PON Physical Twin: Enabling Third-party Research on FTTH Optimization with Open Datasets

L. Inglés(1,2,3), L. Anet Neto(1,3), C. Rattaro(2), M. Morvan(1,3), A. Castro(2), L. Nuaymi(1)

(1) IMT Atlantique, 655 Av. du Technopole, 29280 Plouzané, France, lucas.ingles-loggia@imt-atlantique.fr
(2) Universidad de la República, 565 Av. Julio Herrera y Reissig, 11300 Montevideo, Uruguay
(3) Lab-STICC CNRS UMR 6285, Technopôle Brest-Iroise - CS 83818, 29238 Brest Cedex 3, France

Abstract *We present a programmable PON testbed using commercial equipment and SDN control, enabling configuration exploration, performance evaluation, and reproducible experiments—fostering open datasets for network optimization and machine learning research.*

Introduction

Passive Optical Networks (PONs) have become a widely adopted solution for providing broadband connectivity to end-users, particularly in residential scenarios. Its greatest advantage is an efficient use of optical fibers, with the typical point-to-multipoint (PtMP) topology allowing service providers to share fiber infrastructure among multiple users via time-division multiplexing/multiple access (TDM/TDMA). The PtMP nature of the optical access is the reason why the performance of PONs is not solely determined by the physical (PHY) transmission layer but also by dynamic bandwidth allocation mechanisms and protocol-imposed constraints, both from the transmission convergence (TC) layer realm, and hardware limitations. All of these can severely impact throughput, latency and packet jitter.

A major challenge to address this issue lies in the opaque, vendor-specific nature of dynamic bandwidth allocation (DBA) algorithms[1]. In order to adapt the upstream priorities of different service flows, PON relies on five types of traffic containers (T-CONT). Those can group services with similar priorities and are used to provide a fair bandwidth allocation for the different users using one same service. However, the details of such algorithms are unknown and cannot be modified by network operators[2], forcing them to investigate system behavior and often make decisions regarding network provisioning with incomplete information. Unable to accurately predict delay, frame loss, or total assigned capacity, operators often resort to oversimplified, set-and-forget engineering rules for the DBA configurations. This uncertainty frequently results in overprovisioning to meet performance targets, cost inflation and underutilization of network resources. While manual adjustments or synthetic probes can be used to optimize PON configuration, trial-and-error risks service degradation, and probes add extra network load that can impact user experience.

In this context, the absence of open datasets based on commercial PON deployments remains a major obstacle to advancing data-driven network optimization. Even the most sophisticated modeling tools and network simulator cannot fully replicate real-world behavior[3], especially when evaluating physical metrics and time-sensitive parameters. This gap between simulated and operational performance limits the effectiveness of research efforts and slows down innovation particularly for academic institutions that lack access to real production environments.

Our work addresses these challenges by providing publicly accessible performance measurements from a real PON using an SDN-enabled testbed with automated monitoring (see Fig. 1). This platform acts as a physical twin of a commercial PON, enabling reproducible experiments for rigorous DBA configuration research and democratizing access to empirical data for researchers lacking physical PON infrastructure. It also serves as a benchmark for comparing vendors and assists operators in planning networks without disrupting service. While previous works have explored testbeds for dataset generation in other domains[4]–[6], to the best of our knowledge, this is the first SDN-orchestrated PON testbed and the first to make its generated datasets publicly available[7], covering a wide range of DBA configurations, traffic profiles, and performance indicators. In this paper, we detail the testbed and SDN controller, describe the dataset, and illustrate its utility through a practical use case.

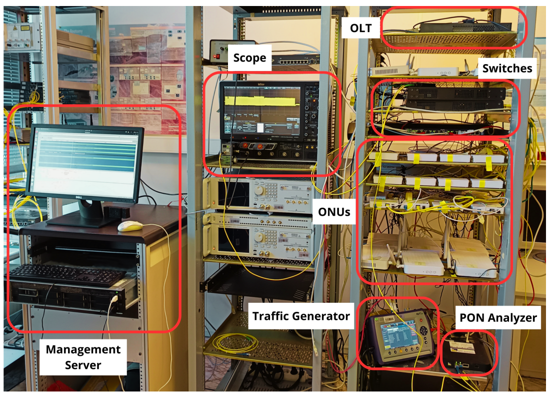

Fig. 1: Photograph of the PON testbed.

