Interleaved Polling with Adaptive Cycle Time (IPACT) Implementations Using OPNET

N. Moradpoor*, G. Parr, S. McClean, B. Scotney and G. Owusu (BT Laboratories)
India-UK Advanced Technology Centre of Excellence in Next Generation Networks, Systems and Services
School of Computing and Information Engineering,
University of Ulster, Coleraine, NI, UK, BT52 1SA
**Email: moradpoor_sheykhkanloo-n@email.ulster.ac.uk, IEEE Member*

Abstract

The Ethernet Passive Optical Network (EPON) has been considered as one of the most promising candidates for the next-generation optical access solutions. In EPON, which is also referred as the Time-Division-Multiplexed PON, upstream fibre is shared among multiple users in a timely manner. Therefore, bandwidth allocation is a challenging and critical issue which needs to be addressed efficiently in order to provide diverse Quality of Service (QoS) guarantee for different Class of Services (CoSs). The Interleaved Polling with Adaptive Cycle Time (IPACT) [1] is a classic Dynamic Bandwidth Allocation (DBA) algorithm proposed for the TDM-PON (EPON). It has also regarded as the performance comparison benchmark by the most existing EPON-DBA algorithms.

This paper includes the full implementation of the IPACT using OPNET Modeler [5] and reflects the strong aspects of it as a traditional EPON DBA which allocates bandwidth by taking into account the exact need of each Optical Network Unit (ONU) in every cycle.

Introduction

Broadband PON (BPON, ITU-T G.983.x), Gigabit PON (GPON, ITU-T G.984.x) and Ethernet PON (EPON, IEEE 802.3 ah) are three PON major standards and technologies.

BPON employs the Asynchronous Transfer Mode (ATM) cells for the data transmission over the optical fibre by the maximum speed of 1.244 Gb/s downstream and 622.08 Mb/s upstream data transmissions.

The GPON provides the maximum speed of 2.448 Gb/s for each downstream and upstream data path and employs both ATM cells and Ethernet frames for encapsulating data all the way from the ONU to the OLT in central office (CO). The EPON uses Ethernet frames as the standard data unit and provides maximum speed of 1 Gb/s for both downstream and upstream data paths. It uses Ethernet as the standard interface which is an inexpensive, ubiquitous, mature and popular technology all over the world. The Ethernet is also interoperable with a variety of equipments with adopted Quality of Service (QoS).

EPON appears to be the preferred choice among other PON data link technologies such as: ATM-based PON (APON such as BPON) or SONET-based PON (SPON). For instance, APON adds the additional cost and complexity to the network by breaking the IP packets in the source and reassemble them in the destination or SPON is too expensive for the local loop and not efficient for the data traffic transmissions. A typical EPON environment is a point-to-multipoint (P2MP) network topology including: an Optical Line Terminal (OLT) in central office (CO), Optical Network Units (ONUs) near customer premises and 1:N passive splitter/combiner between OLT and ONUs, Fig.1.

The Time Division Multiplexing (TDM), Wavelength Division Multiplexing (WDM) and Code Division Multiplexing (CDM) are the three media access technologies in PON environment. They have unlike challenging issues in terms of the cost and complexity. In TDM PON such as EPON, BPON, GPON the optical carrier is shared between the ONUs by employing the 1: N passive splitter. The 1: N passive splitter is a simple power that divides the ingoing wavelength from the OLT equally to the supported number of the outgoing fibre. For instance, if the input wavelength is 1Gb/s (like the traditional EPON) and the number of the supported out ports is 16 each subdivision would equally get about 62.5 Mb/s (1 Gb/s divided by 16).

In TDM PON as a wavelength is shared between the ONUs in a timely manner the number of the ONUs is limited in which avoids the long queuing delay inside each individual ONU. The 32 and 16 are the maximum possible splitting ratios for a typical 1 : N passive splitter up to distance of 10 km and 20 km, respectively from the ONU. The TDM PON also needs the synchronization which adds the extra cost and complexity to the network.

