An XG-PON Module for the NS-3 Network Simulator

Xiuchao Wu, Kenneth N. Brown, Cormac J. Sreenan
Department of Computer Science
University College Cork, Ireland
{xw2, k.brown, cjs}@cs.ucc.ie

Pedro Alvarez, Marco Ruffini, Nicola Marchetti, David Payne, Linda Doyle
CTVR / The Telecommunications Research Centre, Trinity College Dublin, Ireland
{pinheirp, marco.ruffini, marchetn, ledoyle}@tcd.ie,
david.b.payne@btinternet.com

ABSTRACT

10-Gigabit-capable Passive Optical Network (XG-PON), one of the latest standards of optical access networks, is regarded as one of the key technologies for future Internet access networks. In this paper, we propose and discuss the design and implementation of an XG-PON module for the NS-3 network simulator. The aim is to provide a standards-compliant, configurable, and extensible module that can simulate XG-PON with reasonable speed and can support a wide range of research topics. These include analysing and improving the performance of XG-PON, studying the interactions between XG-PON and the upper-layer protocols, and investigating its integration with various wireless networks.

Categories and Subject Descriptors

I.6.5 [**Simulation and Modeling**]: Model Development—*Modeling Methodologies*

General Terms

Algorithm, Design, Performance

Keywords

Passive Optical Network, XG-PON, Internet Access Network, Network Simulator 3

1. INTRODUCTION

During the last few decades, we have witnessed the huge success of the Internet, which has changed our daily life significantly and has become one of the main economy engines. In these years, the infrastructure of the Internet kept evolving to provide better performance, and optical communication is one of its driving forces.

Transmission links in the network core are already based on optical fiber technology, which provides huge amount of bandwidth through the matured DWDM (Dense Wavelength Division Multiplexing) technology. More recently, optical fibers have also found their application in access networks to provide high speed Internet access to end users. FTTx (Fiber To The Home/Building/Curb, etc.) networks based on Passive Optical Network (PON) technologies, such as Gigabit-capable PON (GPON) standardized by the Full Service Access Network (FSAN) group of the International Telecommunications Union (ITU) [5] and Ethernet PON (EPON) standardized by the Ethernet in the First Mile (EFM) task force of the Institute of Electrical and Electronics Engineers (IEEE) [3], have been widely deployed in many countries such as the US, Korea and Japan.

10-Gigabit-capable Passive Optical Network (XG-PON) is a new standard released by the FSAN that improves G-PON, by increasing the default downstream data rate to 10 Gb/s, while increasing the upstream data rate to 2.5 or 10 Gb/s. Also, the maximum number of users per wavelength is increased from 64 to 256, and amendments are being defined for extending the physical reach up to 60 Km.

Since XG-PON could pave the way for many bandwidth-intensive applications (IPTV, Video On Demand, Video Conference, etc.), it is very important to study the performance issues arising with the deployment of XG-PON. For instance, it is valuable to study the impacts on the performance of XG-PON, when the propagation delay is much longer than that of the current PON networks [22]. It is also important to investigate the interactions between XG-PON and the upper-layer protocols (TCP [20], etc.) for improving user experience [14]. In addition, XG-PON has been proposed for Fiber To The Cell, in which XG-PON acts as the backhaul for multiple base stations of a cellular network [8]. Under this scenario, it is also very valuable to study its integration with various wireless networks (LTE [1], WiMAX [4], etc.) for providing high speed mobile Internet access.

Considering that XG-PON is still in its early stage, the above research topics should be first studied through simulation. In this paper, we present an XG-PON module for NS-3 [6], a state of the art open-source network simulator. Our XG-PON module is based on a series of G.987 Recommendations from the FSAN group of ITU. These recommendations mainly define the specifications of Physical Media Dependent (PMD) and Transmission Convergence (TC) layers of XG-PON. To study the above research topics with reasonable simulation speed, the optical distribution network and the operations of physical layer are simplified significantly. This XG-PON module focuses on the issues of TC layer, such as frame structure, resource allocation, Quality of Service (QoS) management, and Dynamic Bandwidth Assignment (DBA) algorithms for the upstream wavelength. During the

design and implementation of this module, we have also paid a lot of attention on its extensibility and configurability.

This XG-PON module is built completely in C++ with 55 classes and approximately 15,000 lines of code. These code are under the GNU General Public License and will be released through the websites of our institutions (CTVR[1] and MISL[2]) in the near future. To the best of our knowledge, it is the first XG-PON module for NS-3. We believe that this work is a significant contribution to the scientific community as it allows to simulate XG-PON, the next generation optical access network, and study the performance issues that arise with the deployment of XG-PON.

This paper is organized as follows. Section 2 briefly introduces PON, NS-3, and related works. The details of XG-PON are then presented in section 3. Section 4 presents the design details of an XG-PON module for NS-3, followed with its implementation status. The important trade-offs made by us in terms of simulation acuracy and simulation speed are also discussed throughout. Finally, section 5 concludes this paper with several directions for future work.

2. BACKGROUND AND RELATED WORK

2.1 Passive Optical Network (PON)

Compared to copper, optical fiber can provide higher bandwidth over a longer distance. However, the deployment of optical fiber in access networks is severely hindered by the cost issue. In fact, optical fiber was seriously considered for access networks only after the emergence of PON.

