ETSI GS F5G 011 V1.1.1 (2022-11)

GROUP SPECIFICATION

Fifth Generation Fixed Network (F5G); Telemetry Framework and Requirements for Access Networks

Disclaimer

The present document has been produced and approved by the Fifth Generation Fixed Network (F5G) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference

DGS/F5G-0011Telemetry

Keywords

F5G; telemetry; YANG

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Fifth Generation Fixed Network (F5G).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document defines the F5G Telemetry Framework and Requirements for the F5G Access Network. The framework specifies the key functions and interfaces. The F5G Access Network telemetry requirements include requirements for the functions, the overall system, and the interfaces with their data models (configuration and streaming/collection).

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS F5G 004 (V1.1.1): "Fifth Generation Fixed Network (F5G); F5G Network Architecture".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] IEEE 802.3TM-2008: "IEEE Standard for information technology".

[i.2] Recommendation ITU-T G.988: "ONU management and control interface (OMCI) specification".

[i.3] Google® Developers | Protocol Buffers | Encoding.

NOTE: Available at https://developers.google.com/protocol-buffers/docs/encoding.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GS F5G 004 [1] and the following apply:

Access Network Telemetry (ANT): monitoring technology that remotely collects data in push mode from the OLT

alignment error packet: packet with bad FCS and with a non-integral number of octets

NOTE: The definition of this term comes from IEEE 802.3 [i.1].

ANT object: specific physical or logical entity in the OLT or ONU (e.g. a PON port, a service flow, etc.)

equipment sampling capability: minimum time interval for the OLT to gather the target telemetry data

NOTE: This time interval can be shorter than the sample interval.

EXAMPLE: The equipment sampling capability is x seconds, and the sample interval is y seconds. (x can be shorter than y). A single ANT object is created from the equipment sampled data according to the configuration rules.

error packet: include the following data frames:

- Correct and incorrect data frames with a frame length less than 64 bytes.
- Correct and incorrect data frames whose frame size is greater than the maximum MTU.
- Data frames with FCS errors whose frame length ranges from 64 to the maximum MTU.
- Data frames with alignment errors whose frame length ranges from 64 to the maximum MTU.

NOTE: The definition of this term comes from IEEE 802.3 [i.1].

fragment packet: packets with less than 64 octets in length, excluding framing octets but including FCS octets

NOTE 1: These packets have, and had either a bad FCS with an integral number of octets (FCS error) or a bad FCS with a non-integral number of octets (alignment error).

NOTE 2: The definition of this term comes from IEEE 802.3 [i.1].

jabber packet: packet that is greater than 1 518 octets in length, excluding framing octets but including FCS octets

NOTE 1: These packets have, and had either a bad FCS with an integral number of octets (FCS error) or a bad FCS with a non-integral number of octets (alignment error).

NOTE 2: The definition of this term comes from IEEE 802.3 [i.1].

oversized packet: packet with length greater than 1 518 octets

NOTE: The definition of this term comes from IEEE 802.3 [i.1].

sample interval: time interval for the ANT object in the Telemetry message reported by the OLT to the collector

NOTE: This value is configured by the configuration module of the telemetry system.

sample timestamp: timestamp at which the current ANT object was sampled

sensor group: group of multiple sensor paths

sensor path: data model path of the sensor, which describes the specific ANT objects for collection

service flow: service flow is a consequence of traffic classification based on the identifiers in the Ethernet packets on a physical port or logical port

NOTE 1: For example, an identifier can be a VLAN ID, which means Ethernet packets are classified based on VLANs.

NOTE 2: A service flow can also be a Layer 2 logical channel that carries services between an access node (OLT) and a subscriber (ONU).

undersized packet: packet with length less than 64 octets

NOTE: The definition of this term comes from IEEE 802.3 [i.1].

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

10G-EPON	10 Gbit/s Ethernet Passive Optical Network
AI	Artificial Intelligence
ANT	Access Network Telemetry
BER	Bit Error Ratio
BIP	Bearer Independent Protocol
CLI	Command-Line Interface
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DG	Dying Gasp
DOW	Drift Of Window
DPU	Data Pre-processing Unit
EPON	Ethernet Passive Optical Network
FCS	Frame Check Sequence
FEC	Forward Error Correction
GEM	GPON Encapsulation Mode
GNMI	gRPC® Network Management Interface
GPB®	Google® Protocol Buffer
GPON	Gigabit-Capable Passive Optical Networks
gRPC®	Google® Remote Procedure Call
HEC	Hybrid Error Correction
HTTP	Hyper Text Transfer Protocol
ID	Identity Document
IP	Internet Protocol
IPTV	Internet Protocol Television
JSON	Java Script Object Notation
LOF	Loss Of Frame
LOS	Loss Of Signal
LP	Line Protocol
MAC	Message Authentication Code
MIB	Management Information Base
ML	Machine Learning
MSB	Most Significant Bit
MTU	Maximum Transmission Unit
NE	Network Entity
NETCONF	Network Configuration Protocol
ODN	Optical Distribution Network
OLT	Optical Line Terminal
ONU	Optical Network Unit
P2MP	Point to Multipoint
PON	Passive Optical Network
RPC	Remote Procedure Call
SNI	Service Node Interface
SNMP	Simple Network Management Protocol
TCONT	Transmission - Container
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TSDB	Time Series Database
UDP	User Datagram Protocol
UNI	User Network Interface
VLAN	Virtual Local Area Network
XG	10 GigabitMAC
XG-PON	10-Gigabit-capable Passive Optical Network
XGS-PON	10-Gigabit-capable Symmetric Passive Optical Network
YANG	Yet Another Next Generation data modelling language

4 Framework of Telemetry in Access Network

4.1 Motivation and Business Drivers

Figure 1 depicts the current Access Network deployment. A traditional data pulling methods is used, such as SNMP, syslog and CLI to pull data from the OLT to monitor Access Network and troubleshoot any issues. The interface uses proprietary MIBs from different OLT equipment vendors which are difficult to automate. So, each request to pull data is resource intensive and impact the performance of the OLT, and adds complexity because there is more than one pull request per OLT. The pulling method does not efficiently scale.

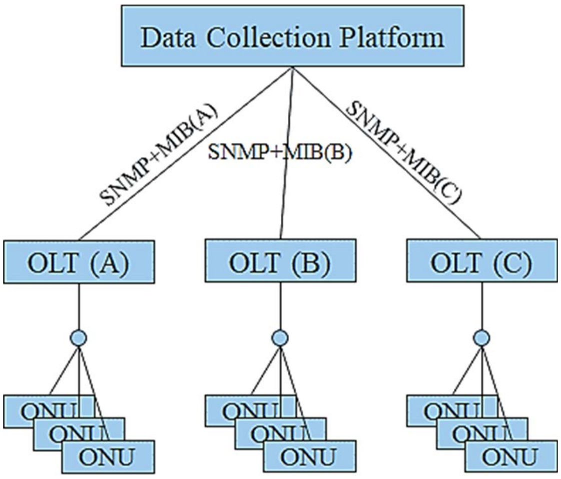


Figure 1: Traditional Access Network architecture

As the complexity of the Access Network increases, it is crucial to maintain the network health. To achieve this, the Access Network can provide better visibility compared to existing methods via automated real-time data collection. Telemetry replaces the pull method, and uses the push method to continuously stream data from the OLT and provides notifications to the data collection platform. Telemetry has the advantages of scale, speed and automation. With the flexibility of telemetry, the data of interest can be selected from the OLT and the OLT can transmit it in a structured format to a data collection platform for monitoring. In addition, the data collection platform can expose F5G Access Network information to the application layer.

Telemetry introduces finer granular data points and more frequent data streaming in the Access Network. It enables better performance monitoring and therefore better control over large Access Network. Telemetry data can assist in the prediction of network problems and take preventative actions without impacting the performance of the OLT. The operators can gain better visibility and insight into the network. The operator can enhance the network operational performance by using data analytics. Telemetry technology opens the door to big data and machine learning methods in the Access Network.

4.2 Telemetry Architecture Overview

Figure 2 illustrates the F5G Access Network architecture of the telemetry technologies. The Access Network equipment supports the telemetry collection function, which adopts the active push mode, supports structured data and has higher execution efficiency and real-time collection accuracy. To meet the needs of refined, visualized, intelligent monitoring of operation and maintenance, telemetry provides the basis of big data analysis for the rapid locating of network problems and network quality optimization and adjustment.

In the deployment scenario of Access Network equipment which supports telemetry technology, the telemetry architecture can be partitioned into the telemetry system and the OLT. The telemetry system is responsible for the subscription configuration, receiving telemetry collection data reported from the OLT, and data processing, storage and analysis. The OLT is responsible for reporting telemetry collection data according to the subscription configuration.

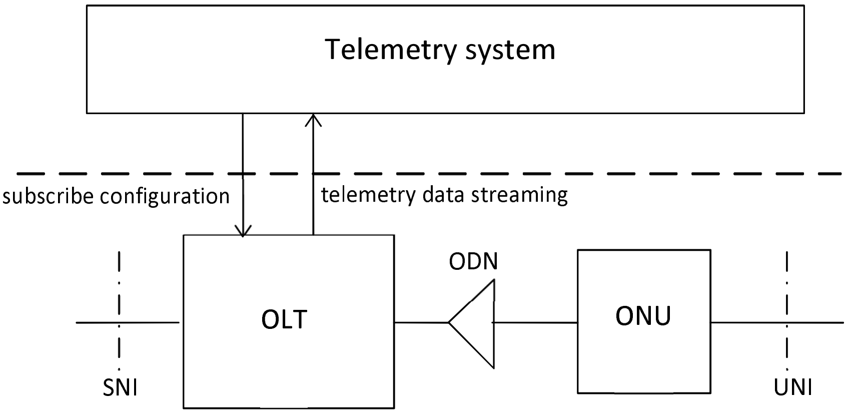


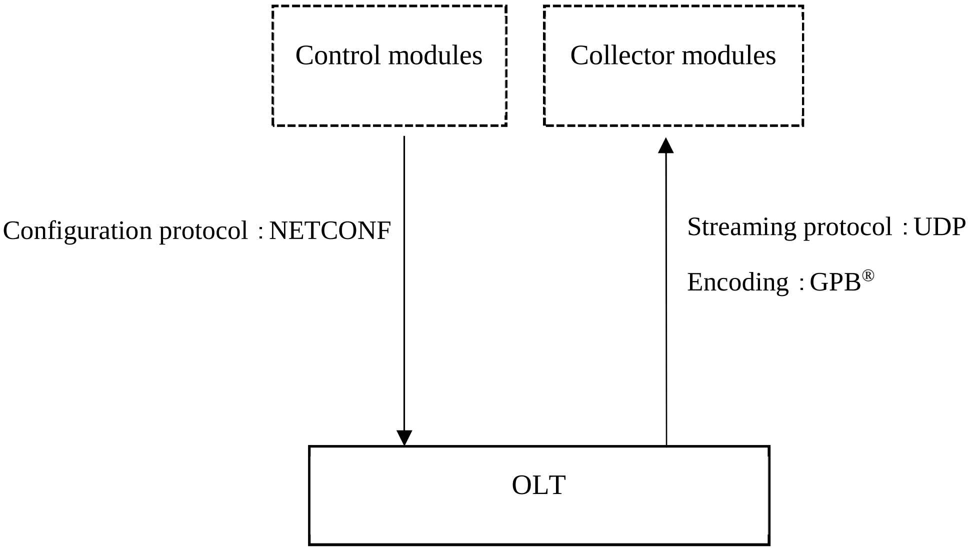
Figure 2: Telemetry architecture in the Access Network

5 Technical Solutions

5.1 UDP Streaming Telemetry Mode

The telemetry system shall support both control and collection features. The control modules should support the NETCONF protocol to send subscription configuration. The corresponding parameters are described in Clause 6 of the present document. If UDP streaming telemetry mode is chosen, the OLT equipment should support UDP encapsulated data reporting. The serialization of the data is based on GPB®.