WDM PON supports multiple wavelengths over the same fibre infrastructure; therefore, it provides higher bandwidth rather TDM PON. However, it is more costly when it compares to the traditional TDM PON. In order to provide the higher bandwidth for ongoing demands for more bandwidth hungry applications and services such as video conferencing, online computer games, high-definition television (HDTV), music, multimedia, etc. the current traditional TDM PON needs to be upgraded to the WDM PON.

The bandwidth allocations over the traditional TDM PON are the critical issues which need to be considered efficiently in order to provide the End-To-End (ETE) QoS over the optical infrastructure. It aims to arbitrate and utilize the upstream shared wavelength among the associated ONUSs in a cost-aware manner. An efficient resource management mechanism includes the following three steps [3]: Resource Negotiation, Upstream Scheduling and Upstream Bandwidth Allocation.

The *Resource Negotiation* is between the OLT and each individual ONU in order to report the ONU's immediate queue length in every cycle. The *Upstream Scheduling* is inside each individual ONU to arbitrate the uplink transmission order and *Upstream Bandwidth Allocation* is inside the OLT in CO which decides the length of the allocated time slot to each single ONU.

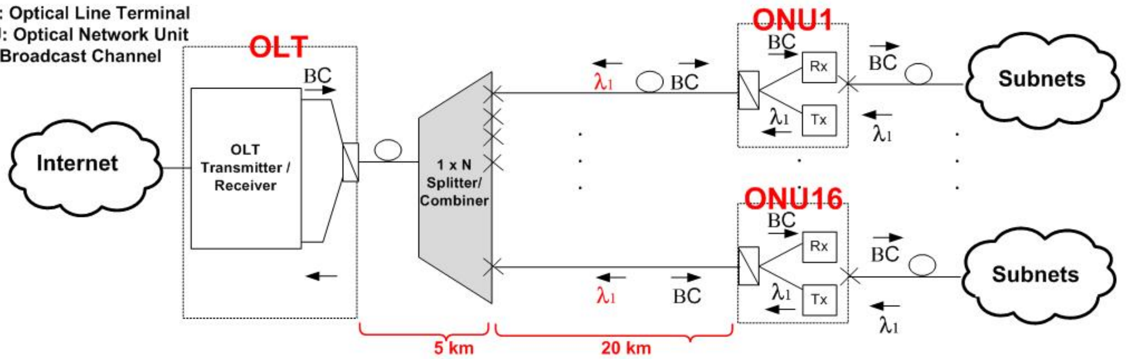

Fig.1 EPON Architecture

Many bandwidth allocation algorithms have been proposed in the literature targeting to improve the resource utilizations over the shared medium in traditional TDM PON (EPON). The extensive review of the TDM-based bandwidth allocation algorithms can be found in [3] and [6] in which the existing PON bandwidth allocation algorithms were discussed in three groups of Fixed Bandwidth Allocation (FBA), IPACT-based Bandwidth Allocation (IBA) and Prediction-based Bandwidth Allocation (PBA). In this paper, we discuss the implementation procedures of the first proposed Dynamic Bandwidth Allocation (DBA) algorithm for TDM PONs which was named as IPACT.

The rest of this paper is organized as follows. In the next section, we detail the Multi Point Control Protocol (MPCP) as the EPON's standard Media Access Control protocol along with all the specified messaging formats. The IPACT theory along with an example has been specified next. We then explain the implementation of the OLT and ONU node models and OLT and ONU process models in simulation scenario. The implementation of the IPACT algorithm inside the OLT in CO is specified next. The simulation experiments have been conducted using OPNET Modeler [5] and initial results have been collected and discussed at the end of the paper.

EPON MAC Protocol

The TDM PON's MAC defines particular control messages, cell or frame fields to enable the resource negotiation process between the OLT and ONUs. The Multi Point Control Protocol (MPCP) was developed by the IEEE 802.3ah in order to negotiate and manage real-time resource allocations between the OLT and ONUs. However, it did not mention any specific bandwidth allocation technique and left it open for venders, researchers and manufacturers all around the world.

MPCP Extension [2] was also proposed to provide wavelength assignment features inside the conventional MPCP. MPCP and MPCP Extension are the two protocols which were defined for TDM-based and WDM-based PON, respectively. MPCP includes five 64-bytes MAC control messages, which facilitate both auto-discovery and real-time resource negotiations in EPON environments comprising: REGISTER_REQUEST, REGISTER, REGISTER_ACKNOWLEDGE, REPORT and GATE messages, Fig.2.

