Integrated management of 10G-PON network element using NETCONF and OpenFlow

André Berti Sassi*, Manuel Santos Gavidia*, Eder Leão Fernandes* and Marcelo Ribeiro Nascimento*
*Converged Networks Corporate Department
CPqD Research and Development Center in Telecommunications, Campinas, Brazil
Email: {asassi,manuelg,ederlf,marcelon}@cpqd.com.br

Abstract—**This paper describes an integrated management architecture and solution for controlling network elements composed of controller, optional aggregation switch and a variable number of optical line terminals (OLTs), including multiple OLTs based on different technologies and presenting distinct features. The YANG language was used to model configuration and state data manipulated by the Network Configuration Protocol (NETCONF). The system allows the configuration of OpenFlow switches, providing greater protocol flexibility and forwarding algorithms to the aggregation network. The architecture has been implemented and tested using a real Gigabit-capable Passive Optical Network (GPON), a stacked GPON and switch equipment in a variety of scenarios.**

I. Introduction

Passive Optical Networks (PONs) have been widely deployed due to their characteristics of physical media sharing and huge amount of bandwidth offered to users. Given the increasing need of bandwidth required by residential or commercial users, it is a goal for next-generation networks to offer yet more bandwidth preserving investments in legacy PONs fiber infrastructure [1] [2].

In this direction, ITU-T has developed recommendations for XG-PON technology [1], delivering up to 10 Gbps in the downstream, 2.5 Gbps in the upstream and coexisting with legacy Gigabit-capable Passive Optical Network (GPON). However, some studies implement coarse wavelength division multiplexing (CWDM) PON overlay, or PON stacking, using a single Optical Distribution Network (ODN) to carry multiple independent colored PON systems [2] [3] [4]. Penze [5] shows an experimental proof of concept of a stacked GPON system using four wavelengths in the standard GPON downstream and upstream frequency bands, multiplexed in a single ODN.

The access network using PON technology is composed of an Optical Line Terminal (OLT) usually located at a local carrier's central office and of Optical Network Terminals (ONTs) located at users premises – residential or commercial – or Optical Network Units (ONUs) installed in cabinets located close to users. The ODN is composed of optical fibers and passive optical splitters, so that the optical signal from the OLT is passively distributed to ONTs and ONUs. The aggregation network is composed of one or more tiers of network elements between the access network and the Broadband Network Gateway (BNG) aggregating user traffic from multiple OLTs.

However, the traditional network architecture is not suited for the dynamic traffic engineering demands of today's competitive carrier environments and data centers. The new paradigm of Software Defined Networks (SDNs) is emerging to enable carriers and enterprises to build highly scalable, flexible networks that readily adapt to changing business needs [6]. SDN is considered to be a solution that will unify the control plane in converged access and aggregation networks, mostly because it offers a flow for defining detailed rules for handling a variety of traffic classes and sources [7]. OpenFlow is the first standard interface designed specifically for SDN, providing high-performance and granular traffic control across multiple vendors' network devices [8].

The objective of this work is to describe a system architecture to control network elements (NE) comprising one or more OLTs and an OpenFlow-based aggregation switch as well as its implementation and test. In Section II, a description of the network targeted by the architecture is presented. In Section III the software architecture to control the network element is detailed. In Section IV, the details on how the system configuration was modeled are presented. Finally, in Section V the implementation and validation of the architecture, as well as the test bed used and other important implementation decisions, are presented.

II. System Description

The proposed platform architecture is a system with a variable number of 10G-PON and GPON OLTs, a top-of-rack OpenFlow switch and a controller unit. The controller connects to the switch and OLTs providing the system's operator or a Network Management System (NMS) an integrated management interface to control all equipment, i.e., it must allow the operator or the NMS to manage any OLT or ONT from a single management connection. The controller must provide a unified view of resources used by more than one equipment. Each OLT is connected to the switch through one or more high speed connections concentrating the user traffic towards the network aggregation interface.