PON Testbed

Our experimental infrastructure (Fig.2) is based on four elements: (1) The PON infrastructure, (2) the real-time physical measurement infrastructure, (3) the traffic generation stage and (4) the control and orchestration planes.

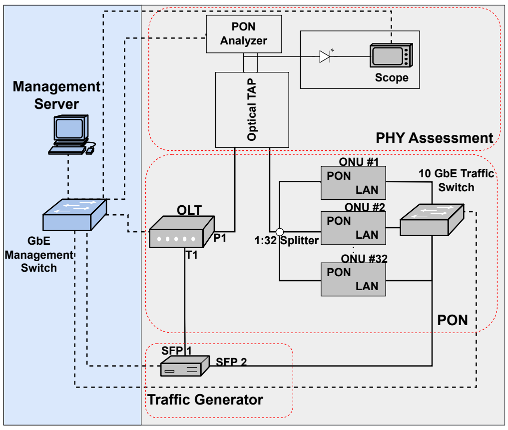

Fig. 2: Experimental setup block diagram.

1. For the PON infrastructure we use a commercial Gigabit-capable PON (G-PON) with an Optical Line Terminal (OLT), 32 Optical Network Units (ONUs) and two optical splitters.
2. We incorporate a measurement stage between the OLT and ONUs, featuring a high-resolution oscilloscope and PON protocol analyzer (sniffer). These instruments provide real-time monitoring of waveform integrity, timing parameters and key metrics at both PHY and TC layers, offering essential visibility into the dynamic behavior of the optical access.
3. A commercial traffic generator creates different traffic patterns, with individual flows per ONU to simulate diverse user demands via an intermediate Ethernet switch using 802.1Q. The OLT's metro-side output loops back to the traffic generator, creating a closed-loop system with synchronized timing references for both up and downstream flows, which is critical for accurate and repeatable performance measurements. The generator also performs evaluation of the transmitted flows and completes the list of available metrics with layers 2 and 3 indicators such as latency, jitter or frame loss rate.
4. A custom-developed control plane, through which all the devices talk with a management entity with the help of a custom-developed orchestrator and SDN framework.

Custom-developed Control Plane

The orchestrator is developed following a modular paradigm, enabling the integration of additional functionalities and promoting interoperability. Fig. 3 details the control plane in our platform. Three different layers can be identified here: (1) The orchestration layer, (2) the SDN controller layer and (3) the physical network functions layer.

The orchestrator can be accessed via a custom graphical user interface (GUI), a web interface, or with custom scripts. These entities communicate with the Policy Manager, which acts as the gateway between the user interface and the control plane server. It provides functions for continuous network state monitoring and dynamic configuration adjustments, enabling performance optimization and adaptation to changing conditions. It communicates with the SDN controller through a RESTful interface.

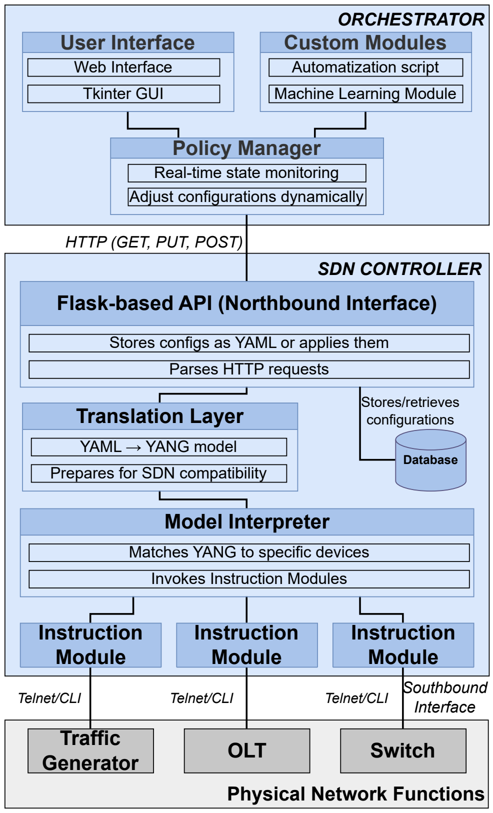

Fig. 3: Control plane architecture.

