Distributed Architecture Supporting Intelligent Optical Measurement Aggregation and Streaming Event Telemetry

Pol González[1], Ramon Casellas[2], Jose-Juan Pedreno-Manresa[3], Achim Autenrieth[3], Fabien Boitier[4], Behnam Shariati[5], Johannes K. Fischer[5], Marc Ruiz[1], Jaume Comellas[1], and Luis Velasco[1*]

[1] Universitat Politècnica de Catalunya, Spain; [2] CTTC, Spain; [3] ADVA Optical Networking Germany; [4] Nokia Bell Labs, France; [5] Fraunhofer HHI, Germany
e-mail: luis.velasco@upc.edu

Abstract: A distributed telemetry system integrating optical measurement and event data collection is demonstrated. Measurements of optical spectra from Nokia Bell Labs, of optical transponders from ADVA and SDN controller events from CTTC will be showcased. © 2023 The Authors

1. Overview

Data collected from observation points in the devices (*measurements*) are typically sent to a central system for further analysis [1]. Although protocols specifically devised for telemetry, like gRPC, can reduce data volume, scalability is an issue because of the huge amount of measurement data to be collected with high collection frequency for each observation point.

In addition, *events* generated by applications/platforms (e.g., Software Defined Networking (SDN) controllers and management systems) can be used to keep consistency among systems. In fact, an event streaming mechanism is made available as an alternative to traditional notifications. The streaming capability is distinct from Transport API (TAPI) notifications [2] and is designed to better deal with scale and to provide an improved operational approach. In this context, any component of the SDN control plane may act as a source of event telemetry, which should be transported and distributed unaltered to other systems in the control and management planes.

In this demonstration, we will showcase a telemetry architecture designed in the H2020 B5G-OPEN project [3] that supports both telemetry families: measurements and events [4]. For the former, intelligent data aggregation and feature extraction are placed nearby data collection sources to reduce data volumes, whereas event telemetry is transported transparently. Specifically, the demo will show integration of: *i*) heterogeneous *measurements* from an optical spectrum analyzer (OSA) collected from the Nokia Bell Labs testbed located near Paris (France) and from commercial ADVA optical transponders (TP) in a testbed in the Fraunhofer HHI premises in Berlin (Germany) [5]; and *ii*) connection set-up and teardown *events* from an SDN controller located in CTTC premises near Barcelona (Spain).

2. Innovation

The demonstration will show a multivendor unified system for measurements and event telemetry with distributed data processing. The demonstration presents timely innovations related to the integration of heterogeneous telemetry data sources, which will enable a complete network vision and help to simplify operation in complex network scenarios. The main innovations of this demonstration are: *i*) automatic integration of heterogeneous telemetry data sources from different vendors; a demarcation point is defined to manage responsibilities and for supervision purposes; *ii*) different levels of distributed data processing (from pure streaming to intelligent data aggregation and feature extraction) are available to greatly reduce measurements telemetry data volumes; *iii*) unified transport of measurements and events, which allows dimensioning IT and networking resources dedicated to telemetry; and *iv*) telemetry-family specific storage and distribution to external systems to separate back measurements and events.

3. OFC Relevance

Availability of data is a requirement for network automation purposes, including Machine Learning training, degradation and anomaly detection, and distributed control and management systems consistency. Therefore, telemetry solutions that allow collecting massive amount of data volumes and perform in-site data processing or transparent data streaming are key enablers. However, it is still not clear *how often* telemetry data needs to be collected, *where* they need to be processed, *how* their volume can be reduced. The demonstration will answer these and other related questions, and, in consequence, it will attract the community's attention and will be of interest to a broad OFC audience. In particular, this demonstration is specifically designed for those operators and vendors interested in network automation solutions.

The research leading to these results has received funding from the H2020 B5G-OPEN (G.A. 101016663) and the MICINN IBON (PID2020-114135RB-I00) projects and from the ICREA Institution.