The system must support a variable number of OLTs and ONUs with minor impact to system performance, considering both data and control planes. In the data plane, the system performance depends on the capacity of physical components, and the data transmission rates are one of the main performance metrics. Thus, the interconnection must be designed to support maximum rates on unit ports. In the control plane, the system performance depends mainly on the amount of information to be processed or stored. This amount is proportional to the number of equipment in the system. The system should be designed to process information efficiently and, whenever possible, regardless of the number of OLTs or ONUs used.

978-3-901882-48-7 © 2012 IFIP

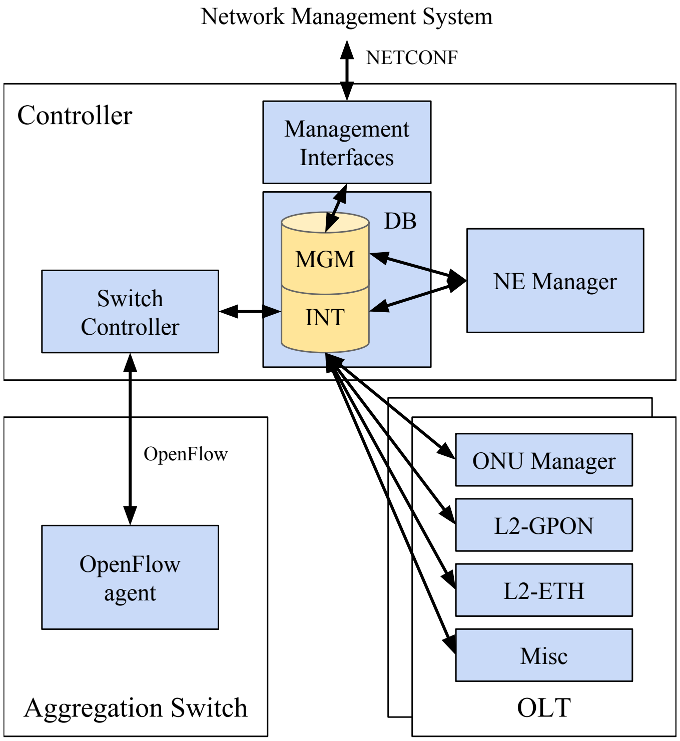

Fig. 1. Software architecture.

In order to allow a seamless transition from a GPON technology to stacked GPON or XG-PON, the system must be able to simultaneously control equipment using both technologies. The system design must also support new deployment scenarios, different technologies, and failure isolation, as well as being prepared for future integration with high availability frameworks.

As a remote programmable device, the control software of the top-of-rack switch will run on the controller, from where it must be able to send instructions to the equipment, which will operate like a standard Layer 2 switch with Virtual Local Area Network (VLAN) and static link aggregation support. The goal is to aggregate the traffic from OLTs that goes out of the network element, and also forward the internal traffic among OLTs.

III. Management Software Architecture

The designed software architecture is depicted in Fig. 1, which shows the software components that runs in the controller, in each OLT and in the aggregation switch.

The controller has a database (DB) divided into two distinct parts: the management model and the internal model. The management model abstracts hardware and implementation details modeling the system from an operators perspective and is exposed to the operator or NMS through a set of management interfaces, while the internal model abstracts detail of the hardware, software or technology concepts and is used only by the system software.

The Network Configuration Protocol (NETCONF) defines a mechanism through which a network device can be monitored, managed and configuration data can be retrieved, uploaded and manipulated [9]. The NETCONF is used as the main interface to the system configuration and status.

The mapping between management and internal models contains most of business logic to control both the OLT and the aggregation switch. As the internal model is almost a direct representation of the underlying hardware and software components, implementing the control of each equipment component is simple, as it only requires the system to read and apply the configuration from the DB almost as is. Some exceptions must be considered, such as the fact that because the internal model is not close enough to the configuration it needs to be applied and that the controller must convert some parameters into a more detailed structure. An example is the management of ONUs using the ONU Management and Control Interface (OMCI): only the relevant OMCI-managed entities parameters were modeled. In this case, the controller must derive the set of managed entities and its attributes from the configuration and use OMCI mechanisms to control ONUs. However, no additional information from the management model or resulted from crossing data from different parts of the model is required in this process.