REGISTER_REQUEST, REGISTER and REGISTER_ACKNOWLEDGEMENT are the discovery messages which use to accomplish the registration process for newly joined ONUs. REPORT and GATE messages are the two control messages instructed in MPCP for the resource negotiations and allocations between the OLT and ONUs.

ONU generates the REPORT messages in order to report its latest queue status (buffer length) to the OLT in each cycle. The REPORT message will be passed to the Dynamic Bandwidth Allocation (DBA) Algorithm which resides inside the OLT in CO. The DBA algorithm uses the arrived REPORT messages in order to set up the uplink data transmissions for different numbers of ONU. The DBA decision will then broadcast towards the downstream direction inside the GATE message which includes the ONU id, transmission start time and transmission duration. The GATE message, which carries the DBA arbitration decision, will be received by all ONUs and will be discarded by non-matching ONUs.

There are two options to convey the next REPORT message from the correspondent ONU to the OLT. One way is to tail it to the end of the latest received time-slot on ONU. The other way is to dedicate a very small time-slot for transferring just the bandwidth request from one ONU to the OLT. While the latter way requires twice laser on/off, the former way reduces the laser on/off, overhead and guard time. The former technique is widely used in TDM-based PON resource negotiations. Please refer to our previous work [4] for the full implementation of the EPON MAC protocol using OPNET Modeler [5].

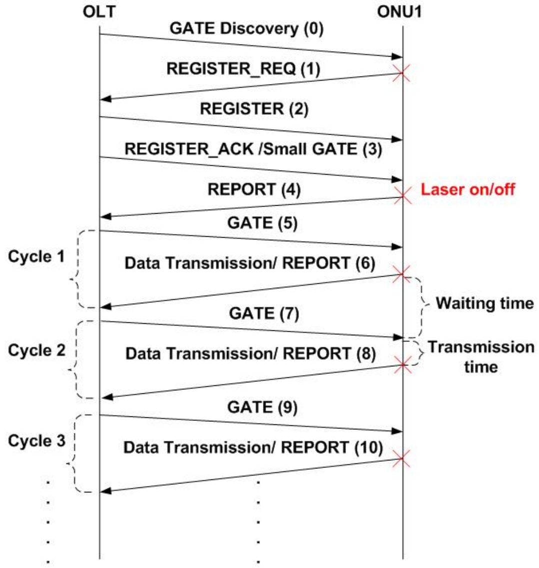

Fig.2 EPON Resource Negotiations

IPAC Algorithm Theory

Interleaved Polling with Adaptive Cycle Time (IPACT) [1] is the first DBA algorithm which was proposed for the TDM-based PON (EPON). It is also regarded as the performance comparison benchmark by the most existing EPON-DBA proposals. The high-level overviews of the IPACT functionality are detailed as the following seven steps. We consider three ONUs in a sample EPON environment for the simplicity of illustrations.