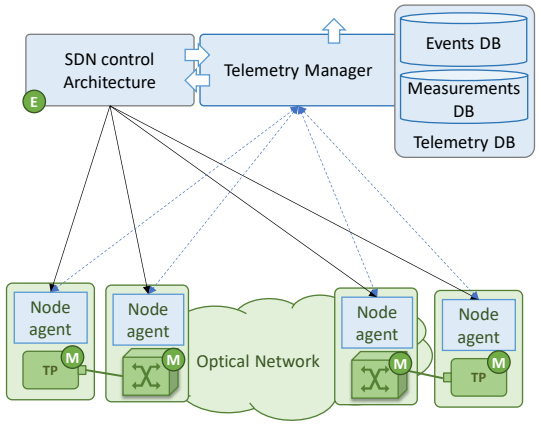

Fig. 1: Overall network architecture

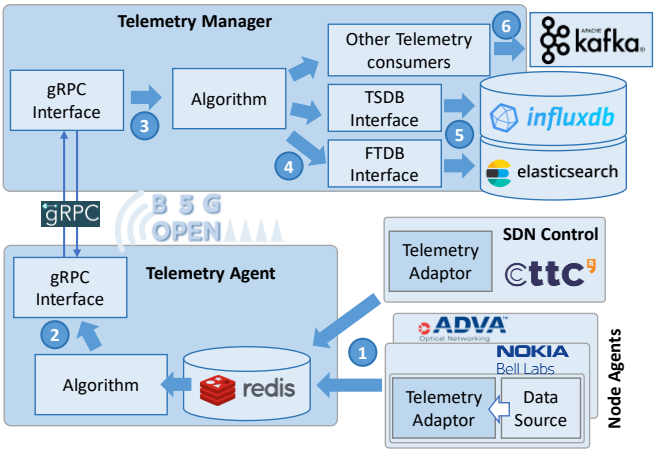

Fig. 2: B5G-OPEN Telemetry architecture and workflow

4. Demo content & implementation

The objective of this demo is to show integration of multivendor measurements and event telemetry in a single telemetry system. Intelligent measurement telemetry data aggregation will be demonstrated to reduce connectivity requirements between telemetry data sources and the centralized telemetry repository. In addition, event telemetry will be transparently transported, which allows ensuring consistency of different network control and management systems. In particular, three different data sources from Nokia Bell Labs, ADVA TPs and CTTC SDN controller will be integrated.

Fig. 1 presents the reference network scenario, where an SDN architecture controls a number of TPs and optical nodes, in the data plane. Note that the SDN architecture might include a hierarchy of controllers, including optical line systems and parent SDN controllers. A centralized telemetry manager is in charge of receiving, processing and storing telemetry data, including measurements and events. The telemetry database (DB) includes two repositories: *i*) the measurements DB is a time-series (TS) DB that stores measurements; and *ii*) the events DB is a free-text search (FT) engine. In addition, telemetry data can be exported to other external systems. Every node in the data plane is locally managed by a node agent, which translates the control messages received from the related SDN controller into operations in the local node and exports telemetry data collected from observation points (labelled M) enabled the optical nodes or in specific optical devices, like OSAs. In addition, events can be collected from applications and controllers (labelled E).

A detailed architecture of the telemetry system is presented in Fig. 2. The internal architecture of telemetry agents and the telemetry manager is shown. Telemetry agents can be integrated with node agents in the case of measurements telemetry, or be deployed as separate elements for event telemetry. Internally, both, telemetry agents and telemetry manager are based on three main components: *i*) a *manager* module configuring and supervising the operation of the rest of the modules; *ii*) a number of components that include *algorithms*, e.g., data processing, aggregation, etc. and *interfaces*, e.g., gRPC; and *iii*) a *Redis DB* to communicate the different modules among them. This solution provides an agile and reliable environment that simplifies communication, as well as integration of new modules. A gRPC interface is used for the telemetry agents to export telemetry to the telemetry manager, as well for the telemetry manager to tune the behaviour of algorithms in the agents.

Let us describe now the telemetry workflow, which is valid for a wide range of measurements and event telemetry use cases. For measurement telemetry, the node agent includes modules (named *data sources*) that gather measurements from observation points in the optical nodes. A telemetry adaptor has been developed, so data sources can export collected data to the telemetry system; specifically, the adaptor receives raw data from the data source and generates a structured JSON object, which is then published in the local Redis DB (labelled 1 in Fig. 2), which acts as a *demarcation point*. A number of algorithms can be subscribed to the collected measurements. As an example, let us assume that only one algorithm is subscribed, which processes the measurements locally. Such processing might include doing: *i*) no transformation on the data (*null algorithm*); *ii*) some sort of data aggregation, feature extraction or data compression; or *iii*) some inference (e.g., for degradation detection). The output data (transformed or not) are sent to a gRPC interface module (through the Redis DB) (2), which conveys the data to the telemetry manager. As for event telemetry, events generated in an SDN controller (or other system), are injected in the telemetry agent and transported through the gRPC interface transparently to the telemetry manager. In the telemetry manager, data received by a gRPC interface module is published in the local Redis DB, so subscribed algorithms can receive them. Once

processed, the output data are published in the local Redis DB (4) and can be stored in the Measurements DB (we use InfluxDB) or the events DB (we use Elasticsearch) (5) and/or be exported to external systems like Apache Kafka (6).

In the demo, multivendor data source integration will be demonstrated with two testbeds producing measurements telemetry and an SDN controller generating event telemetry. The Nokia Optical Network testbed combines more than 400 km optical fiber with 7 optical nodes. OSAs are deployed at each node ingress and egress ports and collected data are injected in the telemetry system. Each telemetry measurement is injected as a JSON object with the structure in Fig. 3a. Samples are processed by a feature extraction algorithm that characterizes the mean (μ) and the standard deviation (σ) of the power around the central frequency ($fc \pm \Delta f$), as well as a set of features computed as: *i*) equalized noise level (e.g., -60dB + equalization level); *ii*) edges of the signal computed using the derivative; and *iii*) a family of power levels computed with respect to μ minus a number of dB. Each of these power levels generates a couple of cut-off points $f1_{(\cdot)}$ and $f2_{(\cdot)}$ [6].