If UDP streaming telemetry mode is chosen for the telemetry collection, the OLT shall continuously stream the data to the several collectors, once the subscriptions are created as part of the configuration of the OLT and it shall remain the OLT configuration until the subscription is removed. The schematic diagram of UDP streaming telemetry mode is shown in Figure 3.



NOTE: Encoding methods other than GPB® are possible.

Figure 3: UDP streaming telemetry mode

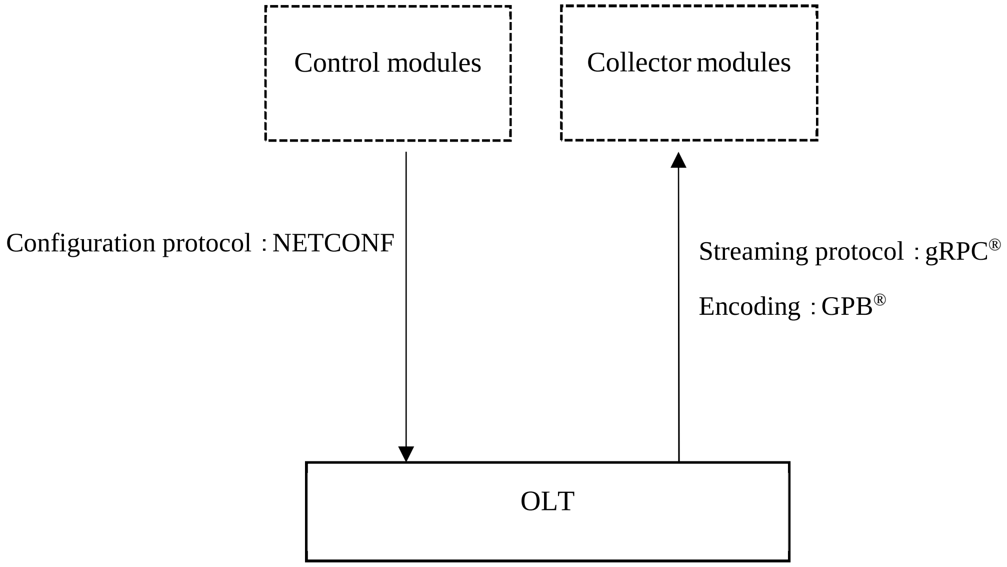
The specific protocol stack layer is shown in Table 1.

Table 1: The telemetry stack layer and requirements of UDP telemetry mode

Telemetry Stack		Requirements
Data layer	Collection data layer	Carries encoded telemetry collection data.
	Telemetry layer	Defines the data header when telemetry data is sent, including sampling path, sampling timestamp, etc. The specific parameters are defined in clause 6.3 of the present document.
Message header layer		Optional support for fragmentation and encoding format indication through the message header layer.
UDP transport layer		UDP provides simple information transmission service, but information might be lost.

5.2 gRPC® Static Telemetry Mode

The telemetry system shall support both control and collection features. The control modules should support the NETCONF protocol to send subscription configuration. The corresponding parameters are described in clause 6. If gRPC® static telemetry mode is chosen, the OLT equipment should support data encapsulation and reporting as a gRPC® client. The schematic diagram of gRPC® static telemetry mode is shown in Figure 4.



NOTE: Encoding methods other than GPB® are possible.

Figure 4: gRPC® Static Telemetry Mode

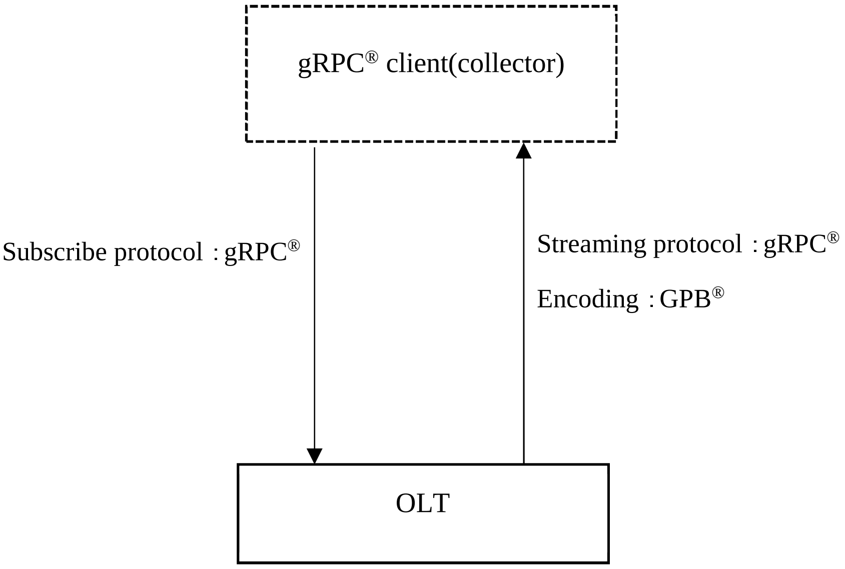
If gRPC® static telemetry mode is chosen for the telemetry collection, the OLT shall continually stream the telemetry data to the several collectors once the subscriptions are created as part of the configuration and it shall remain the OLT configuration until the configuration is removed. The specific protocol stack layer is shown in Table 2.

Table 2: The telemetry stack layer and requirements of gRPC® Static Telemetry Mode

Telemetry Stack		Requirements
Data Layer	Collection data layer	Carries encoded telemetry collection data.
	Telemetry layer	Defines the data header when telemetry data is sent, including sampling path, sampling timestamp, etc. The specific parameters are defined in clause 6.3 of the present document.
	RPC layer	Defines the RPC interfaces when the OLT equipment is reporting telemetry data as a client.
gRPC® layer		Defines the gRPC® protocol interaction format of remote procedure calls.
HTTP 2.0 layer		gRPC® is carried on the HTTP 2.0 protocol.
TLS transport layer		Optional. OLT and telemetry system can perform channel encryption and mutual authentication based on the TLS protocol to realize secure transmission.
TCP transport layer		TCP provides a connection-oriented, reliable information transmission service.
NOTE: The UDP Streaming mode is similar to the gRPC® static mode.		

5.3 gRPC® Dynamic Telemetry Mode

The telemetry system shall support both subscription and collection features. The telemetry system should support creating subscriptions to the OLT as a gRPC® client and receiving streaming data. If gRPC® dynamic telemetry mode is chosen, the OLT equipment should support data encapsulation and reporting as a gRPC® server which supports gRPC® Network Management Interface (gNMI). The schematic diagram of gRPC® static telemetry mode is shown in Figure 5.



NOTE: Encoding methods other than GPB® are possible.

Figure 5: gRPC® Dynamic Telemetry Mode

If gRPC® dynamic telemetry mode is chosen for the telemetry collection, the OLT shall continually stream the telemetry data to the one certain collector when this collector sends the subscriptions to the OLT. This dynamic subscription shall terminate when the collector cancels the subscription or when the session terminates. The dynamic telemetry mode is suitable when the collector exactly knows its telemetry requirements. This mode is convenient as a centralized way of configuring the network and requesting operational data. The specific protocol stack layer is shown in Table 3.

Table 3: The telemetry stack layer and requirements of gRPC® Dynamic Telemetry Mode

Telemetry Stack		Requirements
Data Layer	Collection data layer	Carries encoded telemetry collection data.
	Telemetry layer	Defines the data header when telemetry data is sent, including sampling path, sampling timestamp, etc. The specific parameters are defined in clause 6.3 of the present document.
	RPC layer	Defines the RPC interfaces when the OLT equipment is reporting telemetry data as a server.
gRPC® layer		Defines the gRPC® protocol interaction format of remote procedure call.
HTTP 2.0 layer		gRPC® is carried on the HTTP 2.0 protocol.
TLS transport layer		Optional. OLT and telemetry system can perform channel encryption and mutual authentication based on the TLS protocol to realize secure transmission.
TCP transport layer		TCP provides a connection-oriented, reliable information transmission service.

6 Interface Requirements

6.1 Overview

The gRPC® layer, the telemetry layer and the collection data layer play different roles in the telemetry system. The gRPC® layer shall only exist when the streaming protocol is gRPC®. The telemetry layer and the collection data layer shall always exist in telemetry messages and carries the main contents.

Clause 6 of the present document specifies the technical requirements and the key parameters of the gRPC® layer, the telemetry layer and collection data layer.

6.2 gRPC® Layer Requirements

6.2.1 gRPC® Static Telemetry mode

When the streaming protocol is gRPC® and it is gRPC® Static Telemetry mode, the OLT shall stream collection data through an RPC interface to the telemetry system as a gRPC® client according to the telemetry configuration. The structure of this RPC interface has been defined in this layer.

The RPC structure shall contain the following elements:

- Request ID.
- Streaming telemetry data structure and its elements are defined by the Telemetry layer. The telemetry layer requirements are defined in clause 6.3.

6.2.2 gRPC® Dynamic Telemetry mode

When the streaming protocol is gRPC® and it is gRPC® Dynamic Telemetry mode, the telemetry system shall send a subscription request through an RPC interface to the OLT. The structure of this subscribe RPC interface has been defined in this layer.

The subscribe RPC interface structure shall contain the following elements:

- Request ID.
- Encoding method.
- Data model path of the sensor which describes the specific ANT objects for collection.
- Sample interval.

When the OLT receives the subscription request, it shall stream the collection data through a corresponding RPC reply to the telemetry system as a gRPC® server.

The reply of the subscription request via the RPC interface shall contain the following elements:

- Subscription ID.
- Request ID.
- Streaming telemetry data structure and its elements are defined by the Telemetry layer. The telemetry layer requirements are defined in clause 6.3.

When the gRPC® dynamic telemetry session needs to be terminated, a cancel subscription request RPC shall be sent by the telemetry system.

The cancel subscription request via the RPC interface shall contain the following elements:

- Request ID.
- Subscription ID.

When the OLT receives the cancel subscription request, it shall reply the result of the cancel request.

The reply of the cancel subscription request via the RPC interface shall contain the following elements:

- Request ID.
- Response code.
- Error description.

6.3 Telemetry Layer Requirements

When the streaming protocol is gRPC®, the telemetry layer is carried in the gRPC® layer. When the streaming protocol is UDP, the telemetry layer can exist independently in the data layer of the telemetry message. This layer defines the data header of the telemetry collection data. It shall contain the following elements:

- OLT node ID.
- Subscription ID.
- Data model path of sensor which describes the specific ANT objects for collection.
- Collection ID.
- Collection start time.
- Collection end time.
- Message timestamp.
- Encoding method.
- Current time interval of data sampling.
- Data sampling timestamp.
- The specific collection data structure and its elements are defined by the collection data layer. The collection data layer requirements are defined in clause 6.4 of the present document.

6.4 Collection Data Layer Requirements

The collection data structure is carried in the telemetry structure. The collection data structure defines the telemetry collection items of the Access Network. For these specific collection items refer to clause 8 of the present document.

For the actual collected data, the value '0' is meaningful data which shall be reported. If some collection items cannot be sampled by the OLT due to the inability of the OLT, the specific items should be indicated in the collection data layer. According to the equipment capabilities and user needs, an "empty" collection item can be indicated in either one of the following ways:

- If one collection item cannot be sampled by the OLT, it will not be generated in the telemetry message.
- The telemetry message transport mechanism can support the capability to optimize the message when the collection item is empty and it is able to signal the reason why a particular collection item is not available. In the case that the encoding format of the collection data layer is GPB®, the GPB® encoding may need extensions for a more efficient telemetry streaming. These extensions compress the telemetry message and are more explicit explaining why a telemetry collection item cannot be collected. Refer to Annex C for more details on a feasible implementation.