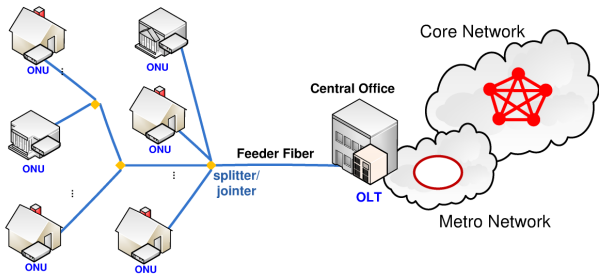

Figure 1: An Illustration of PON

As shown in Figure 1, PON is a point-to-multipoint fiber network and there are three kinds of equipment: the OLT (Optical Line Terminal) in central office, ONUs (Optical Network Unit) in/near customer premise, and passive optical splitters/jointers in the middle. Through splitter/jointer, OLT and the feeder fiber are shared by multiple users. Compared with the point to point architecture, PON can significantly reduce the amount of required optical fibers and the central office equipments. Since the passive optical splitters/jointers do not need power supply, the cost of deployment, maintenance and operation can also be reduced. Thus, PON could reduce both capital expenditure and operational expenditure significantly.

In a classical TDMA (Time Division Multiple Access) based PON network, downstream traffic is broadcast by the OLT to all ONUs that share the same optical fiber and encryption is used to prevent eavesdropping. Upstream traffic from ONUs is interleaved by OLT for using the optical fiber in a TDMA-like manner. Since ONUs normally have different distances to the OLT, the data bursts from these ONUs must be scheduled carefully for providing a collision-free and efficient upstream data communication. To accommodate the dynamics in bandwidth demands from users and exploit the gain of statistical multiplexing, dynamic bandwidth assignment (DBA) is normally used for managing the upstream bandwidth. More specifically, ONUs will report their buffer occupancy to OLT, which will then allocate the upstream bandwidth to ONUs based on their bandwidth demands and their Service Level Agreement (SLA).

Some standards have been developed for PONs by both EFM of IEEE (EPON) and FSAN of ITU-T (GPON). EPON is designed for carrying Ethernet frames and GPON can carry various traffics through encapsulation, such as Ethernet frames and ATM cells. Although EPON and GPON have different frame structures, they share the same network architecture and data communication follows the same principles described above. One important difference between EPON and GPON is that GPON is well standardized for QoS management. Hence, GPON can provide full service with the same network and it is highly preferred by ISPs. XG-PON is the new standard developed by FSAN based on GPON. Its details will be introduced in section 3.

2.2 NS-3 Network Simulator

NS-3 [6] is a state of the art open-source network simulator. Based on many lessons from the well-known NS-2 simulator [16], NS-3 is written from the scratch and it is a completely new network simulator. NS-3 has many attractive features, such as high emphasis on conformance to real networks, good support for testbeds, a novel attribute system for configuring simulation parameters, automatic memory management, and a configurable tracing system [13]. It has also been reported that NS-3 performs much better than other simulators in terms of simulation speed and memory overhead [23].

The first release of NS-3 was made in June 2008 with support for a number of modules including CSMA, Point-to-Point, WiFi (IEEE 802.11), TCP, UDP and IPv4. In the last few years, many new modules have been developed and added into NS-3, such as WiMAX module from Inria [11] and LTE module from CTTC [19].

2.3 Related Work

Although simulation has been used to study PON, the existing work cannot be used directly or extended easily to study the performance issues arising with the deployment of XG-PON.

In [22], the authors developed their own simulator to study dynamic bandwidth assignment (DBA) algorithms when the physical reach is much longer than the current PON networks. This simulator has limited functions and there is no Internet protocol stack, wich is needed to study many research topics.

EPON and GPON had also been studied with OPNET [2] and several models have been implemented by different authors [10][18]. However, these EPON/GPON models are not available to the public. Furthermore, OPNET simulates too many details (CPU of a router, etc.) and the simulation speed is slow even when the simulated network bandwidth is lower than 1 Gb/s. Since OPNET is not an open-source simulator, we cannot change its core to simulate a 10 Gb/s XG-PON network with a reasonable simulation speed.

[1] www.ctvr.ie
[2] http://www.ucc.ie/en/misl/

In addition, one simple EPON module has been developed for OMNeT++ [9] and the code is available to the public. Since there are a lot of between EPON and XG-PON, the code of this EPON module may not be very helpful to implement one XG-PON module for OMNeT++. Considering the good points of NS-3 discussed above, it should be better to develop one XG-PON module from the scratch for the NS-3 network simulator.

Hence, an XG-PON module is designed and implemented in this paper for NS-3. With such an XG-PON module, we can simulate XG-PON and study its own performance. With the more realistic Internet protocol stack of NS-3, we can study the performance experienced by users/applications in XG-PON networks. With the existing NS-3 modules for various wireless networks (WiFi, WiMAX, LTE, etc.), we can also study the integration between XG-PON and wireless networks, which is the trend of the future Internet access networks. In summary, with this XG-PON module, NS-3 will become a good platform for studying the next generation Internet access networks composed by XG-PON and wireless networks.

Not only XG-PON, we can also extend this XG-PON module to study Long-Reach PON (LR-PON), an evolution of XG-PON with a larger number of users, symmetric data rate (10 Gb/s in both upstream and downstream), and longer reach (100+ km) [17][21]. The aim of our LR-PON research group is to initially build LR-PON from the XG-PON standard, while identifying the required modifications and improvements.

3. XG-PON DETAILS

The XG-PON standard has many similarities with GPON, such as its TDMA scheme used to share the medium, the mechanism to provide QoS, and the DBA scheme used for the upstream wavelength. However, some changes are required in order to support a larger number of users, higher data rate, and extended physical reach. In this section, we will present the details of XG-PON.