A different optical network testbed in Fraunhofer HHI includes ADVA commercial TPs that generate telemetry measurements as JSON objects with the structure in Fig. 3b. Intelligent data aggregation on the received samples will look for meaningful changes on the measurements to decide whether samples are to be conveyed immediately to the telemetry manager or averaged and sent with larger periodicity on the contrary.

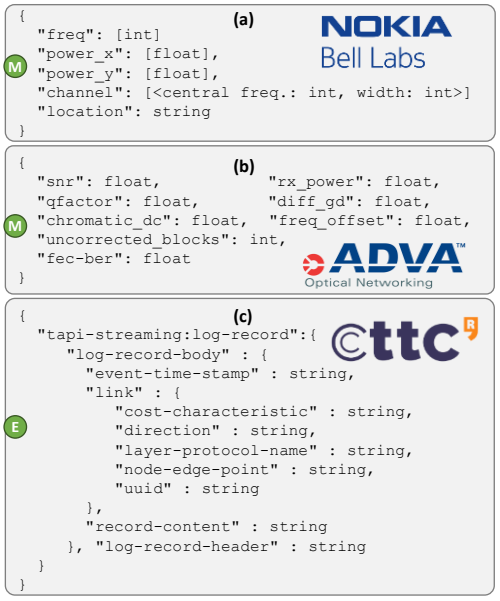

Fig. 3: Measurement telemetry from Nokia (a) and ADVA (b) data sources and event telemetry from CTTC (c)

Finally, the SDN controller runs in CTTC and reports changes of state of the controlled system to another (usually superior) management-control entity. In this case, events are TAPI entities, i.e., YANG sub-trees and allow a client to achieve and maintain eventual consistency with the state of the controlled system (Fig. 3c). In particular, the TAPI proxy SDN controller will act as data source. For this, the internal architecture of the software is modified to report asynchronous events that happen in the network.

The telemetry system will run in 4 Virtual Machines (VM) deployed in the Fraunhofer HHI infrastructure using OpenStack as the virtual infrastructure manager and Ubuntu Server 22.04 LTS as operating system. All the software, including the manager, algorithms, interfaces and the telemetry adaptor, have been implemented in Python and will be executed using Python 3.10.4. Every telemetry agent and the manager with their respective Redis DB instances run inside Docker containers and will be deployed using Docker Compose. Containerized versions of Influx DB 2.4.0 as measurements DB and of Elasticsearch 8.3.3 as Events DB, will be deployed. To visualize data, Grafana 9.1.1 and Kibana 7.14 both running in Docker containers are used. In addition, a Web UI that offers a general view of telemetry system has been implemented in Python using Django 4.1 and runs locally in the device supporting the demonstration.

The workflow of the demonstration will be as follows: *i*) measurements from OSAs in the Nokia Bell Labs testbed and ADVA TPs in the Fraunhofer HHI one will be examined at the demarcation point; data samples are received in the Redis DB in the related telemetry agent every few seconds and features are extracted; *ii*) activity (connections set-up and teardown) in the SDN controller will generate events, which will be examined once received in the Redis DB; *iii*) measurements data aggregation will be demonstrated using a simple but effective algorithm that reduces the number of samples actually conveyed to the telemetry manager; *iv*) evolution of measurements will be visualized in Grafana and events in Kibana; *v*) a synthetic measurements telemetry data source will be used to reproduce a degradation in an optical connection, which will be detected first in the algorithm performing intelligent data aggregation in local telemetry agent, which will convey all the received samples to the telemetry manager; the change in the resolution of the measurements will be visualized in Grafana to demonstrate the usefulness of intelligent data aggregation. The Web UI will facilitate iteration of the attendees to the system, so they can start, stop, and modify the configuration of the different components of the architecture.

References

[1] L. Velasco *et al*., "Monitoring and Data Analytics for Optical Networking: Benefits, Architectures, and Use Cases," IEEE Network, 2019.
[2] OIF: SDN Transport API, [On-line] https://www.oiforum.com/technical-work/hot-topics/sdn-transport-api-2/.
[3] H2020 Beyond 5G - OPtical nEtwork continuum (B5G-OPEN) project. [On-line] https://www.b5g-open.eu/
[4] L. Velasco, P. Gonzalez, and M. Ruiz, "An Intelligent Optical Telemetry Architecture," submitted to OFC 2023.
[5] B. Shariati *et al*., "Demonstration of latency-aware 5G network slicing on optical metro networks," IEEE/OSA JOCN, 2022.
[6] A. P. Vela *et al*., "Soft Failure Localization during Commissioning Testing and Lightpath Operation," IEEE/OSA JOCN, 2018.