7 Telemetry Functional Requirements

7.1 Overview

For higher precision network monitoring and the automation of network optimization, the telemetry system and the OLT shall implement some basic telemetry functions. Clause 7 of the present document specifies the functions required for the basic telemetry scenarios in the Access Network.

7.2 Telemetry System

The telemetry system is an automated controller for Access Network telemetry. It shall implement telemetry collection and may have the capability to dynamically configure and generate the telemetry subscriptions. In addition, the telemetry system can provide a telemetry service interface for other applications using the telemetry data.

The telemetry system consists of six functional components as illustrated in Figure 6 and are described as follows.

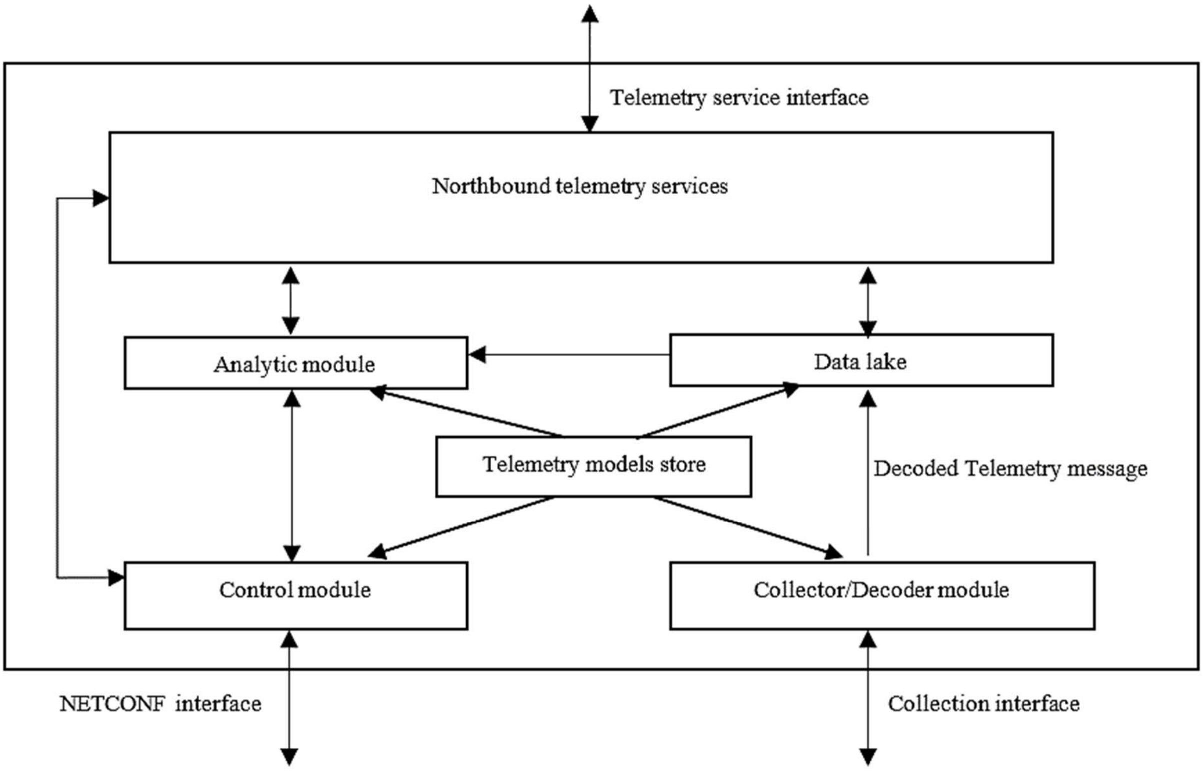


Figure 6: Telemetry System

- **Telemetry models store:** storage for telemetry models of different OLT versions and the storage for defined metrics for the analytic module such as window size, time lag, sample interval, start time, etc. The collection interface of the collector/decoder module can be generated based on the models stored in the telemetry model store. The telemetry models stored in the telemetry model store are used to generate NETCONF interfaces on the control module. In addition, the telemetry model store can provide configurations of telemetry data structures for the data lake and define metrics of analytic module for Telemetry data.

- **Control module:** interfaces to the OLT via a NETCONF client in order to subscribe to different data streams. It also sets the periodicity and the granularity of streams and other parameters defining the data collection settings.

- **Collector/Decoder module:** interfaces to the OLT. It collects the subscribed telemetry data streams, decodes them based on the collection interface, and populates the data lake.

- **Data lake:** stores the history of the collected streams. It can provide inputs to the northbound telemetry services and the analytic module.

- **Analytic module:** implements the different policies and network rules in order to forecast the network and it automates some actions for network optimization based on the collected data. It can use some Machine Learning/Artificial intelligence techniques and generate corresponding ML/AI models to propose different advanced network services, such as real-time forecasting for future networking. The collected data used by the analytic module should be verified to be completed, valid and non-redundant for training and analysis accuracy. It should be able to subscribe to a specific stream or change the settings of the collection parameters by communicate directly to the control module. Its different services or properties are exposed to the northbound telemetry services.

- **Northbound telemetry services:** exposes the different services delivered by the telemetry system to external systems.

7.3 OLT

7.3.1 OLT Internal Functions

The OLT internal functions are the essential blocks to enabling Access Network telemetry. The internal functional blocks are:

- Sensor Path: The sensor path describes the specific ANT objects for collection.

- Exclude Filters: Filter to exclude certain values from the collection values.

- Sensor Group: The sensor group represents a reusable grouping of multiple sensor paths and exclusion filters.

- Subscriptions: The subscriptions contain both static subscription and dynamic subscription. A static telemetry subscription is configured locally on the OLT through the telemetry system, and is permanently saved on the OLT even if the OLT restarts. A dynamic subscription is typically configured through an gRPC® channel, and does not persist if the OLT restarts or if the gRPC® channel is reset or torn down. A telemetry subscription consists of a set of collector addresses, sensor groups and exclusion filters.

- Encoding: Data which is streamed by the OLT may be encoded into GPB® encoding format.

- Transport protocol: The OLT can stream telemetry data through one of the following supported transport protocols:

  - gRPC®.

  - UDP.

- UDP Streaming Initiation: The telemetry system cannot start the UDP streaming initiation. The OLT can continuously stream the telemetry data through UDP.

- gRPC® Session Initiation: There are two options for who initiates the streaming data as described in clause 5:
    - Option one is that the telemetry system initiates a session with the OLT and subscribes to collection data to be streamed by a gRPC® channel.
    - Option two is that the OLT initiates the sessions by several gRPC® channels to more than one collector based on the subscription.
- Collection Capabilities Exchange: Items such as different model formats, and optional feature can impact the collector decoding of the telemetry data. The OLT should support sending explicit telemetry collection items which are modelled by the model file to the collector.

7.3.2 Collection Capabilities Exchange Process

The collection capabilities exchange process between the OLT and the telemetry system (collector) is shown in Figure 7.

The telemetry collection data shall be defined by the telemetry data models. Before the telemetry subscription is sent to the OLT, the telemetry system sends a model request to the OLT and the OLT should report the telemetry data models to the collector in the telemetry system. It is assumed that the connectivity between the OLT and the telemetry system including L3 is stable. The collector compares the local telemetry data models with models reported by the OLT. In case of a mismatch, various options are possible including the telemetry system updates the local telemetry data models to match the OLT models.

After the telemetry subscription is sent to the OLT, the OLT combines the data into binary messages based on the corresponding telemetry data models and streams the telemetry data to the collector. The collector decodes and outputs the subscribed streaming telemetry data based on the corresponding telemetry data models in the local telemetry models store.

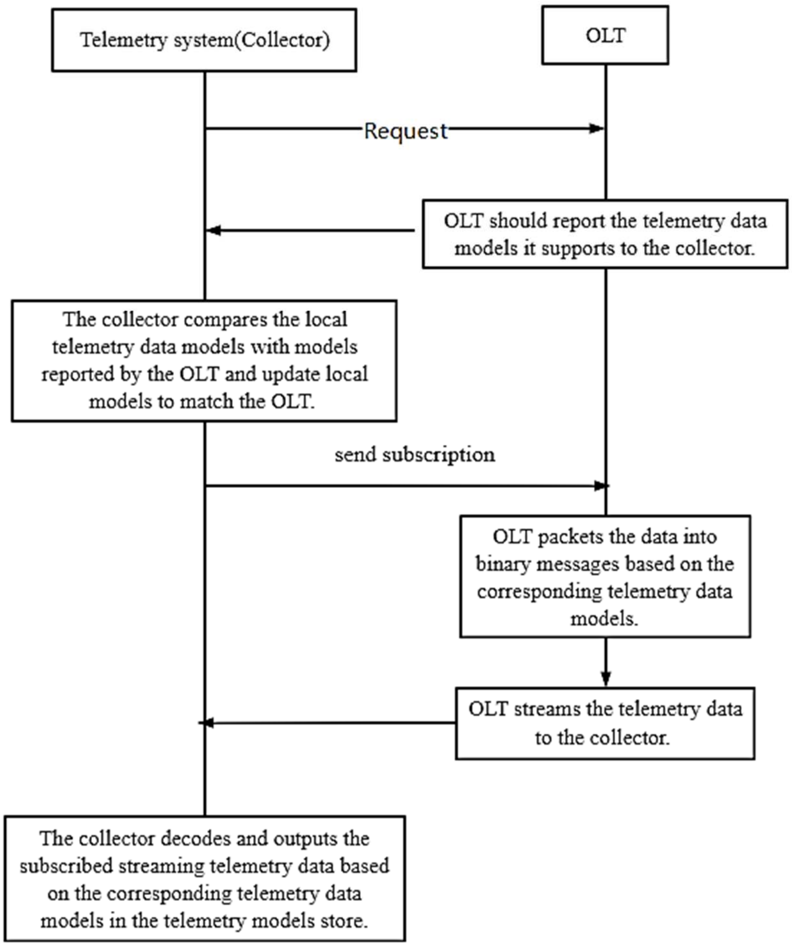

Telemetry system(Collector)
OLT
Request
OLT should report the telemetry data models it supports to the collector.
The collector compares the local telemetry data models with models reported by the OLT and update local models to match the OLT.
send subscription
OLT packets the data into binary messages based on the corresponding telemetry data models.
OLT streams the telemetry data to the collector.
The collector decodes and outputs the subscribed streaming telemetry data based on the corresponding telemetry data models in the telemetry models store.

Figure 7: Communication Process between OLT and Telemetry System

7.3.3 OLT Performance Requirements

Table 4 shows the basic performance requirements of the OLT in telemetry technologies.

Table 4: The performance of OLT in telemetry technologies

Functions	Performance Requirements
Maximum supported number of subscriptions	10
Maximum supported number of sensor path contained by the sensor group	10
Maximum supported number of sensor groups associated with each subscription	5
Maximum supported number of session initiation	25
The number of collection items collected by one board of OLT per second	at least 2 000
The number of telemetry packets streamed by the OLT (one board) per second	that least 500

8 Collection Parameters

8.1 Overviews and Definitions

Clause 8 of the present document specifies the collection parameters in the Access Network, which are divided into three types:

- Access Network Traffic information collection.
- Optical link information collection.
- ONU information collection.

The OLT PON port in clause 8 may be EPON, 10G-EPON, GPON, XG-PON, XGS-PON, GPON/XG-PON dual mode, or GPON/XGS-PON dual mode. The ONU information collection is identified by the ONU type which is either EPON or GPON.

EPON includes EPON, 10G-EPON.

GPON includes GPON, XG-PON, XGS-PON.

The content "error packets" in Table 4 and Table 7 means all error packets which include undersized packets, oversized packets, CRC error packets, and alignment error packets, jabber packets and fragment packets.