This mapping is the responsibility of the Network Element Manager, which transforms configuration data committed to the database in a high abstraction level (management model) into a more low-level per-equipment configuration (internal model). The result of the transformation is written back to the database, so that each equipment can read it independently and configure itself. As the Network Element Manager defines the configuration of every equipment, it is responsible for ensuring the coherence and consistency of the system configuration.

The solution for the aggregation switch control consists of using an OpenFlow protocol enabled equipment with an Open vSwitch running in it, so an application can control it externally. This application is the Switch Controller that runs in the controller and interfaces with the DB. The changes in configuration are received by the Switch Controller and translated into OpenFlow messages or Open vSwitch commands – for operations currently not supported via OpenFlow, in order to configure switch features.

The OLT software is divided into the following components: ONU Manager, L2-GPON, L2-ETH and Misc. Each of these components has well defined responsibilities and work together to control the OLT hardware and software components and all ONUs connected to the PON ports.

The ONU Manager applies the configuration from the internal model to the ONUs by translating it into a series of OMCI packets sent to the ONU to update its management information base (MIB). The ONU Manager is also responsible for performing other ONU management functions, such as MIB synchronization, alarm handling and software image upgrade. The OMCI is implemented by our custom OMCI stack, which implements the standard OMCI mechanisms and managed entities, but also supports advanced sequencing of messages and extending the MIB with new managed entities and attributes as well. The OMCI stack is designed to be lightweight and scalable, requiring very few resources from the underlying system.

The L2-GPON is responsible for handling configuration of PON links, ONU registration, activation and other PON management functions that are performed using Physical Layer Operation, Administration and Maintenance (PLOAM) messages, also needed for service activation. This component is also responsible for configuring the manipulation of packet headers during Ethernet frame encapsulation into GPON or XG-PON frames and vice-versa.

The L2-ETH is responsible for controlling Ethernet interfaces in the OLT and the traffic aggregation from all PON ports running into Ethernet ports. This component is also responsible for implementing the network protocols that should run on the OLT. For instance, the Internet Group Management Protocol (IGMP) is often used to reduce bandwidth usage of IPTV multicast video streams.

The Misc handles remaining OLT hardware components and auxiliary software features, such as the OLT software version control.

The relationship between components is also indicated in Fig. 1. The DB has a central role, implementing the configuration database and the interfaces to access the data. The NE Manager only needs to communicate with the DB to transform the system configuration. As the transformation result is written to the DB, other components can read directly from it the configuration they should apply. This way, each component only needs to inspect the configuration for a small part of the model representing the hardware component it controls.

IV. Data Modelling

YANG is a language used as a powerful approach to model data for the NETCONF protocol [10]. A YANG module defines a hierarchy of data that can be used for NETCONF-based operations, including configuration, state data, RPCs, and notifications. This allows a complete description of all data exchanged between a NETCONF client and server. The YANG hierarchical DB model organizes data into a tree-like structure and provides clear and concise descriptions of the nodes, as well as the interaction between them.

A system model with system-wide configuration data and a list of extensible units have been developed. A unit model defines data, operation and constraints for a generic network device, such as network interfaces and interconnection rules. Technology-specific models define their own unit extensions, adding new configurations, operations and rules to the generic ones. Even though each unit has a plethora of interfaces, only predefined interfaces are exposed for management, the so-called system interfaces. All other interfaces may be used for interconnection. This type of modeling supports systems with any interconnection patterns and a set of system interfaces, including a stand-alone line card.

The PON model consists of PON interfaces, ONUs and services. PON interfaces are modeled just as any other unit interface and are presented in the list of system interfaces.

The list of ONUs is used by the operator to register all ONUs in the network by assigning a name to them. The PON interface name and the ONU serial number are used to bind them with the physical ONU. Many ONU configuration parameters are available, including the list of user ports and its parameters, multicast groups, software image banks and Voice over IP (VoIP) parameters.