3.1 Overview of XG-PON

A series of recommendations has been released by FSAN of ITU-T for XG-PON. ITU-T G.987 explains several important concepts of XG-PON and ITU-T G.987.1 presents the general requirements, such as network architecture, migration and coexistence with GPON, services to be supported, hardware specifications, protocol stack, etc. ITU-T G.987.2 focuses on issues of the physical media dependent (PMD) layer, such as the used wavelength and the supported data rates. ITU-T G.987.3 presents the details of transmission convergence (TC) layer. Not only the protocols for data communication, it also covers QoS management and Dynamic Bandwidth Assignment (DBA) scheme for the upstream wavelength. Another related recommendation is ITU-T G.988, which specifies ONU management and control interface (OMCI) for both GPON and XG-PON.

3.2 Network Architecture

XG-PON has been proposed for various deployment scenarios to serve different customers, such as residential, business, and cell site. To serve these customers, XG-PON lists the services to be provided, such as Telephony, high speed Internet access, mobile back-haul, etc. XG-PON also introduces many ONU variants that provide different functions and interfaces. In summary, XG-PON has been well standardized for providing full services to various users with one optical network.

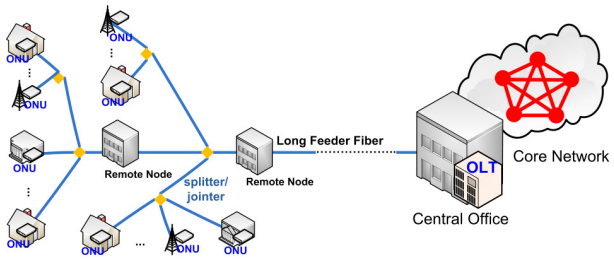

Figure 2: XG-PON Network Architecture

As for optical distribution network, XG-PON can be deployed as a classical PON, but mechanisms to extend its reach up to 60 km are currently being defined. As illustrated in Figure 2, to support this longer reach, active component (Reach Extender) can be applied in remote nodes and one XG-PON can be composed of multiple passive segments connected through REs. These REs can be optical amplifiers or optical-electrical-optical regenerators that could fulfill the necessary optical link budget.

3.3 PMD Layer

There are two flavours of XG-PONs based on the upstream line rate: XG-PON1, featuring a 2.5 Gb/s upstream path, and XG-PON2, featuring a 10 Gb/s one. The downstream line rate is 10 Gb/s in both XG-PON1 and XG-PON2. ITU-T G.987.2 focuses on the PMD layer for XG-PON1. As for XG-PON2, it hasn't been standardized yet.

In XG-PON1, the used wavelengths are 1575-1580 nm (downstream) and 1260-1280 nm (upstream). The exact downstream line rate is 9.95328 Gb/s and the upstream one is 2.48832 Gb/s. ITU-T G.987.2 also specifies its line coding and the requirements for hardwares, such as optical fiber, transmitter/receiver, etc.

3.4 Transmission Convergence Layer

The XG-PON Transmission Convergence (XGTC) layer is where the Medium Access Control (MAC) protocol of XG-PON are defined.

To carry traffic between the OLT and the ONUs, the XGTC layer maintains logical connections between these two entities, designated XGEM Ports. Each connection is identified by a unique XGEM Port-Id, which enables to send a packet to the correct ONU and associate a connection to a certain Quality of Service (QoS) agreement.

To reduce the overhead of the DBA scheme, upstream bandwidth is allocated to groups of connections belonging to a single ONU. These groups are designated as Transmission Containers (T-CONT) and each group/T-CONT is identified by a unique identifier, the Alloc-Id.

XGTC comprises three sublayers: service adaptation, framing, and PHY adaptation, from top to bottom. Following these sublayers, XGTC is introduced below.

3.4.1 Service Adaptation Sublayer

The service adaptation sublayer is responsible to adapt the upper layer traffic to the transmission mechanisms of XG-PON. It will do this by mapping upper layer traffic to the corresponding connections, encapsulating/decapsulating

data, segmenting/reassembling SDUs when necessary and inserting padding when there is not enough data to fill an XGTC frame. If needed, it is also this sublayer's responsibility to encrypt/decrypt SDUs.

To map upper layer data to and from the connections of XGTC layer, the OLT will maintain all connections and the ONU will maintain the connections that belong to itself.

When the upper layer has something to transmit, it is also the service adaptation sublayer's responsibility to select the connections to be served according to their QoS parameters. When a connection is scheduled to be served, the service adaptation sublayer will then get data from its queue and insert an XGEM header to create an XGEM frame. The XGEM header will contain an XGEM Port-Id and some other information related to segmentation, padding, etc.

When receiving an XGEM frame, the service adaptation sublayer will get the XGEM Port-Id from the XGEM header. If the corresponding connection exists in the connections maintained by the OLT/ONU, this sublayer will carry out reassembly (if necessary) and pass the data to upper layer. Otherwise, this XGEM frame will be discarded.

3.4.2 *Framing Sublayer*

In XG-PON, the OLT will send downstream XGTC frames every 125 μs, to broadcast traffic to all ONUs. In the upstream, each ONU sends a variable length XGTC burst to the OLT for its upstream traffic. The length and start time of these upstream bursts are determined by the OLT through a DBA algorithm.

The framing sublayer is responsible to generate and parse these XGTC frames/bursts. When generating one downstream XGTC frame, the framing sublayer gets XGEM frames from service adaptation sublayer and joins them together into an XGTC payload. To create an upstream XGTC burst, the framing sublayer may create multiple XGTC payloads, where each payload will carry XGEM frames from a single T-CONT. When parsing an XGTC frame/burst, the framing sublayer will send its payloads to the service adaptation sublayer for further processing.