1. We assume that the Table.1 is the polling table which is stored in OLT (CO) in a given cycle C_n . It indicates the latest queue sizes reported by the ONUs at the cycle C_{n-1} and the round trip time (RTT) to each single ONU. The latest queue size is the exact number of bytes, which is buffered in each ONU's queue when the ONU generates the REPORT message.
2. In the given cycle C_n , OLT generates a GATE message for the first ONU in the polling table (ONU1) to let it transfers its data towards OLT (6000 bytes), Fig.3.The 64 bytes GATE message will broadcast to the downstream fibre indicating the specific ONU id (ONU1), transmission start time as well as the transmission length (6000 bytes). However, the received GATE message will be discarded by non-matching ONUs (ONU2 and ONU3).
3. Upon receiving the GATE message from the OLT, ONU1 starts sending its buffered data up to the size of the granted time-slot (6000 bytes) and keeps receiving data packets from the users. At the end of the granted time-slot, ONU1 generates the 64 bytes REPORT message representing its immediate buffer length and tails it to the end bit of the 6000 bytes data.
4. According to the polling table, OLT knows how many bytes it has authorized to the ONU1 as well as the correspondent RTT. Therefore, it knows when the last bit of the ONU1 arrives. This information helps the OLT to schedule and generate a GATE message for the ONU2 in order to transmit 3200 bytes before receiving the ONU1's REPORT message. The OLT also considers a small guard time between two consecutive GATE messages to provide protection for RTT fluctuations and different GATE and REPORT messages processing time of various ONUs.
5. After some times, the data from ONU1 arrives while carries the ONU1's buffer status at the end of the previously granted time-slot. The OLT uses ONU1 REPORT message to update the polling table for the correspondent ONU in order to use it for the next cycle.
6. Similar to the previous steps, the OLT knows when the last bit of the ONU2 arrives, so it schedules and generates the GATE message to the ONU3 to send 1800 bytes towards the OLT. As the result, the first bit of ONU3's date arrives right after the last bit of the ONU2's data.
7. Likewise, before the OLT receives the REPORT message from ONU3, it knows when the last bit of ONU3's date arrives. Therefore, it begins generating and sending a GATE message to the ONU1 for the next polling cycle (C_{n+1}) in such order that the first bit of ONU1's data arrives after the last bit of ONU3's data.

Table.1 OLT's polling table in the given cycle (C_n)

ONU	Bytes	RTT
ONU1	6000	200
ONU2	3200	150
ONU3	1800	170

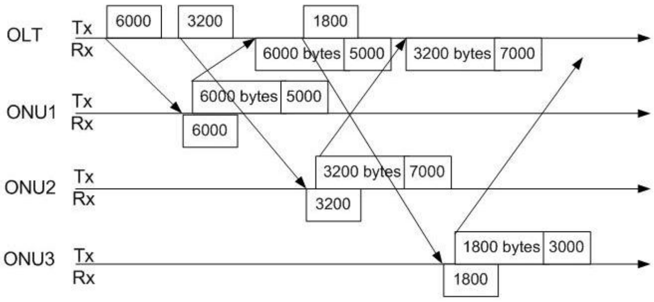

Fig.3 Steps of the IPAC Algorithm

Simulation Model Using OPNET

Our EPON simulation model composed of two key node models (OLT node model and ONU node model) and two key process models (OLT process model and ONU process model). The technical structure of node models and process models will be discussed in next sections.

OLT Node Model

The OLT node model includes two pairs of point-to-point transmitter and receiver (*ptp_tx*, *ptp_rx* and *out_tx*, *out_rx*) and two processors (*OLT_classifier* and *OLT_processor*), Fig. 4. OLT connects to the downlink data path through the *ptp_tx* and *ptp_rx* which help transmitting and receiving traffic flows, respectively between OLT and ONUs. The *OLT_classifier* receives traffic flows from ONUs and then classifies and directs them over different streams. The data traffic will be forwarded outside the EPON system through the *out_tx* which is connected to IP, SONET, ATM, etc. backbones.

The control messages which need to be processed are passed to the *OLT_processor*. The *OLT_processor* is the main processor inside the OLT where the EPON DBA resides and arbitrates the uplink bandwidth negotiations between the ONUs. It processes and also generates the MPCP control messages including both auto-discovery messages (REGISTER_REQUEST, REGISTER, and REGISTER_ACKNOWLEDGEMENT) and bandwidth negotiation messages (REPORT and GATE). The OLT decision sends back to the EPON system towards the *ptp_tx*.