8.2 Access Network Traffic Information Collection

8.2.1 Overviews and Definitions

Clause 8.2 of the present document specifies the traffic information of the OLT PON port, the OLT Ethernet uplink port and the ONU PON port in the Access Network, which contains PON traffic, EthernetCsmacd OLT uplink traffic, queue traffic, service flow traffic and ONU PON traffic. Apart from the above, the ONU Ethernet traffic and Wi-Fi® traffic are for further study.

In the upstream direction, there is an insignificant percentage of broadcast traffic in the Access Network. In the downstream direction all traffic is broadcast due to the P2MP characteristic of PON. The broadcast traffic will be collected in the parameters of the transmitting direction of the OLT PON port, such as port-tx-bytes and port-tx-packets. The traffic in the PON Access Network is categorized into unicast traffic and multicast traffic. The unicast flows and multicast flows represent different services used by clients. For example, IPTV is delivered as multicast flows and generic network traffic such as web traffic is delivered as unicast flows. Both flow types may be identified and collected in the telemetry system. The following definitions apply to the OLT PON traffic and the OLT EthernetCsmacd traffic as used in clause 8.2.2.

1) port-tx-bytes: The number of Ethernet frame bytes sent by the port (OLT PON port/Ethernet uplink port).

2) port-rx-bytes: The number of Ethernet frame bytes received by the port (OLT PON port/Ethernet uplink port).

3) port-tx-packets: The number of Ethernet packets sent by the port (OLT PON port/Ethernet uplink port).

4) port-rx-packets: The number of Ethernet packets received by the port (OLT PON port/Ethernet uplink port).

5) port-tx-discard-packets: The number of Ethernet packets discarded in the transmitting direction of the port (OLT PON port/Ethernet uplink port).

6) port-rx-discard-packets: The number of Ethernet packets discarded in the receiving direction of the port (OLT PON port/Ethernet uplink port).

7) port-rx-crc-error-packets: The number of packets with Ethernet CRC errors in the receiving direction of the port (OLT PON port/Ethernet uplink port).

8) port-tx-crc-error-packets: The number of packets with Ethernet CRC errors in the sending direction of the port (OLT PON port/Ethernet uplink port).

9) port-rx-oversized-discard-packets: The number of oversized Ethernet packets discarded in the receiving direction of the port (OLT PON port/Ethernet uplink port).

10) port-tx-oversized-discard-packets: The number of oversized Ethernet packets discarded in the sending direction of the port (OLT PON port/Ethernet uplink port).

11) port-rx-undersized-discard-packets: The number of undersized Ethernet packets discarded in the receiving direction of the port (OLT PON port/Ethernet uplink port).

12) port-tx-undersized-discard-packets: The number of undersized Ethernet packets discarded in the sending direction of the port (OLT PON port/Ethernet uplink port).

13) port-rx-error-packets: The number of Ethernet error packets received by the port (OLT PON port/Ethernet uplink port).

14) port-tx-error-packets: The number of Ethernet error packets detected by the port (OLT PON port/Ethernet uplink port).

15) port-tx-rate: Average transmit rate of the port (OLT PON port/Ethernet uplink port).

16) port-rx-rate: Average receive rate of the port (OLT PON port/Ethernet uplink port).

17) port-tx-peak-rate: Peak transmit rate of the port (OLT PON port/Ethernet uplink port) in seconds

18) port-rx-peak-rate: Peak receive rate of the port (OLT PON port/Ethernet uplink port) in seconds.

19) port-rx-alignment-error-packets: The number of misaligned Ethernet packets received by the port (Ethernet uplink port).

20) port-tx-fragment-packets: The number of fragment packets sent by the port (Ethernet uplink port).

21) port-rx-fragment-packets: The number of fragment packets received by the port (Ethernet uplink port).

22) port-tx-jabber-packets: The number of jabber packets sent by the port (Ethernet uplink port).

23) port-rx-jabber-packets: The number of jabber packets received by the port (Ethernet uplink port).

24) port-tx-unicast-bytes: The number of unicast frame (A frame in which the least significant bit (that is, the eighth bit) of the first byte of the destination MAC address in an Ethernet frame is 0.) bytes sent by the port (OLT PON port/Ethernet uplink port).

25) port-rx-unicast-bytes: The number of unicast frame bytes received by the port (OLT PON port/Ethernet uplink port).

26) port-tx-multicast-bytes: The number of multicast frame (A non-broadcast frame in which the least significant bit (that is, the eighth bit) of the first byte of the destination MAC address in an Ethernet frame is 1.) bytes sent by the port (OLT PON port/Ethernet uplink port)).

27) port-tx-unicast-rate: Average transmit unicast traffic rate of the port (OLT PON port/Ethernet uplink port).

28) port-rx-unicast-rate: Average receive unicast traffic rate of the port (OLT PON port/Ethernet uplink port).

29) port-tx-multicast-rate: Average transmit multicast traffic rate of the port (OLT PON port/Ethernet uplink port).

30) port-tx-peak-unicast-rate: Peak transmit unicast traffic rate of the port (OLT PON port/Ethernet uplink port) in seconds

31) port-rx-peak-unicast-rate: Peak receive unicast traffic rate of the port (OLT PON port/Ethernet uplink port) in seconds.

32) port-tx-peak-multicast-rate: Peak transmit multicast traffic rate of the port (OLT PON port/Ethernet uplink port) in seconds.

The following definitions apply to the queue traffic of the OLT PON port and the OLT Ethernet uplink port in clause 8.2.2. The queue specifically refers to the transmit direction:

1) pass-bytes: The number of bytes forwarded by port queue.

2) pass-packets: The number of packets forwarded by port queue.

3) drop-packets: The number of packets discarded (coloured red traffic by the policing function) by port queue.

4) pass-green-bytes: The number of bytes coloured green traffic (by the policing function) forwarded by port queue.

5) pass-green-packets: The number of packets coloured green traffic (by the policing function**)** forwarded by port queue.

6) drop-green-packets: The number of packets of coloured green traffic (by the policing function) discarded by port queue.

7) pass-yellow-bytes: The number of bytes coloured yellow traffic (by the policing function) forwarded by port queue.

8) pass-yellow-packets: The number of packets coloured yellow traffic (by the policing function) forwarded by port queue.

9) drop-yellow-packets: The number of packets coloured yellow traffic (by the policing function) discarded by port queue.

The following definitions apply to the traffic of the OLT Service flows in clause 8.2.2. Packets of service flows are sent out through port queues. Packets that are discarded in port queues are counted and can be associated with a service flow:

1) downstream-queue-drop-count: The total number of packets dropped by the queue in the downstream direction.

2) downstream-queue-pass-count: The total number of packets passed through the queue in the downstream direction.

3) downstream-queue-drop-max: The maximum number of packets dropped per second by the queue in the downstream direction.

4) downstream-queue-drop-min: The minimum number of packets dropped per second by the queue in the downstream direction.

5) downstream-queue-drop-rate-max: The maximum packet loss rate in seconds in the downstream queue.

6) downstream-queue-drop-rate-min: The minimum packet loss rate in seconds in the downstream queue.

7) downstream-queue-drop-seconds-count: The minimum packet loss rate of the queue in seconds in the downstream direction.

8) downstream-average-rate: The average transmit rate of the service flow in the downstream direction.

9) upstream-pass-bytes: Bytes passed by the service flow in the upstream direction.

10) upstream-pass-count: The total number of packets passed by the service flow in the upstream direction.

11) upstream-drop-count: The total number of packets dropped by the service flow in the upstream direction.

The following definitions apply to the ONU PON traffic in clause 8.2.2:

1) tx-rate: Average transmit rate of the ONU.

2) rx-rate: Average receive rate of the ONU.

3) tx-peak-rate: Peak transmit rate of the port in seconds.

4) rx-peak-rate: Peak receive rate of the port in seconds.

8.2.2 Table of Access Network Traffic Information Collection

Table 5: Tabulates the Access Network traffic information collection items and corresponding time interval requirements

Type	Items	Contents	Equipment sampling capability		Sample interval	
			Single ANT object	Set of ANT objects	Single ANT object	Set of ANT objects
Traffic	PON traffic	1. port-tx-bytes 2. port-rx-bytes 3. port-tx-packets 4. port-rx-packets 5. port-tx-discard-packets 6. port-rx-discard-packets 7. port-rx-crc-error-packets 8. port-rx-oversized-discard-packets 9. port-rx-undersized-discard-packets 10. port-rx-error-packets 11. port-tx-error-packets 12. port-tx-rate 13. port-rx-rate 14. port-tx-peak-rate 15. port-rx-peak-rate 16. port-tx-unicast-bytes 17. port-rx-unicast-bytes 18. port-tx-multicast-bytes 19. port-tx-unicast-rate 20. port-rx-unicast-rate 21. port-tx-multicast-rate 22. port-tx-peak-unicast-rate 23. port-rx-peak-unicast-rate 24. port-tx-peak-multicast-rate	< 1 s	< 1 s	< 1 s	< 1 s
	EthernetCsmacd-traffic	1. port-tx-bytes 2. port-rx-bytes 3. port-tx-packets 4. port-rx-packets 5. port-tx-discard-packets 6. port-rx-discard-packets 7. port-rx-alignment-error-packets 8. port-tx-crc-error-packets 9. port-rx-crc-error-packets 10. port-tx-oversized-packets 11. port-rx-oversized-packets 12. port-tx-undersized-packets 13. port-rx-undersized-packets 14. port-tx-fragment-packets 15. port-rx-fragment-packets 16. port-tx-jabber-packets 17. port-rx-jabber-packets 18. port-tx-error-packets 19. port-rx-error-packets 20. port-tx-rate 21. port-rx-rate 22. port-tx-peak-rate 23. port-rx-peak-rate 24. port-tx-unicast-bytes 25. port-rx-unicast-bytes 26. port-tx-multicast-bytes 27. port-tx-unicast-rate 28. port-rx-unicast-rate 29. port-tx-multicast-rate 30. port-tx-peak-unicast-rate 31. port-rx-peak-unicast-rate 32. port-tx-peak-multicast-rate	< 1 s	< 1 s	< 1 s	< 1 s

Type	Items	Contents	Equipment sampling capability		Sample interval	
			Single ANT object	Set of ANT objects	Single ANT object	Set of ANT objects
	Queue-traffic	1. pass-bytes 2. pass-packets 3. drop-packets 4. pass-green-bytes 5. pass-green-packets 6. drop-green-packets 7. pass-yellow-bytes 8. pass-yellow-packets 9. drop-yellow-packets	< 1 s	< 1 s	< 1 s	< 1 s
	Service-flow-traffic	1. downstream-drop-count 2. downstream-pass-count 3. downstream-drop-max 4. downstream-drop-min 5. downstream-drop-rate-max 6. downstream-drop-rate-min 7. downstream-drop-seconds-count 8. downstream-pass-bytes 9. downstream-average-rate 10. upstream-pass-bytes 11. upstream-pass-count 12. upstream-drop-count	< 3 s	< 3 s	<3 s	< 3 s
	ONU-traffic	1. tx-rate 2. rx-rate 3. tx-peak-rate 4. rx-peak-rate	< 1 s	< 1 s	< 1 s	< 1 s

8.3 Optical Link Information Collection

8.3.1 Overviews and Definitions

Clause 8.3 of the present document specifies the optical link information of the PON port in the Access Network. When the PON port is dual mode, it has two PON channels (channel 1 and channel 2), otherwise, it only have one PON channel (channel 1).

The following definitions apply to the performance of the OLT PON port transceivers and be used in clause 8.3.2:

1) temperature: Optical module temperature.

2) supply-voltage: Optical module power supply voltage.

3) channel-1-type: Type of channel 1.

4) channel-1-tx-bias: Current transmit bias of channel 1.

5) channel-1-tx-power: Transmit power of channel 1.

6) channel-1-idle-rssi: Optical power at idle time of channel 1.

7) channel-2-type: Type of channel 2.

