors-packets"];

//ONU所有TCONT队列丢包个数

uint64 tcont\_queue\_dropped\_packets = 15 [json\_name = "tcont-queue-dropped-packets"];

//ONU所有TCONT队列转发包个数

uint64 tcont\_queue\_passing\_packets = 16 [json\_name = "tcont-queue-passing-packets"];

}

repeated GponPmOnuRemoteInfo gpon\_pm\_onu\_remote\_info = 1 [json\_name = "gpon-pm-onu-remote-info"];

}

1. an-gpon-pm-onu-traffic.proto

syntax = "proto3";

package an\_gpon\_pm\_onu\_traffic;

message GponPmOnuTraffics {

message GponPmOnuTraffic {

//ONU名称，格式：v\_ani.f.s.p.ONUid

string name = 1 [json\_name = "name"];

//ONU下行平均速率，单位kbps

uint32 tx\_rate = 2 [json\_name = "tx-rate"];

//ONU上行平均速率，单位kbps

uint32 rx\_rate = 3 [json\_name = "rx-rate"];

//ONU下行秒级峰值速率，单位kbps

uint32 tx\_peak\_rate = 4 [json\_name = "tx-peak-rate"];

//ONU上行秒级峰值速率，单位kbps

uint32 rx\_peak\_rate = 5 [json\_name = "rx-peak-rate"];

}

repeated GponPmOnuTraffic gpon\_pm\_onu\_traffic = 1 [json\_name = "gpon-pm-onu-traffic"];

}

附录C  
（规范性）  
sensor path格式

sensor path格式如下：

业务数据层数据模型文件名：message名/…/message名

其中：

“message名/message名/message名”为数据模型中的节点路径，按需求配置。

示例：采集队列流量使用an-bb-quene-kpi.proto模型中的message QueueKpiRecords，sensor path为an-bb-quene-kpi:QueueKpiRecords。

当需要采集单个采集对象时，应配合an-telemetry-ext.yang中的filter配置。

参 考 文 献

1. https://grpc.io/docs/
2. IETF draft-ietf-netconf-udp-notif-01 基于UDP传输的配置订阅（UDP-based Transport for Configured Subscriptions）