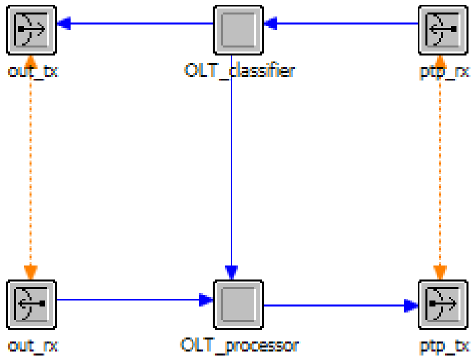

Fig.4 OLT Node Model

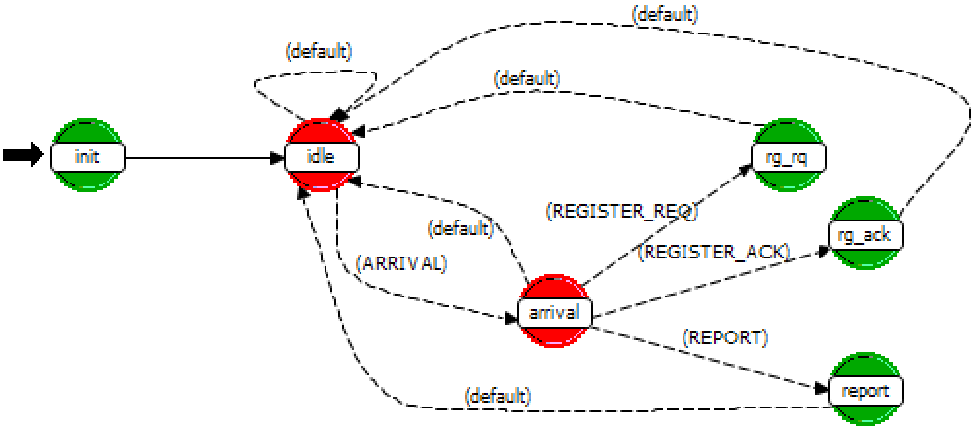

Fig.5 OLT Process Model (*OLT_processor*)

We have developed IPACT [1] algorithm, which is addressed as the standard EPON DBA algorithm, inside the *OLT_processor*. It decides the transmission start time and transmission duration for each single ONU after receiving the REPORT messages. The *OLT_processor* is also able to receive the traffic flows from outside the EPON environment through the *out_rx* and pass it to the entire optical network.

OLT Process Model

We have implemented two process models (*OLT_classifier* and *OLT_processor*) inside the OLT's node model for two different purposes, Fig.5. While the OLT_classifier only helps differentiating dissimilar types of traffic (control messages and data messages) and routing them over two directions, the *OLT_processor* is the ultimate key process model. The *OLT_processor* is where the high-level DBA algorithm is sited. In our simulation, we have implemented the Interleaved Polling with Adaptive Cycle Time (IPACT) as the standard and traditional EPON DBA algorithm inside the *OLT_processor* of the OLT process model.

When the simulation starts, the *OLT_processor* changes from the *init* state immediately to the *idle* state and waits there in order to receive the diverse control messages from the ONUs.

The REGISTER_REQUEST, REGISTER_ACKNOWLEDGE and REPORT are the possible three control messages which can be arrived during the *idle* state. When the message arrives, *idle* state changes to the *arrival* state in order to take appropriate actions with regards to the different message entity. If the arrival message is the REGISTER_REQUEST message, *arrival* state changes to the *rg_rq* state in order to capture, calculate and then save the essential information such as ONU's id and ONU's Round Trip Time (RTT).

The output of *rg_rq* state is REGISTER message, which is an acknowledgment to accept the ONU's request to join the EPON environment. The REGISTER_ACKNOWLEDGMENT is the ONU's response to the REGISTER message which changes the *arrival* state to the *rg_ack* state. It puts an end to the auto-discovery procedure which indicates that henceforth the ONU has been identified by the OLT and is entitled to start sending data traffic.

In order to start sending data traffic, a typical ONU should send subsequent REPORT messages to inform the OLT about its queue size.

The REPORT message changes the *arrival* state to the *report* state where the OLT decides the appropriate time slot allocations to each ONU's single request according to its DBA algorithm. We have implemented the IPACT algorithm inside the *report* state which is used to decide the time-slot distributions between the different numbers of ONU.

IPACT implementations

We have implemented the IPACT as a standard dynamic bandwidth allocation (DBA) algorithm inside the OLT's process model, particularly inside the *report* state, Fig.5. The IPACT algorithm is referred as a performance comparison benchmark by the most existing EPON-DBA algorithms.

In this section, we define the implementation procedure of the IPACT algorithm by assuming Table.1 as the first polling table at the given cycle C_1. The Table.1 is structured when all ONUs finish the auto-discovery processes and report their first queue lengths to the OLT. It indicates the first report messages (queue lengths) received from three ONUs at a given cycle C_0. According to the IPACT algorithm, the OLT pulls each ONU resides in the polling table in a consequent order and allocates time-slots with regards to the reported queue length. The OLT also takes into the account the transmission start time and the transmission duration of the previous ONUs in each single allocation.