In the header of the upstream XGTC burst generated by an ONU, there might be queue occupancy reports for the T-CONTs of this ONU. For each downstream XGTC frame, its header contains one BW_{map}, which instructs ONUs to share the upstream wavelength in a TDMA-like manner. More specifically, BW_{map} specifies the size of bandwidth allocations for T-CONTs, the used burst profile (the preamble length, etc.), and the time to start to transmit. Since the OLT-ONU physical distance could be quite different for ONUs, each ONU should adjust the start time for avoiding collision in the upstream direction.

In the header of one upstream XGTC burst, the ONU can send one PLOAM (Physical Layer Operations, Administration and Maintenance) message to the OLT. As for one downstream XGTC frame, the OLT can send multiple PLOAM messages to multiple ONUs. Through exchanging PLOAM messages, many XGTC functions (key management, etc.) can be fulfilled.

3.4.3 *PHY Adaptation Sublayer*

PHY adaptation sublayer interacts with PMD layer directly. Its main functions are forward error correction (FEC), scrambling, and frame delineation through a Physical Synchronization Block (PSB).

In the downstream, the PHY adaptation sublayer will get an XGTC frame to create a PHY frame. These PHY frames are sent continuously every 125 μs. In the upstream, the PHY adaptation sublayer will get the XGTC burst and create a PHY burst. These PHY bursts have variable length due to the variable-length XGTC bursts. In the PHY burst, the PSB is determined by a burst profile selected by the OLT (through the BW_{map}) among the burst profiles, that are configured through PLOAM messages.

3.5 Scheduling and DBA

To decide the data to be transmitted in a downstream XGTC frame, a downstream scheduler is used by the OLT. Based on QoS parameters and service history of these downstream connections, the downstream scheduler will decide the connections to be served and the amount of data to be transmitted for each of them.

As for the upstream scheduling, the OLT uses a DBA algorithm to allocate the upstream bandwidth to T-CONTs. The DBA algorithm makes decisions based on queue occupancy reports, QoS parameters, and service history of these T-CONTs. The DBA algorithm needs to select the T-CONTs to be served, reserve a short gap time between the consecutive XGTC bursts for tolerating clock synchronization errors, determine the size of each bandwidth allocation, and calculate the start time for each bandwidth allocation. These decisions are broadcast to ONUs through BW_{map}. Since the upstream bandwidth is allocated to T-CONTs and each T-CONT may have multiple upstream connections, the ONU also needs one upstream scheduler to determine the upstream connections to be served during one transmission opportunity assigned to one T-CONT.

These scheduling algorithms, especially the DBA algorithm, are very important to network performance and QoS management. To allow competition and encourage research, these algorithms were intentionally left out of the standard. Indeed, it has been a very hot topic to study DBA algorithms for EPON and GPON [12][15][22]. Hence, there should be many research opportunities for XG-PON too.

4. THE XG-PON MODULE FOR NS-3

This section describes in detail the XG-PON module we have developed for NS-3. Our aim is to provide a standard compliant, configurable, and extensible module that can simulate XG-PON with reasonable speed and can support a wide range of research topics.

4.1 Overview of the XG-PON Module

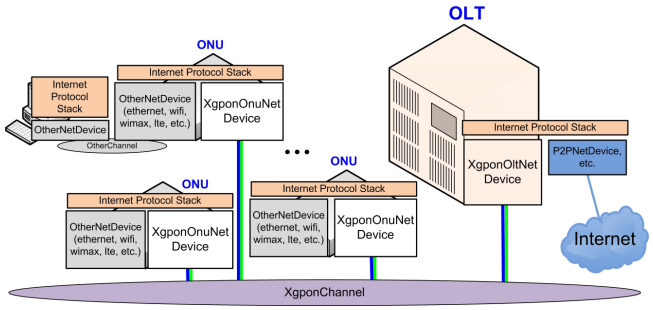

Figure 3: The Reference XG-PON Simulation

Figure 3 illustrates a typical simulation that uses this module and NS-3 to study the performance issues arising

with XG-PON. The OLT is simulated as a node that has one *XgponOltNetDevice* and another network device, such as PointToPointNetDevice, to connect to an external network. The ONU is simulated as a node with one *XgponOnuNetDevice* and another network device (Ethernet, WiFi, WiMAX, LTE, etc.) for connecting user equipments to the ONU. Thanks to NS-3, network devices of a node can be configured and we can study different deployment scenarios of XG-PON easily. Although XG-PON is proposed to carry layer-2 frames, our XG-PON module interacts directly with the IP layer and IP packets are the SDUs. This is reasonable since we focus on FTTx networks connected to the Internet.

The OLT and ONUs are attached to *XgponChannel* that simulates the optical distribution network (ODN) of XG-PON. As illustrated in Figure 2, this ODN is a quite complex tree composed by optical fibers, splitters/jointers, and REs. To produce trustworthy simulation results, it is highly desirable to simulate all details. However, XG-PON is a high speed network with a very complex standard. In this module, many aspects have been simplified for reducing the development workload and speeding up the simulation speed.

Sepcifically, our XgponChannel just simulates d_{max} (the maximal propagation delay of ODN) that can be configured through the attribute system of NS-3. For a downstream PHY frame from the OLT, XgponChannel will postpone it for d_{max} and pass this frame to all ONUs at the same time. As for an upstream PHY burst, it will also be postponed for d_{max}, but XgponChannel will pass it to the OLT only.