On the SDN controller side, a Flask API serves as the Northbound Interface, processing incoming requests and interacting with a dedicated database. The Flask API supports various operations, including executing received configurations, storing and retrieving previous configurations, and querying end-network equipment. To ensure interoperability, a translation layer is integrated into the orchestrator, converting human-readable YAML configurations into structured YANG models. This enables compatibility with industry-standard SDN frameworks and facilitates integration with broader network management ecosystems.

A Model Interpreter provides intelligent mediation by mapping abstract service definitions to device-specific parameters. It analyzes incoming configurations and delegates execution to specialized Instruction Modules tailored for different network elements (OLTs, switches, etc.). These modules encapsulate device-specific command sets and communication protocols, abstracting hardware complexity from higher-level operations. The Southbound Interface of the SDN controller enables communication with network equipment through Command-line Interface over Telnet commands, issuing both configuration directives and status queries to maintain synchronization between the control system and network state.

The designed control plane enables dynamic adjustment, continuous monitoring, and automated data acquisition, allowing empirical testing of the network's behavior under different parameter settings and traffic conditions. Its modular architecture not only supports dataset generation but also facilitates the integration of configuration algorithms and the platform's future expansion.

Data Acquisition

The system is capable of retrieving end-to-end performance metrics by querying and parsing statistics from the traffic generator and other metrology equipment: it can collect detailed performance data from the PON itself—by directly querying the OLT and the ONUs—as well as PHY and TC layer KPIs from the PON analyzer and the oscilloscope.

We developed a custom module that uses the orchestrator to systematically enqueue multiple traffic profiles and network configurations (T-CONT types and their configuration parameters) in a loop. The system emulates a 24-hour traffic pattern for 16 ONUs, capturing one measurement every 12 minutes, with each network configuration and KPI acquisition operation taking approximately two minutes. Each point reflects the traffic demand of all ONUs under a specific set of DBA parameters. The control plane interacts with the traffic generator to collect performance data, which is parsed and stored in structured CSV files available in our Git repository[7]. This process results in an open collection of datasets to support research on the impact of T-CONT configuration parameters in PON. Each dataset contains:

- Network configurations and user traffic profiles recorded at regular 12-minute intervals across the emulated 24-hour period.
- Performance results for each combination of configuration and traffic profile, including eighty-eight metrics per direction (uplink and downlink) such as latency, frame loss ratio, bit error rate, throughput, and packet size.
- Traffic patterns derived from both real-world studies[8]–[10] and synthetic scenarios, applied to various network configurations.
- Data stored in multi-indexed Pandas DataFrames[11] for structured analysis.

The dataset structure facilitates reproducible experiments and is suitable for tasks such as performance prediction, configuration tuning, and machine learning-based optimization in PON.

Usage Example

Fig.4 demonstrates the results of a machine learning model trained with the dataset to predict PON uplink delay based on peak information rate (PIR) and different traffic profiles across all ONUs. All ONUs in this example are configured with a T-CONT type 3. In this case, the model predicts the delay for a PIR of 50 Mbps, which was not included in the training data.

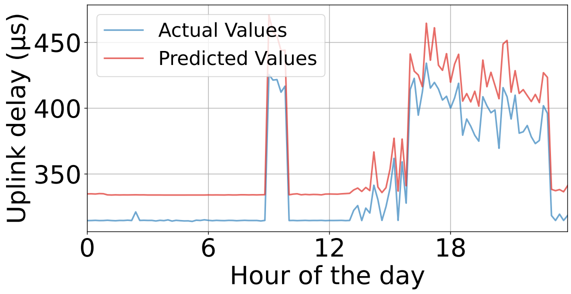

Fig. 4: Comparison of actual and predicted uplink delay across different hours of the day.