8) channel-2-tx-bias: Current transmit bias of channel 2.

9) channel-2-tx-power: Transmit power of channel 2.

10) channel-2-idle-rssi: Optical power at idle time of channel 2.

11) module-type: Optical module type.

12) module-sub-type: Optical module class.

8.3.2 Table of Optical Link Information Collection

Table 6: Tabulates the Optical link information collection items and corresponding time interval requirements

Type	Items	Contents	Equipment sampling capability		Sample interval	
			Single ANT object	Set of ANT objects	Single ANT object	Set of ANT objects
Optical link	OLT transceivers	1. temperature 2. supply-voltage 3. channel-1-type 4. channel-1-tx-bias 5. channel-1-tx-power 6. channel-1-idle-rssi 7. channel-2-type 8. channel-2-tx-bias 9. channel-2-tx-power 10. channel-2-idle-rssi 11. module-type 12. module-sub-type	< 1 min	< 1 min	< 1 min	< 1 min

8.4 ONU Information Collection

8.4.1 Overviews and Definitions

The ONU local information is sampled by the OLT directly.

The ONU remote information is gathered by the ONU device. The ONU streams this information to the OLT through the channel between the OLT and the ONU.

The following definitions apply to GPON ONU information as used in clause 8.4.2 and EPON ONU information in clause 8.4.3:

1) OLT-rx-power: The optical power received by the OLT transceiver.

2) online-duration: The ONU online time.

3) last-down-time: The last offline time of ONU.

4) last-down-cause: The reason for last offline of ONU.

5) ONU-status: The ONU status.

The following definitions apply to GPON ONU uplink information as used in clause 8.4.2:

1) LOFi-alarm-count: The number of ONU LOFi (Loss of Frame indication) alarms on the OLT for a given ONU. It is valid for GPON ONU, but is not supported by XG(S)-PON ONU.

2) DOWi-alarm-count: The number of ONU DOWi (Drift of Window indication) alarms after the ONU is initiated by the OLT. GPON/XG-PON ONU valid.

3) upstream-delimiter-error-count: The number of ONU uplink frame delimitation errors after the ONU goes online. It is valid for GPON ONU, but is not supported by XG(s)-PON ONUs.

4) upstream-BIP-error-count: The number of ONU Bit-interleaved parity errors in the upstream frame, once the ONU is enabled. It is valid for both GPON/XG(S)-PON ONU.

5) downstream-BIP-error-count: The number of ONU Bit-interleaved parity errors in the downstream frame once the ONU is enabled. It is valid for GPON ONU, but is not supported by XG(S)-PON ONU.

6) upstream-FEC-block: The number of upstream corrected FEC (forward error correction) blocks.

7) upstream-FEC-error-block: The number of upstream uncorrected FEC (forward error correction) block errors.

8) upstream-FEC-total-block: The sum of upstream corrected blocks and uncorrected FEC block errors.

9) upstream-FEC-byte: Upstream FEC corrected bytes. It is valid for both GPON/XG(S)-PON ONU.

10) upstream-HEC-error-count: The number of upstream HEC (Header Error Code) errors. It is valid for both GPON/XG(S)-PON ONU.

11) upstream-gem-count: The number of upstream GEM frames.

12) LOSi-alarm-count: The number of ONU LOSi (loss of signal indication) alarms.

13) DGi-alarm-count: The number of ONU DGi (Dying Gasp indication) alarms.

The following contents apply to ONU transceivers in clause 8.4.2 and clause 8.4.3. Parameters 1 through 5 are defined in Recommendation ITU-T G.988 [i.2], clause 9.14.6 and Table 11.2.10-1:

1) optical-unit-rx-power [i.2]: Received optical power.

2) optical-unit-tx-power [i.2]: Transmit optical power.

3) optical-unit-laser-bias-current [i.2]: The current bias of the port.

4) optical-unit-temperature [i.2]: The current temperature of the optical module.

5) optical-unit-voltage [i.2]: The current voltage of the optical module.

6) module-type: Optical module type.

7) module-sub-type: Optical module subtype.

The following contents apply to ONU status and downlink quality in clause 8.4.2 and clause 8.4.3. Parameters 1 through 5 are defined in Recommendation ITU-T G.988 [i.2], clause 9.2.22. Parameters 6 and 7 are defined in Recommendation ITU-T G.988 [i.2], clause 9.2.15. Parameters 8 through 10 are defined in Recommendation ITU-T G.988 [i.2], clause 9.1.18:

1) downstream-FEC-corrected-bytes [i.2]: The number of downstream FEC corrected bytes.

2) downstream-FEC-corrected-words [i.2]: The number of downstream FEC corrected words.

3) downstream-FEC-uncorrected-words [i.2]: The number of downstream FEC uncorrected words.

4) downstream-total-rx-code-words [i.2]: The total number of downstream received code words.

5) downstream-FEC-seconds [i.2]: Downstream FEC correction time.

6) XG(S)-PON-GEM-HEC-error-count [i.2]: The number of GEM HEC errors received by ONU.

7) XG(S)-PON-GEM-key-error-count [i.2]: The number of discarded XG-PON GEM frames.

8) memory-occupation [i.2]: Memory utilization of ONU.

9) CPU-occupation [i.2]: CPU utilization of ONU.

10) CPU-temperature [i.2]: CPU temperature of ONU.

11) ONU-PON-send-packets: The number of packets sent by the ONU PON port.

12) ONU-PON-receive-packets: The number of packets received by the ONU PON port.

13) ONU-PON-receive-errors-packets: The number of error packets received by the ONU PON port.

14) TCONT-queue-dropped-packets: The number of packets discarded in all T-CONT queues of the ONU.

15) TCONT-queue-passing-packets: The number of packets forwarded in all T-CONT queues of the ONU.

8.4.2 Table of GPON ONU Collection

Table 7: Tabulates the GPON ONU collection items and corresponding time interval requirements

Type	Items	Contents	Equipment sampling capability		Sample interval	
			Single ANT object	Set of ANT objects	Single ANT object	Set of ANT objects
ONU local information	GPON ONU information	1. OLT-rx-power 2. online-duration 3. last-down-time 4. last-down-cause 5. ONU-status	< 5 min	< 5 min	< 5 min	< 5 min
	GPON ONU uplink information	1. LOFi-alarm-count 2. DOWi-alarm-count 3. upstream-delimiter-error-count 4. upstream-BIP-error-count 5. downstream-BIP-error-count 6. upstream-FEC-block 7. upstream-FEC-error-block 8. upstream-FEC-total-block 9. upstream-FEC-byte 10. upstream-HEC-error-count 11. upstream-gem-count 12. LOSi-alarm-count 13. DGi-alarm-count	< 5 min	< 5 min	< 5 min	< 5 min
ONU remote information	ONU transceivers	1. optical-unit-rx-power 2. optical-unit-tx-power 3. optical-unit-laser-bias-current 4. optical-unit-temperature 5. optical-unit-voltage 6. module-type 7. module-sub-type	< 5 min	< 15 min	< 5 min	< 15 min
	ONU status and downlink quality	1. downstream-FEC-corrected-bytes 2. downstream-FEC-corrected-words 3. downstream-FEC-uncorrected-words 4. downstream-total-rx-code-words 5. downstream -FEC-seconds 6. XG-PON-GEM-HEC-error-count 7. XG-PON-GEM-key-error-count 8. memory-occupation 9. CPU-occupation 10. CPU-temperature 11. ONU-PON-send-packets 12. ONU-PON-receive-packets 13. ONU-PON-receive-errors-packets 14. TCONT-queue-dropped-packets 15. TCONT-queue-passing-packets	< 5 min	< 15 min	< 5 min	< 15 min

8.4.3 Table of EPON ONU Collection

Table 8: Tabulates the EPON ONU collection items and corresponding time interval requirements

Type	Items	Contents	Equipment sampling capability		Sample interval	
			Single ANT object	Set of ANT objects	Single ANT object	Set of ANT objects
ONU local information	EPON ONU information	1. OLT-rx-power 2. online-duration 3. last-down-time 4. last-down-cause 5. ONU-status	< 5 min	< 5 min	< 5 min	< 5 min
ONU remote information	ONU transceivers	1. optical-unit-rx-power 2. optical-unit-tx-power 3. optical-unit-laser-bias-current 4. optical-unit-temperature 5. optical-unit-voltage 6. module-type 7. module-sub-type	< 5 min	< 15 min	< 5 min	< 15 min
	ONU status and downlink quality	1. memory-occupation 2. CPU-occupation 3. CPU-temperature 4. ONU-PON-send-packets 5. ONU-PON-receive-packets 6. ONU-PON-receive-errors-packets	< 5 min	< 15 min	< 5 min	< 15 min

Annex A (informative): Examples of Telemetry Technical Solutions

A.1 UDP Streaming Telemetry Mode use case

Initially:

Make sure the connectivity between the OLT and the telemetry system including L3 is stable.

Define a subscription by the telemetry system:

Send NETCONF configuration to create a subscription which defines the data planned to be streamed from the OLT to the telemetry system. Here is the procedure:

1) Create one or more destinations to collect telemetry data from the OLT. Define a destination-group to contain the details about the destinations which includes the destination IP addresses, and UDP port number.

2) Specify the sensor path, which describes the specific ANT objects for collection. Create a sensor-group to contain multiple sensor paths.

3) Create a subscription to telemetry data that is streamed from the OLT to the collector. A subscription binds the destination-group, sensor-group, encoding method, time interval and transport protocol, which is UDP.

The telemetry system starts to receive telemetry collection data:

The OLT establishes a UDP session with each destination in the subscription. After the session is established, the OLT streams data to the telemetry system to create a data pool. The session will not be terminated until the subscription configuration is removed.

Operate on telemetry collection data for analysis and network automation:

The received telemetry data is analysed and stored in the telemetry system. The telemetry system trains the machine learning models of the network and forecasts the future network. Meanwhile, the telemetry management will optimize the network based on user needs according to the forecast result. This is a step towards network self-sustainability.

A.2 gRPC® Static Telemetry Mode use case

Initially:

Make sure the connectivity between the OLT and the telemetry system including L3 is stable.

Define a subscription by the telemetry system:

Send NETCONF configuration to create a subscription which defines the data planned to be streamed from the OLT to the telemetry system. Here is the procedure:

1) Create one or more destinations to collect telemetry data from the OLT. Define a destination-group to contain the details about the destinations which includes the destination IP address, and TCP port.

2) Specify the sensor path which describes the specific ANT objects for collection. Create a sensor-group to contain multiple sensor paths.

3) Create a subscription to telemetry data that is streamed from the OLT. A subscription binds the destination-group, sensor-group, encoding method which is appointed to GPB®, time interval and transport protocol which is gRPC®.

The telemetry system starts to receive telemetry collection data:

The OLT establishes a gRPC® session with each destination in the subscription. After the session is established, the OLT streams data to the telemetry system to create a data pool. The session will not be terminated until the subscription configuration is removed.

Operate on telemetry collection data for analysis and network automation:

The received telemetry data is analysed and stored in the telemetry system. The telemetry system trains the machine learning models of the network and forecasts the future network. Meanwhile, the telemetry management will optimize the network based on user needs according to the forecast result. This is a step towards network self-sustainability.

A.3 gRPC® Dynamic Telemetry Mode use case

Initially:

Make sure the connectivity between the OLT and the telemetry system including L3 is stable.

Enable the gRPC® server on the OLT to receive incoming subscription request from the telemetry system and be ready for streaming telemetry collection data.

Define a subscription from telemetry system to the OLT:

The telemetry system establishes the gRPC® session with the OLT, subscribes to data to be streamed, and send the gRPC® subscription request to the OLT which contains following elements: the OLT IP address and its TCP port, encoding method, sensor path and time interval.