This means that the propagation of optical signals and the differential delays of the ONUs are not simulated by the XgponChannel. This design is reasonable since the targeted research topics are related with MAC and upper layers. Through these simplifications, simulation speed can also be significantly improved. Otherwise, many events must be scheduled to pass a downstream XGTC frame to ONUs at different times. It is also very CPU-intensive to calculate the optical signal strength for each downstream frame when it arrives to each ONU.

In the following subsection, we will present how the XG-PON protocol stack is simulated. More specifically, we will identify its functional blocks, followed by the design and implementation details.

4.2 XG-PON Functional Blocks

Since we are interested in the performance of a running network, the functional blocks of XG-PON are identified below through explaining the data transmission paths in both downstream and upstream directions.

4.2.1 *Downstream Traffic on OLT Side*

As shown in Figure 4, when one SDU is received from the upper layer, it will first be classified (based on the destination IP address) and put into the corresponding queues. Thus, there must be one algorithm for mapping the IP address to a XGEM Port-Id.

Since the OLT needs to broadcast the downstream XGTC frames every 125 μs, it will periodically ask the OLT's *Framing Engine* to generate a XGTC frame. This engine will first generate an XGTC header since the available space for data in the frame depends on the size of the XGTC header.

For the payload of a downstream XGTC frame, the Framing Engine resorts to the *XGEM Engine* to get an XGTC payload. This payload is comprised of concatenated XGEM

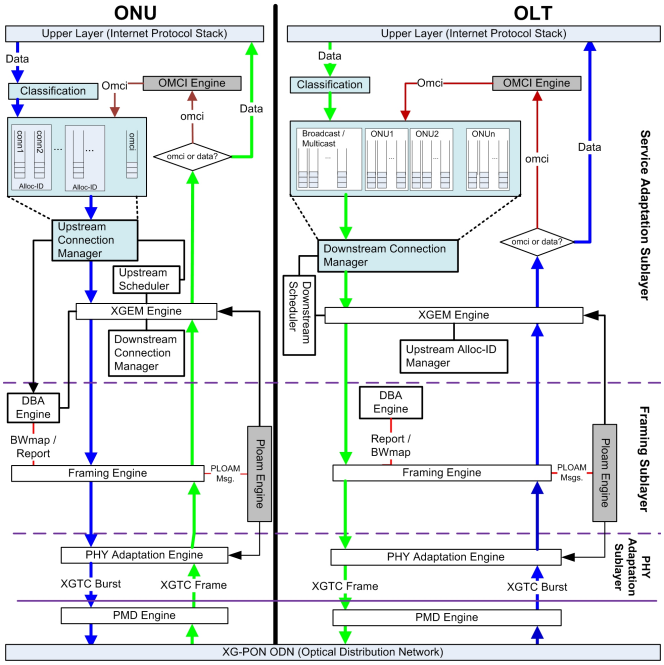

Figure 4: Function Block Diagram of XG-PON

frames that occupy all the available space. As for the SDUs to be encapsulated and transmitted, the XGEM Engine lets the *Downstream Scheduler* decide the connections to be served. This scheduler makes decisions based on *Downstream Connection Manager* which knows queue length, QoS parameters, and service history of each downstream connection. When carrying out encapsulation, segmentation will be carried out by XGEM Engine if one SDU is too long for the current transmission opportunity. XGEM Engine is also responsible to encrypt these SDUs to avoid eavesdropping. The keys used for data encryption are negotiated through PLOAM messages and are maintained by *Ploam Engine*.

To construct the XGTC header of the frame, the *DBA Engine* is used to generate BW_{map} that tells ONUs how to share the upstream wavelength. DBA Engine makes decisions based on queue occupancy reports, QoS parameters, and service history of T-CONTs. As for the PLOAM messages in the header, they are generated by Ploam Engine.

The downstream frame is sent to the ODN after passing through *PHY Adaptation Engine* and *PMD Engine*.

4.2.2 *Downstream Traffic on ONU Side*

When a downstream PHY frame arrives to one ONU, it will pass through PMD Engine and PHY Adaptation Engine which will remove the physical-layer overhead. The Framing Engine is then responsible to parse the resulting downstream XGTC frame.

The PLOAM messages from the XGTC header will be given to the Ploam Engine, which will process the messages related with this ONU. The DBA Engine is responsible to process BW_{map} in the header, i.e., schedule its upstream XGTC bursts if required by this BW_{map}.

As for the payload, the XGEM frames are passed to XGEM Engine. Based on the list of its connections maintained by Downstream Connection Manager, the XGEM frames for this ONU are first extracted. XGEM Engine then carries

out decapsulation, decryption, and reassembly (if needed)[3]. The received SDUs are then sent to the upper layer.

4.2.3 Upstream Traffic on ONU Side

As illustrated in Figure 4, when a IP packet is received at the ONU, it is also first classified (based on the source IP address) and put into the corresponding queues that are organized by *Upstream Connection Manager*.

When it is the time to transmit one upstream XGTC burst (scheduled by the DBA Engine based on BW_{map} from the OLT), the Framing Engine is resorted to produce the XGTC burst. To do this, the Framing Engine asks the XGEM Engine to get an array of XGTC payloads. Each of these payloads is a concatenation of XGEM frames belonging to one T-CONT scheduled in the BW_{map}. To decide the SDUs to be encapsulated, the *Upstream Scheduler* is also needed since the upstream bandwidth is allocated to T-CONT and multiple upstream connections might be mapped to the same T-CONT. This scheduler makes decisions based on the amount of bandwidth allocated to one T-CONT, queue length, QoS parameters, and service history of this T-CONT's upstream connections.