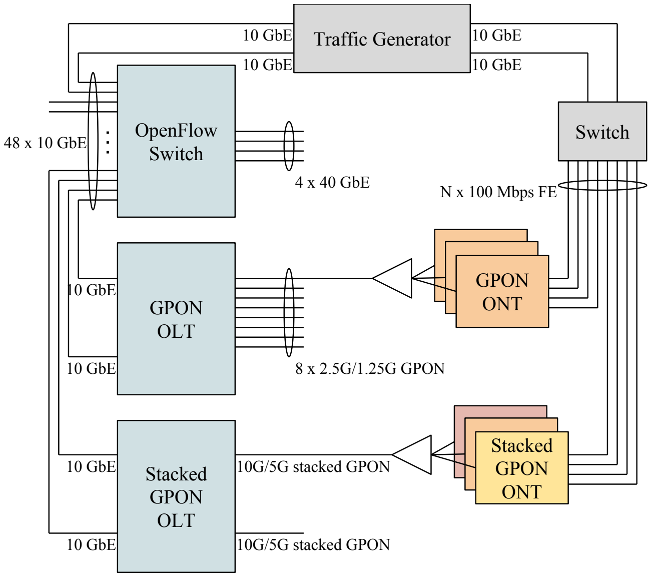

Fig. 2. Block diagram of the system used to validate and test the architecture depicting the network connections between elements.

The list of services models all provisioned services, such as data, IPTV or VoIP services. Each service defines a logical path between a Service VLAN (S-VLAN) and the user port, and has configurable parameters, such as upstream bandwidth assignment rules and downstream encryption mode. Some VLAN tagging manipulations can also be configured according to TR-156 [11] standard.

There is also a list of S-VLANs, which is a unified list of all S-VLAN provisioned by the system. The concept of S-VLAN is used to bind the user traffic to a specific set of ports in the aggregation switch. As each service specifies the S-VLAN it belongs to, only the external switch ports are added by the operator to the S-VLAN, as the system is able to add the ports connecting to the OLTs as needed. Thus, the operator is no longer responsible for configuring the ports connecting the OLTs and the switch.

V. Implementation and Testing

The proposed architecture was validated through its implementation in an specific hardware platform. A test bed was assembled using two 1U OLT units: one 10G stacked GPON OLT with two 10/5 Gbps PON ports and one GPON OLT with eight 2.5/1.25 Gbps PON ports, both with two 10 Gbps uplink Ethernet ports, one OpenFlow Ethernet switch with 48 x 10 Gbps ports and four 40 Gbps ports and a standard PC running Ubuntu 12.04 as a controller. The stacked GPON system (OLTs and ONTs) was developed by CPqD. All ONTs were triple-play ONTs designed for residential or small business users with four Fast Ethernet ports, some of them operating in bridge mode (layer 2) and others in IP router mode. The Fig. 2 shows how these pieces of equipment were connected.

In the controller, a proprietary software from Tail-f [12], ConfD, was used as the NETCONF implementation and database management system. It also provides other management interfaces, such as command-line and Web interfaces for direct human interaction, and REST and SNMP as alternative protocols for NMS, all of them rendered automatically from the YANG models.

Aiming at the ease of use of the data modeled in YANG, a plug-in to the open source pyang tool [13] was developed. Such tool generates C++ declarations equivalent to those of the model and with the additional capability to read and write data in the ConfD's database. It produces a code that can efficiently access the database and provide a stable interface regarding model changes during development.

The system integration uses a scalable software architecture regarding the number of OLTs and ONUs, as their configuration is executed by components running dedicatedly and in parallel in each OLT, and the centralized components were developed aiming at scalability regarding the size of system configuration.

The NE Manager's transformation algorithms work essentially over differences of configuration. This way, the cost of transforming to a new configuration is proportional to the size of the change.

The Switch Controller was built using the OpenFlow library *libfluid* [14], developed by CPqD and the Open vSwitch command line tool *ovs-vsctl*.

An important aspect of designing an OpenFlow learning switch is the table explosion, which occurs mainly due to the Cartesian product of the fields belonging to the forwarding rules. If the number of MAC addresses, VLAN and ports is too big, the single table of the version 1.0 protocol will not scale to the necessary flow number. To avoid this problem, a design using OpenFlow version 1.3, with multiple tables, was considered. In this scheme, table 0 matches source MAC, VLAN number and input port, while table 1 matches VLAN number and destination MAC. With this simple scheme, the number of flows is reduced because the forwarding logic is split, resulting in a better use of the flow tables. However, the switch used by the solution, though supported OpenFlow version 1.3, did not have multiple tables. Thus, the solution adopted was to use the single table approach when the switch did not support multiple tables.