The telemetry system starts to receive telemetry collection data:

The telemetry system establishes a dynamic gRPC® session with the OLT. After the request, the OLT streams data to the telemetry system to create a telemetry data entry in the data lake. The streaming session will be terminated when the gRPC® session is terminated.

Operate on telemetry collection data for analysis and network automation:

The received telemetry data is analysed and stored in the telemetry system. The telemetry system trains the machine learning models of the network and forecasts the future network. Meanwhile, the telemetry management will optimize the network based on user needs according to the forecast result. This is a step towards network self-sustainability.

Annex B (informative): Example Implementation of the Telemetry system

B.1 Introduction

This annex presents a telemetry streaming workflow and Machine Learning (ML) pipeline implementation option that follows the Telemetry system specified in clause 7.1. Figure B.1 shows an overview of the proposed architecture, which is comprised of seven functional blocks. The figure also shows a mapping between the functional blocks of the architecture and the different modules of the telemetry system. In the following clause, the functional blocks and their interfaces are described in detail.

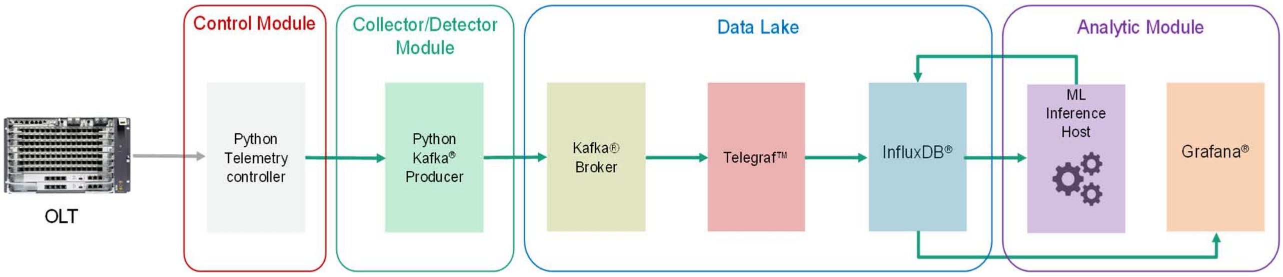


Figure B.1: Functional blocks of the proposed telemetry streaming workflow and ML pipeline

B.2 Control Module

Telemetry provides a mechanism to select data of interest from controller/collector and to transmit it in a structured format to remote telemetry system for monitoring. The telemetry controller in Figure B.1 is responsible for telemetry setup and configuration on the OLT side. The configuration data transmitted from the telemetry controller to the OLT can be modelled using open source or proprietary YANG data models. A telemetry session is initiated using one of the three modes specified in clause 5: UDP Streaming Telemetry mode, gRPC® Static Telemetry mode and gRPC® Dynamic Telemetry mode.

In UDP Streaming Telemetry mode and gRPC® Static Telemetry mode, the controller sends configurations through NETCONF to the OLT. In this mode, sensor-paths and address of collectors are configured and bound together into one or more subscriptions. The model of configuration is typically with a telemetry configuration YANG model. The specifications of the YANG models are for further study. There are three steps for the telemetry collection configuration of OLT with these two modes:

1) **Create a sensor-group.** The sensor group represents a reusable grouping of multiple sensor paths and excludes filters.

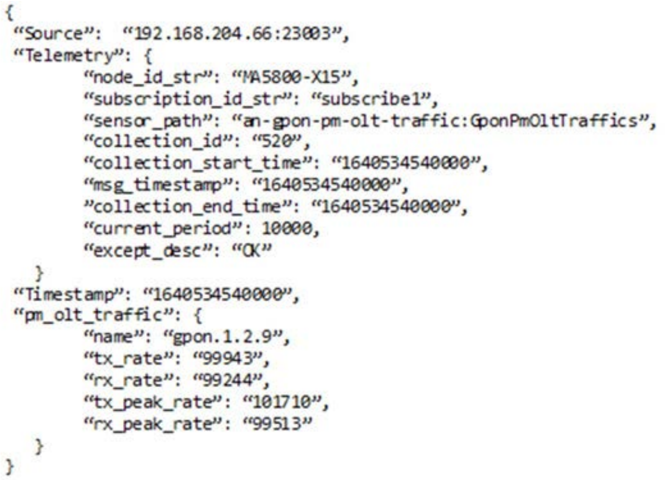
2) **Create a destination group.** The destination group contains the destination IP addresses and UDP/TCP ports of several collectors.

3) **Create a subscription.** The subscription associates destination groups, several sensor-groups with streaming time interval, encoding format and transport protocol.

In gRPC® Dynamic Telemetry mode, the collector creates a gRPC® session with the OLT and subscribes dynamically to one or more sensor-paths. The subscription interface is typically based on a Protobuf model. The Protobuf model specifies two RPC interfaces. They are typically "Subscribe" and "Cancel".

B.3 Collector/Detector Module

The telemetry data producer is responsible for data collection, decoding, and pre-processing before feeding it to the broker. The telemetry data producer acts as an intermediary Data Pre-processing Unit (DPU). The DPU is largely used for adjustment of telemetry data format according to pre-established requirements (parameter filtering, event format adjustment, if different from the one provided by the OLT), as well as acting as a data serializer, preparing the event series to be written into the data lake. Depending on the configured telemetry session, the collected telemetry data is carried using UDP or gRPC® transport/streaming protocols, while encoded using predefined JSON or Protobuf model-based encoding formats. If the telemetry is received in a binary message, the telemetry data producer decodes it into JSON objects. One example of Kafka® producer output format is shown in the Figure B.2.



```
{
 "Source":  "192.168.204.66:23003",
 "Telemetry": {
        "node_id_str": "MA5800-X15",
        "subscription_id_str": "subscribe1",
        "sensor_path": "an-gpon-pm-olt-traffic:GponPmOltTraffics",
        "collection_id": "520",
        "collection_start_time": "1640534540000",
        "msg_timestamp": "1640534540000",
        "collection_end_time": "1640534540000",
        "current_period": 10000,
        "except_desc": "OK"
   }
 "Timestamp": "1640534540000",
 "pm_olt_traffic": {
        "name": "gpon.1.2.9",
        "tx_rate": "99943",
        "rx_rate": "99244",
        "tx_peak_rate": "101710",
        "rx_peak_rate": "99513"
   }
}
```

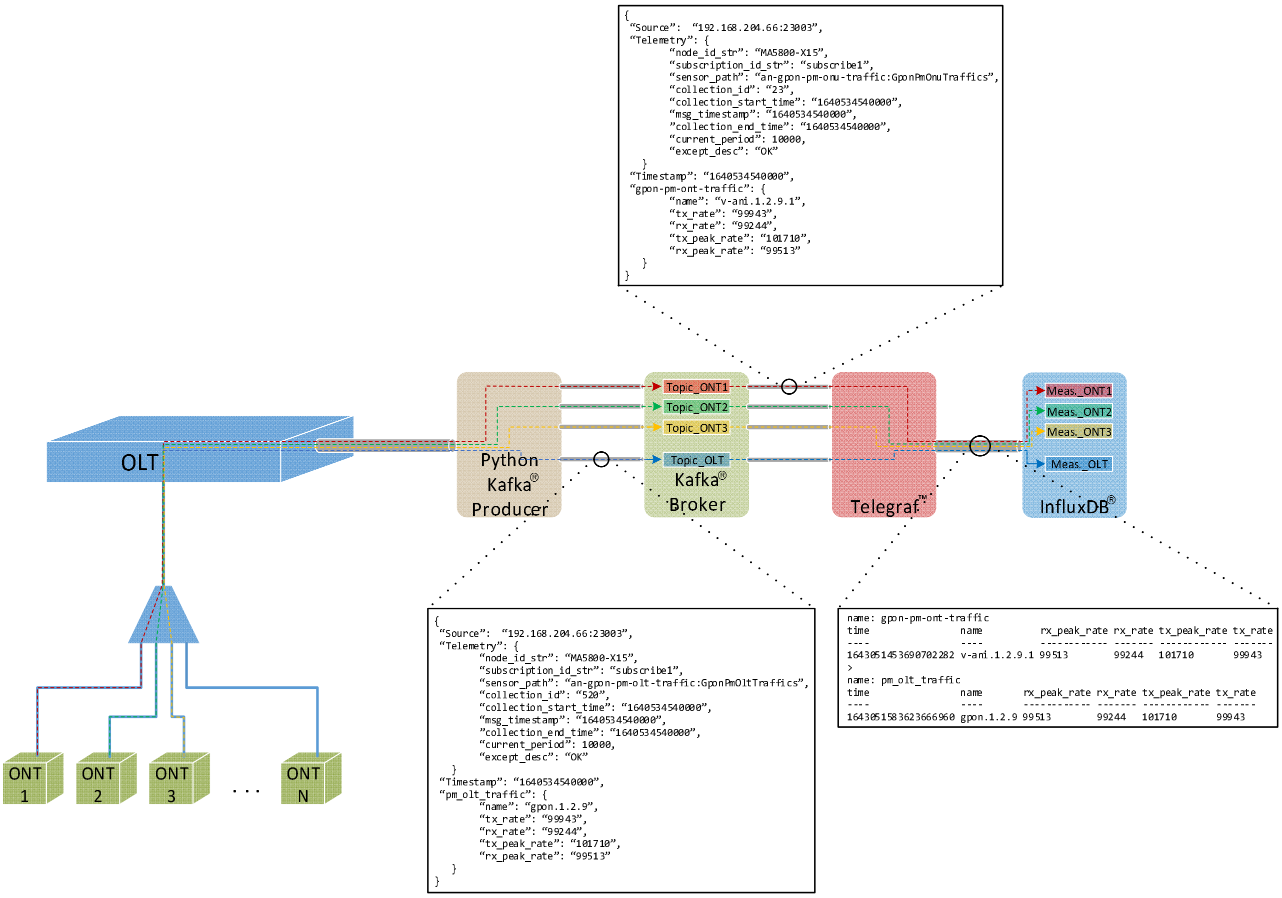
Figure B.2: An example of the Python Kafka® producer (see Figure B.1)

There are different ways to write the telemetry data into a data lake. The Kafka® can be used distributed event-streaming technology, which comprises a Kafka® producer, broker and consumer. The DPU functionality is implemented through a custom Python-based Kafka® producer (see Figure B.1), which has the following functions:

- It acts as an interface/API between the telemetry controller and the Kafka® broker.
- It defines the topics in which the different events are to be written, based on the sensor paths of the OLT. As such, the different event-sets distinguished by their corresponding sensor paths that generated them, are written into specific pre-defined Kafka® topics hosted by the Kafka® broker.

In this annex, the data lake is generated using a telemetry broker (Kafka®) and Time Series Database (TSDB) (InfluxDB®) that are interfaced together using a telemetry consumer (Telegraf™). One of the proposed architectures is illustrated in Figure B.3.