During the allocation process, the OLT considers the RTT per ONU per time-slot allocation, which varies from one ONU to another, as well as the 64 bytes space at the end of each allocated time-slot. The 64 bytes space will be used by a correspondent ONU in order to send the next REPORT message to the OLT. The OLT also allocates the guard time between two consecutive time-slots in order to avoid data overlapping between two transmissions.

A cycle is a sequence of the time-slots during which all ONUs will get a chance to send its buffered data tailing by a single REPORT message for the next cycle at the end.

Table.2 OLT’s calculation for the bandwidth allocation between three ONUs at a given cycle (C_1)

ONU	RTT (ms)	Bytes (C_0)	Transmission_duration	Transmission_start_time
ONU1	200	6000	$\frac{6000*8\text{ (bits)} + (64*8)\text{(bits)}}{1\text{Gb/s}}$ + 0.200 (sec) = 0.200048512 (sec) = 200048.512 (μs)	0 + 512 (μs) = 512 (μs)
ONU2	150	3200	$\frac{3200*8\text{ (bits)} + (64*8)\text{(bits)}}{1\text{Gb/s}}$ + 0.150 (sec) = 0.150026112 (sec) = 150026.112 (μs)	200048.512 (μs) + 512 (μs) = 200560.512 (μs)
ONU3	170	1800	$\frac{1800*8\text{ (bits)} + (64*8)\text{(bits)}}{1\text{Gb/s}}$ + 0.170 (sec) = 0.170014912 (sec) = 170014.912 (μs)	150026.112 (μs) + 512 (μs) = 150538.112 (μs)

Referring to the Table.1, the time-slot allocation cycle starts by polling the first ONU in the polling table, which is ONU1, follows by the ONU2 and finally ONU3. The required transmission duration (sec) for a sample ONU (ONU1) is calculated as (1).

$$ONU_{i=1}^{tranmission\_duration} = \left(\frac{ONU_{i=1}^{queue\_length} + 64\ bytes}{link\_rate\ (1Gb/s)} \right) + ONU_{i=1}^{RTT} \qquad (1)$$

The OLT also needs to consider the transmission start time for each single allocation in order to avoid data transmitting towards the uplink shared fibre being collided. This job is done by taking into account the transmission duration of the allocated time-slot to the previous ONU as well as the guard time between two following time-slots (2). We assumed guard time as 512 µs.

$$ONU_{i}^{tranmission\_start\_time} = ONU_{i-1}^{tranmission\_duration} + guard\ time \qquad (2)$$

The calculated transmission start time and transmission duration will be allocated inside the 64 bytes GATE message and then being sent towards the downstream fibre by the OLT (3), (4). The summary of the calculations can be found in Table.2 including three samples ONUs with three reported queue lengths and RTTs at the given cycle C_1 .

op_pk_nfd_set (gate_msg,"transmission_duration", $ONU_{i=1,2,3}^{tranmission\_duration}$*);* (3)

op_pk_nfd_set (gate_msg,"transmission_start_time", $ONU_{i=1,2,3}^{tranmission\_start\_time}$*);* (4)

ONU Node Model

The ONU node model comprises of three pairs of transmitters and receivers, four processors and two queues, Fig.6. The ONU is connected to the uplink shared fibre through the *ptp_tx* and *ptp_rx*. The *ptp_tx* and *ptp_rx* provide connectivity between the ONU and OLT for transmitting and receiving traffic, respectively. The ONU is attached to the customer premises through *ptp_tx_sn1* and *ptp_rx_sn1* in order to transmit and receive data traffic between the users and optical network. In our model, the ONU is connected to two sub-nets, which can be upgraded easily to support more. Each subnet can support thousands of hundred users!

The *ONU_processor_tx* is where the ONU’s unique MAC address is saved. It is responsible to generate the REGISTER_REQUEST message once the ONU joins the optical network.