If required by the OLT, Framing Engine at ONU will resort DBA Engine at ONU to generate queue occupancy report for the corresponding T-CONT. This report is deduced by Upstream Connection Manager based on the upstream connections of this T-CONT. For various purposes, one PLOAM message may also be generated by Ploam Engine and be put into the header of this XGTC burst.

The upstream XGTC burst is then passed to PHY Adaptation Engine with the burst profile to be used. After going through PMD Engine, this burst is then sent to the ODN.

4.2.4 Upstream Traffic on OLT Side

When the OLT receives one upstream XGTC burst, this burst first passes through PMD Engine and PHY Adaptation Engine. The Framing Engine at OLT is then responsible to parse the header and the payloads of this burst. The potential queue occupancy report will be sent to DBA Engine and the PLOAM messages are sent to the Ploam Engine. As for the XGTC payloads, they are sent to XGEM Engine for decapsulation and reassembly (if needed). Hence, one *Upstream Alloc-ID Manager* is needed to hold the potential segments for reassembly.

As illustrated in Figure 4, both OLT and ONU should have one *OMCI Engine* for exchanging OMCI messages.

4.3 The Design and Implementation Details

Figure 5 shows the main classes of this XG-PON module. Following this class diagram, the design and implementation details of this module are presented below.

4.3.1 Channel and Network Devices

PonChannel and PonNetDevice are the base classes for a general PON network. Through developing different subclasses, we can simulate other PON technology (10G-EPON [7], etc.) and compare with XG-PON. PonChannel is inherited from Channel of NS-3 and is used to simulate the optical distribution network (ODN). It has implemented the functions for managing network devices of the OLT and ONUs attached to this ODN. PonNetDevice is inherited from NetDevice of NS-3 and is responsible to communicate with upper layers and PonChannel.

XgponChannel is the subclass of PonChannel for XG-PON and its implementation has been discussed in 4.1. As a subclass of PonNetDevice, *XgponNetDevice* is used to represent a network device attached to XgponChannel. It also acts as the container for various engines that implement the protocol stack of XG-PON. XgponOltNetDevice and XgponOnuNetDevice are its subclasses for the OLT and ONU.

4.3.2 Frame Structure

PonFrame represents the frame transmitted over the ODN of a PON and it just provides the interfaces for serialization, logging, etc. *XgponXgtcDsFrame* and *XgponXgtcUsBurst* are used to represent the downstream XGTC frame and the upstream XGTC burst, respectively. There are many other classes used to represent the related data structures, such as BW_{map}, PLOAM message, and the header of PHY adaptation sublayer. They are omitted due to space limitation.

XgponXgemHeader is used to represent the header of the XGEM frames. It is inherited from Header of NS-3 and is added to a packet from upper layer for encapsulation.

4.3.3 Connection Management

Since XG-PON traffic is carried by logic connections, many classes are designed and implemented for representing, organizing, and handling these connections.

XgponConnection is used to represent a connection. It mainly contains the identifiers of this connection. XgponConnectionReceiver and XgponConnectionSender are its subclasses that represents a connection at the receiver and sender side. XgponConnectionReceiver mainly holds the received segments for reassembling. XgponConnectionSender contains the service history (XgponServiceRecord), QoS parameters (XgponQosParameters), and the transmission queue for SDUs from upper layer (XgponQueue).

XgponAlloc is the class for representing T-CONT. XgponAllocOnu, its subclass for the ONU, is designed to organize the upstream connections of the same T-CONT. As for XgponAllocOlt, the subclass of XgponAlloc for the OLT, queue occupancy reports from ONU, QoS parameters and service history of this T-CONT are maintained for DBA. XgponAllocOlt is also responsible to hold the received segments for implementing reassembly.

XgponOnuConnManager contains a list of downstream connections (XgponConnectionReceiver) and a list of the active T-CONTs that have one or multiple upstream connections (XgponConnectionSender). It is also responsible to map SDU/XGEM frame to the corresponding connection. Thus, it implements both Downstream Connection Manager and Upstream Connection Manager for the ONU.

XgponOltConnManager is designed to fulfill the functions of Downstream Connection Manager and Upstream Alloc-ID Manager for the OLT. It contains a list of broadcast connections (XgponConnectionSender). As for the uni-cast connections, they are organized into *XgponOltConnPerOnu*. More specifically, XgponOltConnPerOnu has a list of downstream connections (XgponConnectionSender) and a list of T-CONTs (XgponAllocOlt). XgponOltConnManager stores XgponOltConnPerOnu into one vector and the index equals to the corresponding ONU-ID. When assigning IP address to the network attached to one ONU, we use its ONU-ID

[3]For each downstream connection, the Downstream Connection Manager at the ONU should also have one queue for holding the potential segments.