One instance of the telemetry data is referred to as an event, which can be comprised of a multitude of device- or network-related parameters at a particular point in time, such as transmission or reception data rates through a particular port, transmitted or received traffic volume, number of dropped packets, or BER, to name a few. An event can also be a complex package comprised of a variety of such network parameters. Events typically occur at constant time intervals, which are defined by the sampling frequency of the device sensor measuring the aforementioned parameters. As a result, the device generates a discrete series of events distinguished by an individual timestamp. The timestamp is typically preserved together with the event it describes, and acts as unique identifier throughout the entire propagation along the pipeline up to its use by the analytics module. It is assumed that the retrieved timestamp of the OLT is streamed along the whole pipeline to avoid any discrepancy. The specification of the data lake is described in clause B.4.



```
{
 “Source”:  “192.168.204.66:23003”,
 “Telemetry”: {
        “node_id_str”: “MA5800-X15”,
        “subscription_id_str”: “subscribe1”,
        “sensor_path”: “an-gpon-pm-onu-traffic:GponPmOnuTraffics”,
        “collection_id”: “23”,
        “collection_start_time”: “1640534540000”,
        “msg_timestamp”: “1640534540000”,
        ”collection_end_time”: “1640534540000”,
        “current_period”: 10000,
        “except_desc”: “OK”
   }
 “Timestamp”: “1640534540000”,
 “gpon-pm-ont-traffic”: {
        “name”: “v-ani.1.2.9.1”,
        “tx_rate”: “99943”,
        “rx_rate”: “99244”,
        “tx_peak_rate”: “101710”,
        “rx_peak_rate”: “99513”
   }
}
```

```
{
 “Source”:  “192.168.204.66:23003”,
 “Telemetry”: {
        “node_id_str”: “MA5800-X15”,
        “subscription_id_str”: “subscribe1”,
        “sensor_path”: “an-gpon-pm-olt-traffic:GponPmOltTraffics”,
        “collection_id”: “520”,
        “collection_start_time”: “1640534540000”,
        “msg_timestamp”: “1640534540000”,
        ”collection_end_time”: “1640534540000”,
        “current_period”: 10000,
        “except_desc”: “OK”
   }
 “Timestamp”: “1640534540000”,
 “pm_olt_traffic”: {
        “name”: “gpon.1.2.9”,
        “tx_rate”: “99943”,
        “rx_rate”: “99244”,
        “tx_peak_rate”: “101710”,
        “rx_peak_rate”: “99513”
   }
}
```

```
name: gpon-pm-ont-traffic
time                name          rx_peak_rate rx_rate tx_peak_rate tx_rate
----                ----          ------------ ------- ------------ -------
1643051453690702282 v-ani.1.2.9.1 99513        99244   101710       99943
>
name: pm_olt_traffic
time                name       rx_peak_rate rx_rate tx_peak_rate tx_rate
----                ----       ------------ ------- ------------ -------
1643051583623666960 gpon.1.2.9 99513        99244   101710       99943
```

NOTE: The distribution of events into different topics based on their origin source/sensor path and further translation of events into measurements into the TSDB.

Figure B.3: Telemetry pipeline architecture and the events writing procedure

B.4 Data Lake

B.4.1 Overview

The proposed architecture (see Figure B.3) considers a data lake based on three main blocks:

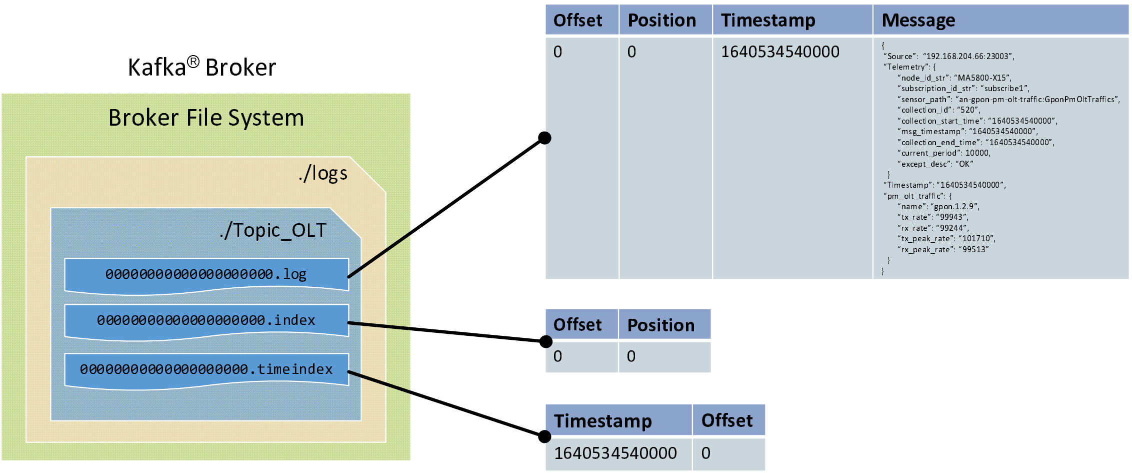
- A telemetry broker.
- A telemetry consumer.
- A TSDB.

The next sub-sections provide detailed information on the telemetry flow and storage in the considered data lake.

B.4.2 Telemetry Broker

The telemetry broker acts as an intermediary event storage from where the data can be retrieved and processed by different consumers. The Kafka® broker is assumed to hold multiple so-called topics, which in turn can be divided into different partitions. Events are organized and stored in Kafka® topics that represent a category of events, which may share some similarities in terms of their nature or origin source. They are defined by a particular name that is unique across the entire Kafka® cluster (a group of Kafka® brokers). In order to write telemetry data from the Collector Module to the Kafka® broker, the events are written by the Kafka® producers into topics, these are then stored for a certain amount of time, the consumers read the data from these topics. The earlier mentioned partitions dividing a topic are used to accelerate event writing and reading by multiple producers and consumers in parallel (parallelization), which ensures a high data availability at all times. Due to its high fault tolerance, accessibility, scalability and data consistency, the Kafka® broker is used for reliability purposes, to guarantee no data loss in the case the connection to the TSDB gets interrupted.

It is worth noting the structure of the topic in Figure B.4. The topic data/events are stored in log-files, where predefined parameters, such as message offset and position. The message offset and position indicate the order of events and their length in the sequence of stored events. A timestamp keeps the event generation time, followed by the event payload itself, i.e. the event message.



Offset	Position	Timestamp	Message

0	0	1640534540000	{
			"Source": "192.168.204.66:23003",
			"Telemetry": {
			"node_id_str": "MA5800-X15",
			"subscription_id_str": "subscribe1",
			"sensor_path": "an-gpon-pm-olt-traffic:GponPmOltTraffics",
			"collection_id": "520",
			"collection_start_time": "1640534540000",
			"msg_timestamp": "1640534540000",
			"collection_end_time": "1640534540000",
			"current_period": 10000,
			"except_desc": "OK"
			}
			"Timestamp": "1640534540000",
			"pm_olt_traffic": {
			"name": "gpon.1.2.9",
			"tx_rate": "99943",
			"rx_rate": "99244",
			"tx_peak_rate": "101710",
			"rx_peak_rate": "99513"
			}
			}

Offset	Position
0	0

Timestamp	Offset
1640534540000	0

NOTE: The structure of a Kafka® topic with the corresponding log files.

Figure B.4: Data/event storage in the Broker File System

The architecture in Figure B.3 considers a per-source writing, in which, every sensor of the OLT or ONU generating a set of parameters (see Figure B.4) that has a dedicated Kafka® topic assigned to it, in which the streamed telemetry data is stored. From these topics, the stored data is then fetched by the Kafka® consumers for its subsequent processing. It is important to emphasize that with the high horizontal scalability of the Kafka® brokers, this sensor source - topic assignment would also be feasible for larger networks comprising a multitude of NEs, each streaming its own set of telemetry data flows.

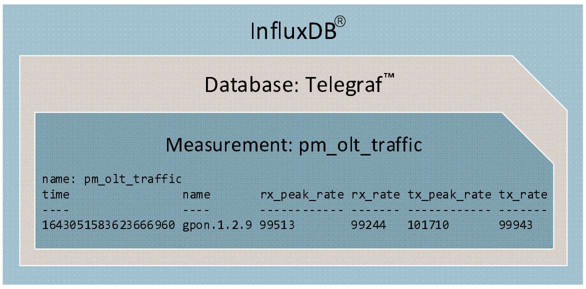
The proposed particular solution (see Figure B.1) is a tailored use-case that involves the progression of telemetry parameters over time (e.g. traffic monitoring for traffic forecasting), for which the dimension of time plays the key role. While the telemetry broker can act as the heart of the data lake providing services to numerous data consumers. The use of a reliable and efficient TSDB that can make it very easy to interface the data lake to the analytic module. Discussion of the envisioned TSDB, based on InfluxDB®, and the telemetry consumer that interfaces the telemetry broker to the TSDB will be covered in clause B.4.3.

B.4.3 Telemetry Consumer

Once the telemetry data is stored in the telemetry broker it is processed by a telemetry consumer. The telemetry consumer is an application that subscribes to a topic or multiple topics to read and process the stored events. Eventually the main telemetry consumer is the Analytic Module. However, the telemetry data has to be first written in a TSDB. The proposal considers InfluxDB® as a sophisticated TSDB to be used. Then, in order to interface the telemetry broker with the TSDB, Telegraf™ is considered as a Kafka® consumer. Telegraf™ is an agent for collecting and reporting of metrics, events, logs and traces, and converts any data format of the Kafka®-stored events into Line Protocol (LP) entries for their subsequent writing/recording into the TSDB. The Telegraf™ agent is defined by its configuration file, which specifies the input event data format(s), sets the collection interval/writing frequency into the TSDB. The Telegraf^{TM} agent defines the real input (Kafka® broker) and output (TSDB) of the services/applications with their corresponding IP addresses and TCP/UDP port numbers.

B.4.4 Time Series Data Base

A TSDB is a database optimized for storing and serving time-stamped data, i.e. associated pairs of values and times. InfluxDB® is a purpose-built TSDB for storage and retrieval of time series data from which the Analytic Module retrieves its data for the subsequent data analytics. As previously mentioned, the writing of events into Kafka® topics is based on the sensor path (origin of these events). As such, every set of parameters streamed by a particular sensor within the ONU or OLT is written into a dedicated topic, which is subsequently translated to its specific InfluxDB® measurement counterpart using Telegraf™'s mapping policy, defined in its configuration file. An instance of the InfluxDB® database with a measurement example is presented in Figure B.5.



```
name: pm_olt_traffic
time                name        rx_peak_rate rx_rate tx_peak_rate tx_rate
----                ----        ------------ ------- ------------ -------
1643051583623666960 gpon.1.2.9 99513        99244   101710       99943
```

Figure B.5: Structure of an InfluxDB® database with a corresponding measurement example

InfluxDB® automatically assigns a timestamp to all of its incoming events, but it can also be configured to utilize the timestamps previously assigned by other components of the telemetry pipeline. It is assumed that the retrieved timestamp of the OLT is streamed along the whole pipeline to avoid any discrepancy.

Figure B.6 shows an alternative to the architecture specified in Figure B.3 for further scalability, the system could use multiple partitions (i.e. the usage of one topic with multiple partitions) similar to the one presented in Figure B.6. Then, multiple Telegraf™ instances can process the topic using the same Consumers groups. In this case, the numbers of Telegraf™ instances are smaller than the number of partitions. Based on real-life implementation of this architecture (i.e. a Kafka® cluster with 3 servers, 12 partitions and 4 Telegraf™ instances) the telemetry streaming workflow is able to manage over 200 million messages per hour. Of course, this performance is highly related to the environment and configurations for which is tested.

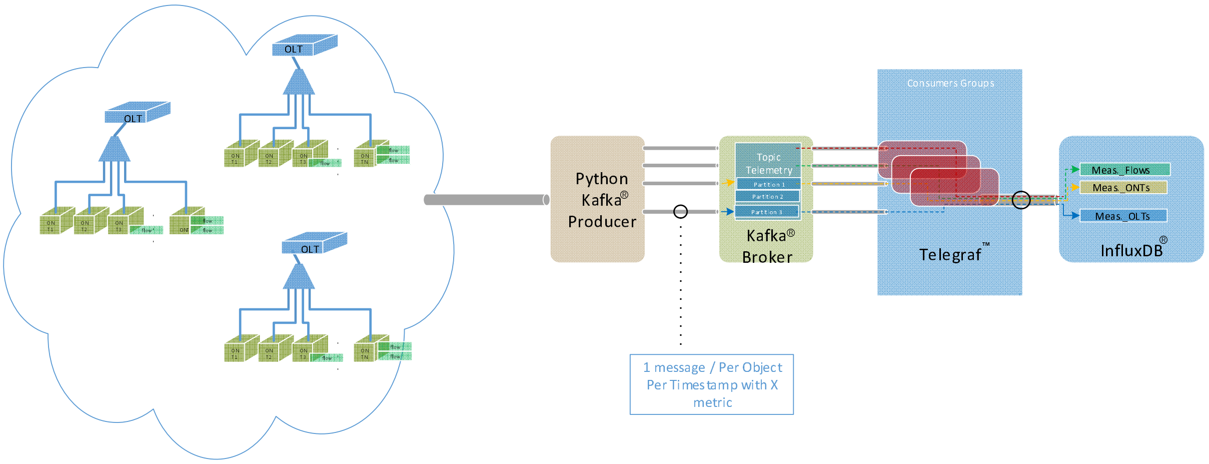


Figure B.6: An alternative approach for data lake creation

B.5 Analytic Module

B.5.1 Overview

The analytic module has two main components (see Figure B.7): an ML inference host and a visualization dashboard based on Grafana®.