The *ONU_classifier* receives traffic flows from the ONU’s sub-nets and directs them to the *Data_q* in order to pass through the uplink fibre. It also receives traffic flows routed from the outside using the *ONU_processor_rx* and directs them towards the correspondent sub-nets. The *ONU_prcossor_rx* operates as a classifier for the ONU’s incoming traffic. If the incoming traffic is a REGISTER message, it will generate the REGISTER_ACKNOWLEDGE message and send it immediately back to the OLT using the *tx_q* module. If it is a GATE message, it will be passed to the *ONU_scheduler* in order to use for the further uplink scheduling. And finally, if it is data traffic, it will be directed to the user domain using the *ONU_classifier*.

The *ONU_scheduler* is the most important module inside the ONU’s node model. From the ONU’s point of view, it is the only component which arbitrates the uplink fibre access for the data inside the *Data_q*. The *ONU_scheduler* plans and schedules the uplink data transmission by considering the transmission start time and transmission duration granted by the OLT to the correspondent ONU in the recent cycle. This information is occupied inside the GATE message and received from the OLT through *ONU_processor_rx*. The *Tx_q* is a regular queue which helps buffering and then routing all the traffic originated from the ONU to the outside the ONU’s node model.

ONU Process Model

We have implemented four process models (*ONU_classifier*, *ONU_scheduler*, *ONU_processor_tx* and *ONU_processor_rx*) inside the ONU’s node model for four different purposes, Fig.7. Among them, *ONU_scheduler* is the key process model which arbitrates the access to the uplink fibre for the data inside the *Data_q*.

According to the Fig.7, when the simulation starts, the *init* state immediately changes to the *idle* state inside the *ONU_scheduler* and waits there to receive the GATE message from the OLT.

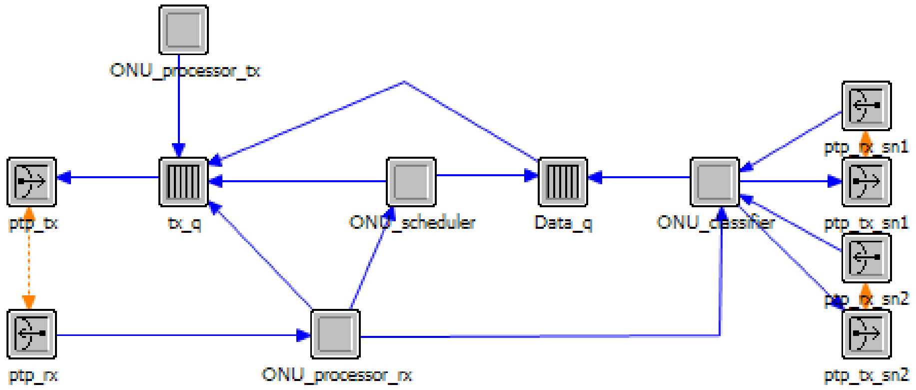

Fig.6 ONU Node Model

The *idle* state changes to the *arrival* state when the GATE message arrives from the OLT, which composes the allocated transmission start time and transmission duration assigned by the DBA inside the OLT in CO. The *arrival* state reads the transmission start time from the GATE message using (5). It schedules an interrupt from the current simulation time to the time specified in the GATE message (transmission start time) using (6) and changes the *arrival* state to the *scheduler* state immediately. The transmission start time is the time during which the correspondent ONU authorized by the OLT to start sending data.

op_pk_nfd_get (gate_msg, "transmission_start", & transmission_start); (5)

op_intrpt_schedule_self (op_sim_time () + transmission_start, SSC_SCHEDULE); (6)

In *Scheduler* state the *Data_q* is notified to start sending data traffic up to the exact queue length (bits) which was reported to the OLT in the previous REPORT message. Then the *Scheduler* state automatically changes to the *idle* state immediately and waits there to receive the next GATE message from the OLT.