The graph in Fig.4 displays uplink delay measurements (in microseconds) plotted against hours of the day. Two data series are presented: actual measured values and predicted values. The results indicate that the prediction model accurately captures the daily pattern of uplink delay fluctuations, with both series exhibiting similar trends and values ranging approximately between 350-450 microseconds throughout the day. Other scenarios closer to network planning use-cases could be easily investigated with our database such as the impacts of changing the peak, committed or fixed information rates of T-CONT profiles or even the optimization of T-CONT type for a specific service.

Conclusions

Our SDN-orchestrated PON testbed provides a programmable platform for transparent performance evaluation in optical access networks. By integrating commercial-grade equipment with SDN control, we enable diverse configurations and optimization strategies in a controlled yet realistic environment. The multi-layered control architecture developed automates experiments and collects performance data across physical, protocol, and application levels, facilitating empirical validation of network behavior and bridging the gap between theoretical models and real implementations. The open datasets generated[7] contribute significantly to the research community, especially for machine learning applications in network optimization, by documenting relationships between configuration parameters and performance metrics under varied traffic conditions.

This platform is designed to support collaborative experimentation, offering a shared infrastructure for advancing the understanding and optimization of optical access networks. New data will be added to the repository as work progresses.

Acknowledgements

This work was funded and supported by the DGE IPCEI ME/CT Orange and the STIC-AmSud RAMONaaS projects as well as the Uruguayan CAP (Comisión Académica de Posgrados). It was also conducted in the framework of the Lab'Optic, a joint research initiative between the UMR-6285 Lab-STICC, the UMR 6082 Foton Institute and Orange Innovation.

References

[1] T. Thangappan, B. Therese, A. Suvarnamma, and G. S. Swapna, "Review on dynamic bandwidth allocation of gpon and epon", *Journal of Electronic Science and Technology*, vol. 18, no. 4, p. 100 044, 2020.

[2] N. E. Frigui, T. Lemlouma, S. Gosselin, B. Radier, R. Le Meur, and J.-M. Bonnin, "Dynamic reallocation of sla parameters in passive optical network based on clustering analysis", in *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, IEEE, 2018, pp. 1–8.

[3] S. Rampfl, "Network simulation and its limitations", in *Proceeding zum seminar future internet (FI), Innovative Internet Technologien und Mobilkommunikation (IITM) und autonomous communication networks (ACN)*, Citeseer, vol. 57, 2013.

[4] C. Santos, B. Shariati, R. Emmerich, C. Schmidt-Langhorst, C. Schubert, and J. K. Fischer, "Automated dataset generation for qot estimation in coherent optical communication systems", in *European Conference and Exhibition on Optical Communication*, Optica Publishing Group, 2022, Tu2–4.

[5] J. Lu, Q. Fan, G. Zhou, *et al.*, "Automated training dataset collection system design for machine learning application in optical networks: An example of quality of transmission estimation", *Journal of Optical Communications and Networking*, vol. 13, no. 11, pp. 289–300, 2021.

[6] A. M. Alam, M. M. Farhad, W. Al-Qwider, *et al.*, "A physical testbed and open dataset for passive sensing and wireless communication spectrum coexistence", *IEEE Access*, vol. 12, pp. 131 522–131 540, 2024. DOI: 10.1109/ACCESS.2024.3453774.

[7] L. Lingleslogia, *PON Dataset*, https://github.com/lingleslogia/PONData, Accessed: 2025-04-07, 2025.

[8] M. Feknous, T. Houdoin, B. Le Guyader, J. De Biasio, A. Gravey, and J. A. T. Gijón, "Internet traffic analysis: A case study from two major European operators", in *2014 IEEE symposium on computers and communications (ISCC)*, IEEE, 2014, pp. 1–7.

[9] Y. Zhang, *Residential network traffic and user behavior analysis*, 2010.

[10] M. Kihl, P. Ödling, C. Lagerstedt, and A. Aurelius, "Traffic analysis and characterization of Internet user behavior", in *International Congress on Ultra Modern Telecommunications and Control Systems*, IEEE, 2010, pp. 224–231.

[11] T. pandas development team, *Pandas-dev/pandas: Pandas*, 2020. DOI: 10.5281/zenodo.3509134. [Online]. Available: https://doi.org/10.5281/zenodo.3509134.