B.5.2 ML Inference Host

The ML inference host is an application providing forecasts based on retrieved telemetry data. The application includes a forecasting function, ML model repository and an input/output module. The forecasting function comprises data pre-processing, forecasting, and postprocessing. The input/output module includes a Kafka® consumer and producer and a TSDB client. Figure B.7 shows the architecture of the ML inference host.

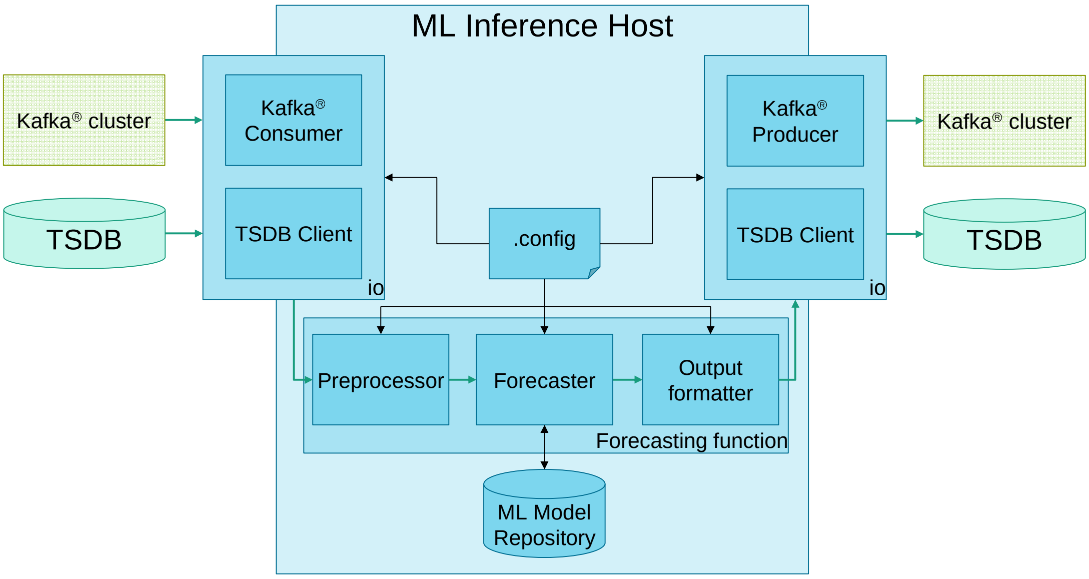


Figure B.7: Architecture of the ML inference host

The ML model repository stores model artifacts, which the ML inference host selects and executes for performing predictions. The model artifacts can be created in independent ML training loops and then imported to the inference host. The I/O module can be configured to either use the TSDB client or the Kafka® consumer/producer for retrieving telemetry data and delivering forecasts. The TSDB client connects to the TSDB of the telemetry pipeline to query data that the forecasting model requires and to insert the predictions of the forecasting model to the time series database. The TSDB client needs to be configured in alignment with the selected ML model, such that the client queries data in the time window relevant to the ML model. The Kafka® consumer receives events from a specific Kafka® topic and buffers their content until enough data is retrieved for the ML model. The Kafka® consumer then forwards the retrieved data to the pre-processor. The Kafka® producer pushes the predictions of the forecasting model to the connected Kafka® cluster. The network administrator can configure the ML inference host's components using a configuration file.

The architecture of the ML inference host is use-case agnostic such that multiple instances of inference hosts utilized for different use-cases are identical in their structure and interfaces but should include different models in the ML model repository suitable for the respective scenario. Depending on the desired use-case, an appropriate model should be selected. The selection of the ML model determines the inputs and outputs of the forecaster from which the pre- and post-processor infer their configuration in combination with the input and output requirements specified by the user. The ML model artifacts should include descriptions of the required input and the provided output.

The performance of the ML inference host depends not only on the quality of the models available in the ML model repository but also on the telemetry data (i.e. its granularity and precision) and the performance and interoperability of the other pipeline components. The telemetry pipeline needs to ensure availability of measurements compliant with the used ML model requirements. Especially, the sampling interval, the measurement accuracy and consistency, and the telemetry delay are the constraining factors. For instance, if the sampling interval requirement of the ML model is below the time difference of observations stored in the TSDB or if observations are missing, the input data of the ML model is less accurate and thus the prediction can be less reliable than expected.

B.5.3 Visualization Dashboard

While the performance of ML models can be evaluated with numerous metrics, their performance in operation can be monitored in a human interpretable way using visualization tools. Therefore, the proposed analytic module envisions a Grafana® based dashboard, which can be easily interfaced with the InfluxDB® instances to visualize not only the streamed telemetry data, but also the output of the ML inference host. As an example, Figure B.8 shows a single plot of the visualization dashboard included in the implemented telemetry system that plots the port Rx Peak Rate and its forecasted value for 10 steps in the future.

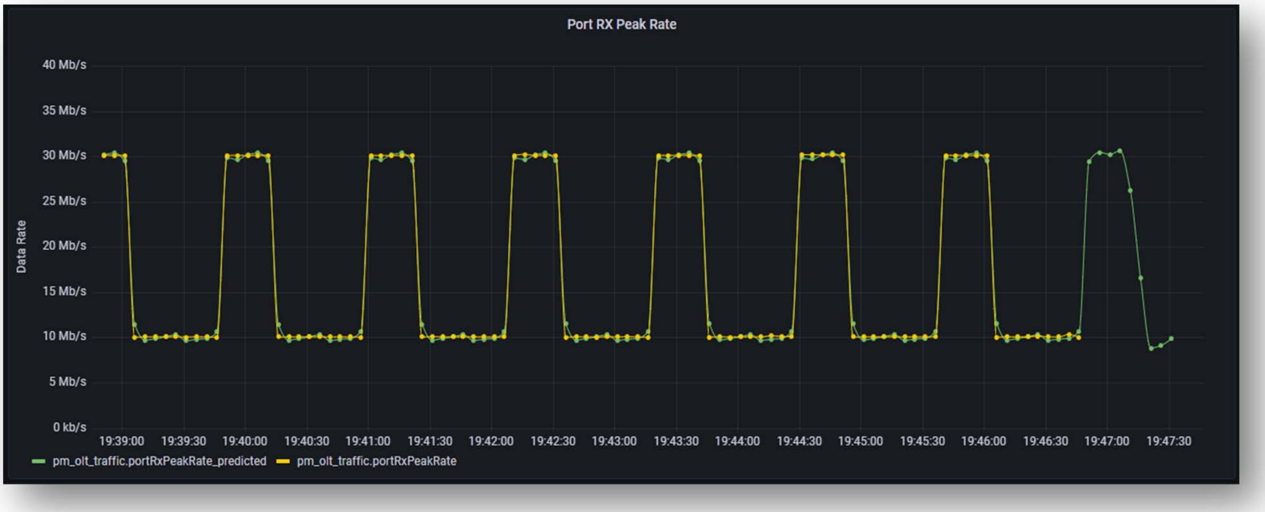


Figure B.8: Evolution of a single telemetry parameter (i.e. port Rx Peak Rate) and its forecasted value over time

Annex C (informative): Feasible Implementation of an Extension of the Telemetry Collection Encoding

C.1 Introduction

For smaller telemetry message size and more explicit information in the telemetry message, this clause makes a feasibility analysis of an implementation extension for telemetry collection encoding using GPB®. Encoding methods other than GPB® are not considered in the present annex.

C.2 Implementation Details

A telemetry data message consists of several collection items (for details refer to the Google® document [i.3] and the GPB® encoding structure). A collection item consists of a data identification block and a data value block as shown as the Figure C.1. A data block consists of several consecutive bytes. The Most Significant Bit (MSB) of each byte indicates whether the data of this block is complete (1: incomplete, 0: complete), and the remaining 7 bits indicate the data content. The last three bits of the last byte in the data identification block identify the data type, and the remaining preceding bits (identify the data label.

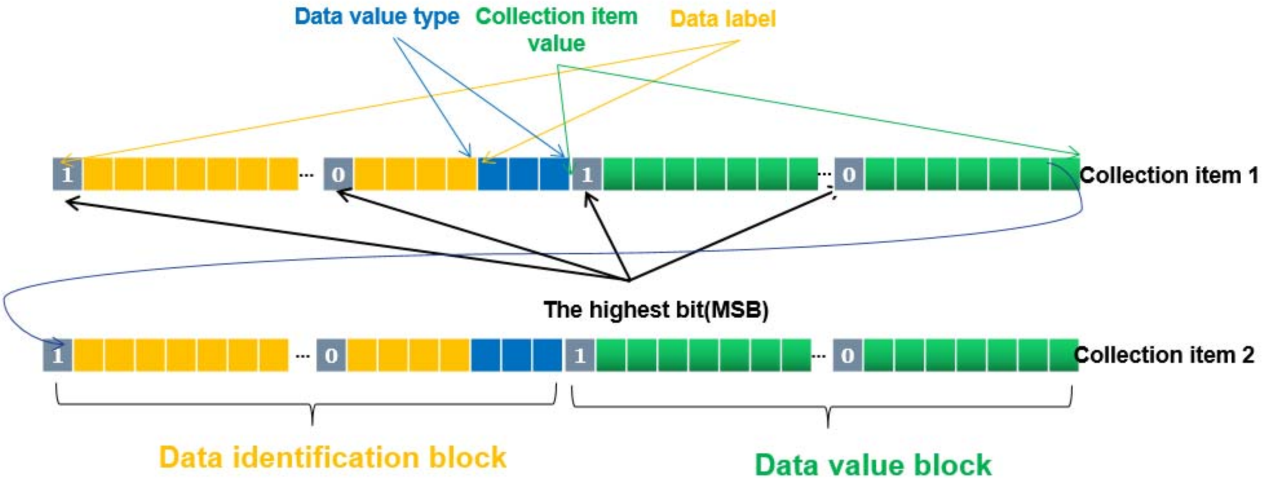


Figure C.1: Encoding Data Structure of Collection Items

The data type is a 3 bits identifier to indicate the data type of one collection item. The data type can be selected according to Table C.1 and its corresponding data value block is the data content or the exception reason. The first six data value type represents the data value block of this encoded collection item and are absolute values. When one collection item's data value type is '6', it represents that an exception has occurred in this collection item and it cannot be sampled by the OLT. The reason for the exception occurrence can be generated in the corresponding data value block of this encoded collection item.

NOTE: This requires a change of the GPB® implementation and does not influence normal data encoding.

Table C.1: Data Value Type

Data value Type	Meaning	Used For
0	Varint	int32, int64, uint32, uint64, sint32, sint64, bool, enum
1	64-bit	fixed64, sfixed64, double
2	Length-division	string, bytes, embedded messages, packed repeated fields
3	Start group	groups
4	End group	groups
5	32-bit	fixed32, sfixed32, float
6	Exception	It is string type and lists the reason for why it cannot be sampled

Annex D (informative): Change History

Date	Version	Information about changes
November 2022	V1.1.1	First published version

History

Document history		
V1.1.1	November 2022	Publication