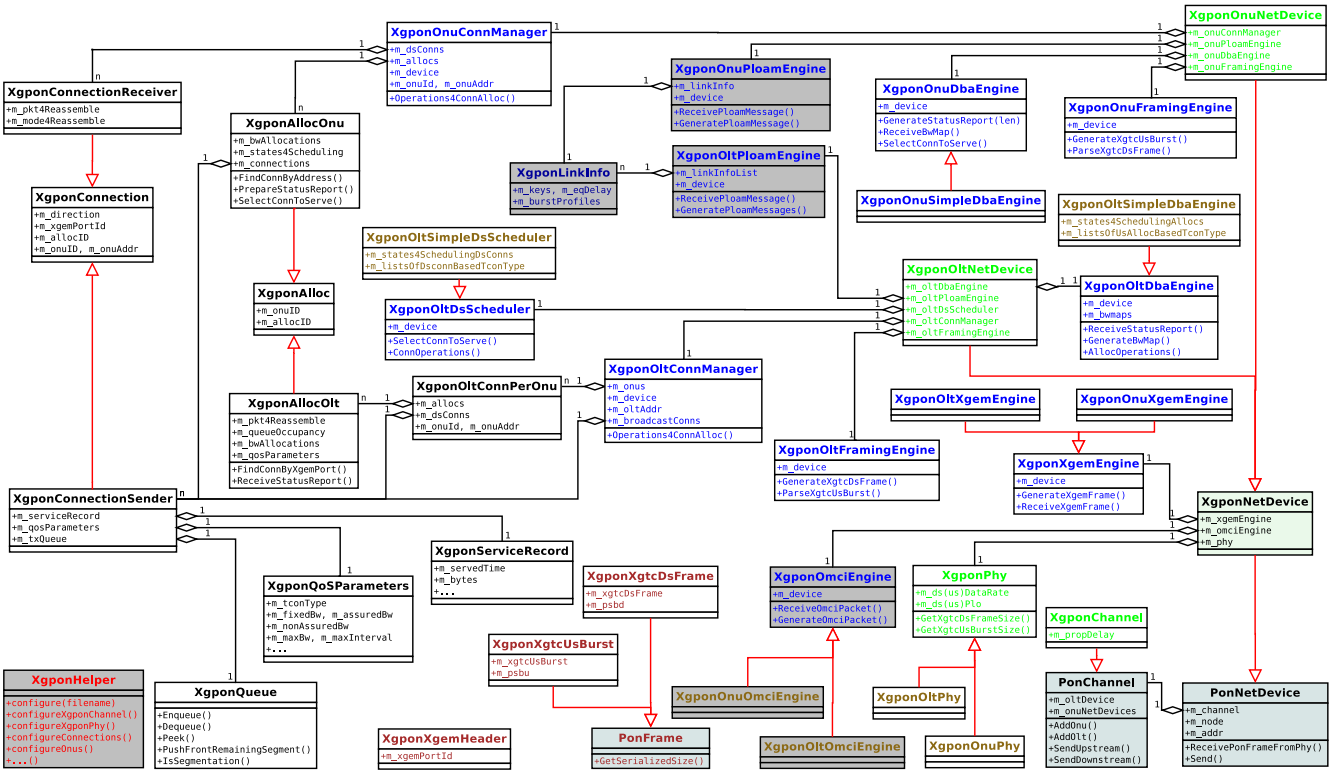

Figure 5: Class Diagram of the XG-PON module for NS-3

as the last 10-bits of the network id. Thus, when receiving one SDU from upper layer, the OLT can find the targeted ONU and the specific connection quickly. It is very important since the OLT could have thousands of connections and need process millions of packets per second.

4.3.4 PMD and PHY Adaptation

PMD Engine and PHY Adaptation Engine in Figure 4 are simplified significantly for simulating XG-PON with reasonable speed. *XgponPhy* is used to implement both of them. Instead of simulating their functions (line coding, FEC, scrambling, etc.) step by step, XgponPhy just passes frames/bursts between XgponChannel and Framing Engine. Hence, we implicitly assume that all frames/bursts can be received correctly. Since the network should be well planned and FEC has been adopted, the observed frame corruption rate will be very low and this assumption is reasonable. In the future, the corruption of frames will be simulated based on empirical measurements from real XG-PON networks.

XgponPhy just simulates the data rates and the bandwidth overhead in both directions. The most important interface is to tell Framing Engine about the amount of bytes that it can transfer within one XGTC frame or burst. For the upstream XGTC burst, the used burst profile has also been considered. In XgponPhy, the data rates and bandwidth overhead can be configured through the attribute system of NS-3. XgponOltPhy and XgponOnuPhy are the two subclasses for the OLT and ONU, respectively.

4.3.5 Framing Engines

XgponOltFramingEngine implements Framing Engine on the OLT side. It is responsible to generate the downstream XGTC frames and parse the upstream XGTC bursts. *XgponOnuFramingEngine* implements Framing Engine on the ONU side. It is responsible to generate the upstream XGTC bursts and parse the downstream XGTC frames. Both of them follow the standard strictly.

4.3.6 XGEM Engine

XgponXgemEngine is responsible to process XGEM frames. It carries out encapsulation, decapsulation, segmentation, reassembly, etc. Segmentation and reassembly are implemented based on the Packet class of NS-3. The logic for data encryption/decryption is also implemented. But the cryptographic algorithm is not implemented and executed for saving CPU. XgponOltXgemEngine and XgponOnuXgemEngine are the two subclasses for the OLT and ONU.

4.3.7 Scheduling and DBA

To study different scheduling and DBA schemes, several abstract classes are used in this module for extensibility. The actual schedulers can then inherit these abstractions and implement their specific algorithm. The related classes in this module are introduced below.

XgponOltDsScheduler acts as the OLT Downstream Scheduler shown in Figure 4. When XgponOltFramingEngine generates a downstream XGTC frame, it will call one virtual function of XgponOltDsScheduler (*SelectConnToServe*) to decide the connection to be served. XgponOltSimpleDsScheduler is one subclass that follows the round robin scheme.

XgponOltDbaEngine is designed for the OLT DBA Engine shown in Figure 4. When XgponOltFramingEngine generates one downstream XGTC frame, it will call one virtual function of XgponOltDbaEngine (*GenerateBwMap*) to get a BW_{map}. It also provides one virtual function to receive queue occupancy reports from ONUs. Currently, a sim-

ple DBA algorithm is implemented in XgponOltSimpleDbaEngine that serves a fixed amount of bytes for each T-CONT in a round robin manner.