Both OLT units used in the proposed system have a conventional Ethernet switch embedded on board to aggregate traffic from all PON ports. However, this removes much of the flexibility supported by the OpenFlow switch thus becoming necessary to use VLAN and MAC address learning concepts when creating flows and, inside OLT, the mapping of VLANs and MAC addresses to GEM Port-IDs. The latter mapping is well defined by TR-156 standards, but is not very suitable for integrating GPON into an SDN [15]. The mapping could be avoided by removing the OLT internal switch.

VI. Conclusion

Modeling management data separately from internal data has made decoupling device-specific configuration data from system management data more flexible, and facilitated the incremental software development. The architecture enabled seamless management of multiple units, their integration into the system, and the addition of new unit types, and provided support for a variable number of OLTs. Using the YANG language to model configuration data, a proprietary software tool (Tail-f's ConfD), allowed the development of all northbound interfaces without extra effort.

The implemented system using the proposed hardware platform and network infrastructure was validated and tested, proving to be a feasible solution that meets all architecture requirements.

As a future work, other forms of modeling the integration between XG-PON OLTs – or other similar broadband access technologies – and an OpenFlow switch may be developed and tested to verify system flexibility. Also, different access technologies can be integrated into the unified management, including the control of equipment located at the ONU, such as LTE-base stations in a scenario where 10G-PON access is used as backhaul.

Acknowledgment

The authors would like to thank FINEP and FUNTTEL for their financial support of this work.

References

[1] *10-Gigabit-capable Passive Optical Network (XG-PON): General Requirements*, ITU-T Rec. G.987.1, 2010.

[2] E. Trojer, S. Dahlfort, D. Hood, and H. Mickelsson, "Current and next-generation PONs: A technical overview of present and future PON technology," *Ericsson Review*, no. 2, pp. 64–69, 2008.

[3] T. Pfeiffer, "Enhancing PON capabilities using the wavelength domain," in *Join ITU-T/IEEE Workshop on Next Generation Optical Access Systems*, Geneva, Switzerland, 2008.

[4] P. Solina, "European PON deployments: A carrier view of the technical, regulatory, and economic challenges," in *Proc. European Congress in Optical Communication (ECOC 2010)*, Torino, Italy, 2010.

[5] R. S. Penze, J. B. Rosolem, and R. B. Filho, "Upgrading and extending PON by using in-band WDM overlay," *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, vol. 11, pp. 15–25, Jun. 2012.

[6] "Software-defined networking: The new norm for networks," White Paper, Open Networking Foundation, Apr. 2012.

[7] H. Woesner and D. Fritzsche, "SDN and OpenFlow for converged access/aggregation networks," in *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, Anaheim, CA, Mar. 2013, pp. 1–3.

[8] McKeown *et al.*, "Openflow: Enabling innovation in campus networks," White Paper, OpenFlow Consortium, Mar. 2008.

[9] "Network configuration protocol (NETCONF)," RFC 6241, IETF, Jun. 2011.

[10] "YANG – a data modeling language for the network configuration protocol (NETCONF)," RFC 6020, IETF, Oct. 2010.

[11] "Using GPON access in the context of TR-101," The Broadband Forum (BBF), Fremont, CA, Tech. Rep. TR-156, Dec. 2008.

[12] Tail-f. (2013) On-device management. [Online]. Available: http://www.tail-f.com/on-device-configuration-management/

[13] (2013) pyang: An extensible YANG validator and converter in python. [Online]. Available: https://code.google.com/p/pyang/

[14] CPqD. (2014) libfluid documentation. [Online]. Available: http://opennetworkingfoundation.github.io/libfluid/

[15] P. Parol and M. Pawlowski, "Towards networks of the future: SDN paradigm introduction to PON networking for business applications," in *Proc. Federated Conference on Computer Science and Information Systems*, Krakow, Poland, Sep. 2013, pp. 829–836.