XgponOnuDbaEngine acts as the ONU DBA Engine shown in Figure 4. Not only generating queue occupancy report and processing BW_{map}, it is also responsible to schedule the upstream connections of the same T-CONT. Hence, it also acts as the ONU Upstream Scheduler shown in Figure 4. XgponOnuFramingEngine will call one virtual function of XgponOnuDbaEngine (*SelectConnToServe*) to get the connection to be served. XgponOnuSimpleDbaEngine is one subclass implemented to serve the connections of one T-CONT in a round robin manner.

4.3.8 Miscellaneous

XgponOltPloamEngine and *XgponOnuPloamEngine* are designed for exchanging Ploam messages between the OLT and ONU. They also uses *XgponLinkInfo* to maintain per-ONU information, such as keys and burst profiles. As for *XgponOltOmciEngine* and *XgponOnuOmciEngine*, they are designed for implementing the OMCI channel. For these classes, we have just implemented their interactions with other classes of this module. We will simulate their messages and the related procedures in the future.

For facilitating researchers to configure one XG-PON network with hundreds of ONUs and thousands of connections, *XgponHelper* is also designed in this module and we are now finalizing its interfaces.

5. SUMMARY AND FUTURE WORK

In this paper, we present the design and implementation of an XG-PON module for the NS-3 network simulator. As the first XG-PON module for NS-3, we believe that this work is a significant contribution to the scientific community as it allows us to simulate XG-PON and study the performance issues that arise with the deployment of XG-PON.

Our module currently implements the core functions of XG-PON and packets can now be transferred through a simulated XG-PON network. In the future, we will develop the corresponding helper classes so that researchers can simulate the XG-PON network more easily. We will also validate this XG-PON module, evaluate its performance, and keep improving its simulation speed. More DBA algorithms will also be implemented. In addition, we will study how to simulate Fiber to the Cell with this XG-PON module and the WiMAX/LTE modules distributed with NS-3.

6. ACKNOWLEDGMENTS

This work is supported in part by Science Foundation of Ireland through the CTVR CSET grant 10/CE/I1853.

7. REFERENCES

[1] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA).

[2] OPNET Modeler (available at http://www.opnet.com/).

[3] IEEE 802.3ah Task Force, June 2004.

[4] IEEE Std. 802.16-2004, IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems, Octobor 2004.

[5] Gigabit-Capable Passive Optical Networks (G-PON). Rec. G.984.x, October 2008.

[6] The NS-3 network simulator (available at `http://www.nsnam.org/`), 2008.

[7] IEEE 802.3av 10G-EPON Task Force, September 2009.

[8] 10-Gigabit-Capable Passive Optical Networks (GPON) Series of Recommendations. G.987.x, March 2010.

[9] A. Bodozoglou. EPON for OMNeT++. Available at: http://sourceforge.net/projects/omneteponmodule/, September 2010.

[10] C.-H. Chang. *Dynamic Bandwidth Allocation MAC Protocols for Gigabit-capable Passive Optical Networks*. PhD thesis, University of Hertfordshire, 2008.

[11] J. Farooq and T. Turletti. An IEEE 802.16 WiMAX Module for the NS-3 Simulator. In *SIMUTools*, 2009.

[12] M.-S. Han, H. Yoo, B.-Y. Yoon, B. Kim, and J.-S. Koh. Efficient dynamic bandwidth allocation for FSAN-compliant GPON. *Journal of Optical Networking*, 7(8):783–795, August 2008.

[13] T. R. Henderson, M. Lacage, and G. F. Riley. Network simulations with the ns-3 simulator. In *Sigcomm (Demo)*, 2008.

[14] H. Ikeda and K. Kitayama. Dynamic Bandwidth Allocation With Adaptive Polling Cycle for Maximized TCP Throughput in 10G-EPON. *Journal of Lightwave Technology*, 27(23):5508–5516, December 2009.

[15] H. C. Leligoun, C. Linardakis, K. Kanonakis, J. D. Angelopoulos, and T. Orphanoudakis. Efficient medium arbitration of FSAN-compliant GPONs. *International Journal of Communication Systems*, 19:603–617, 2006.

[16] S. McCanne and S. Floyd. The LBNL network simulator (NS-2), 1997. http://www.isi.edu/nsnam/ns/.

[17] D. B. Payne and R. P. Davey. The future of fibre access systems? *BT Technology Journal*, 20(4):104–114, October 2002.

[18] Z. Peng and P. Radcliffe. Modeling and Simulation of Ethernet Passive Optical Network (EPON) Experiment Platform based on OPNET Modeler. In *ICCSN*, 2011.

[19] G. Piro, N. Baldo, and M. Miozzo. An LTE module for the ns-3 network simulator. In *WNS-3 in conjunction with SIMUTools*, 2011.

[20] J. Postel. Transmission Control Protocol - DARPA Internet Program Protocol Specification. RFC 793, Sept. 1981.

[21] D. P. Shea and J. E. Mitchell. A 10-gb/s 1024-way-split 100-km long-reach optical-access network. *Journal of Lightwave Technology*, 25(3):685–693, March 2007.

[22] H. Song, B.-W. Kim, and B. Mukherjee. Multi-Thread Polling: A Dynamic Bandwidth Distribution Scheme in Long-Reach PON. *IEEE Journal on Selected Areas in Communications*, 27(2):134, Feburary 2009.

[23] E. Weingartner, H. vom Lehn, and K. Wehrle. A performance comparison of recent network simulators. In *ICC*, 2009.