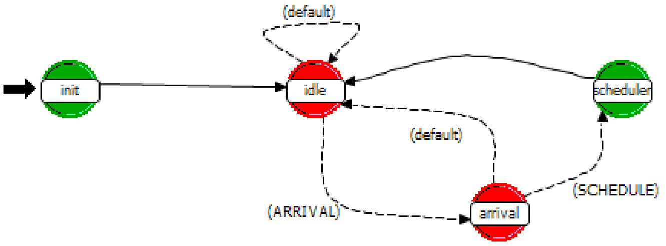

Fig.7 ONU Process Model (*ONU_scheduler*)

Simulation Parameters and Analysis

In order to evaluate the IPACT algorithm our implemented OPNET scenario, we considered the following specifications as the simulation parameters.

We used a system with a given OLT sited inside the CO, which connects the entire optical network to the Internet, a single 1:N passive splitter/combiner between OLT - ONU by the split ratio up to 16 and ONUs which is located near customer premises. The ONUs are all in the same distance of 20 km from the OLT and the distance between each ONU and 1:N passive splitter/combiner is equally set to the 10 km in four scenarios. The 1 Gb/s is the upstream data rate from an ONU to the OLT in CO. The 100 Mb/s is the upstream access link data rate from a given subnets to its associated ONU, Fig.1. We considered the Average Queuing Delay (sec) for two groups of experiences as follows.

For the first group of experiences, we considered scenarios in which each single user (inside the correspondent subnet) generates data traffic (best effort) by the fixed packet size of 4000 bits and packet inter-arrival time of 0.1 sec which is exponentially distributed.

We considered four groups of ONUs (4, 8, 12 and 16) each in a separate scenario and ran all four scenarios for 120 seconds, 1600 value per statistics and 300000 update intervals each. We have captured the Average Queuing Delay inside a given ONU's buffer for four scenarios, Fig.8. As the Fig.8 depicts, the Average Queuing Delay inside a sample ONU raises linearly when the number of ONUs increases from 4 to 8, 12 and finally 16 by almost the constant ratio of 0.8 (ms).

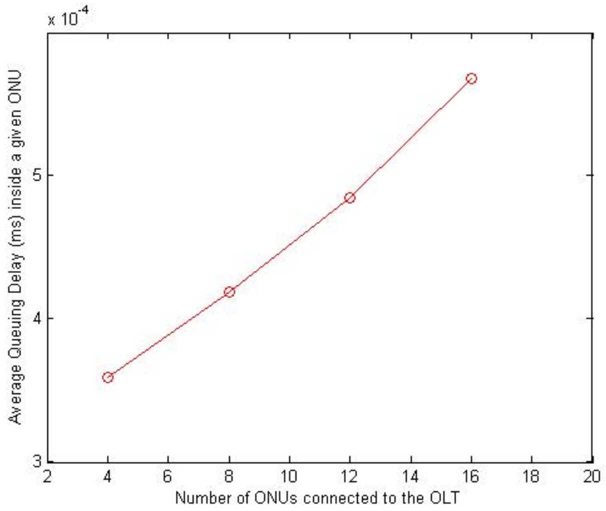

Fig.8 Average queuing delay (ms) inside a given ONU vs. number of ONUs

For the second group of experiences, we considered one single scenario of 16 ONUs under the different load increasing from 0.1 to 0.8 BY 1. The load increment is per ONU as the portion of 100 Mb/s (the upstream access link data-rate of subnet to ONU). For instance, the 0.1 load means: 0.1 x (link rate subnet-to-ONU) = 0.1 x 100 Mb/s, which is equal to 10 Mb/s. We then ran the simulation for 60 seconds.

When the simulation is finished, we captured the Average Queuing Delay (sec) inside a given ONU according to the total network load, Fig.9. The total network load is calculated using the formula (7) as follows.

$$Total\ network\ load = (Number\ of\ ONUs) \times (load\ on\ ONU) \times \left(\frac{subnet\ to\ ONU\ data\ rate}{EPON\ upstream\ data\ rate}\right) \quad (7)$$

For instance, for the load of 0.1 on a given ONU the total network load is calculated as: (16) x (0.1) x (100Mb/s / 1 Gb/s) = 0.16.

As the Fig.9 reveals, when the total network load increases from 0.16 to 0.64 frequently, the average queuing delay inside a given ONU will increase repeatedly by the almost same ratio of 1.2 ms. However, this value reaches to the highest point of 0.005 ms quickly when the total network load achieves 0.8.

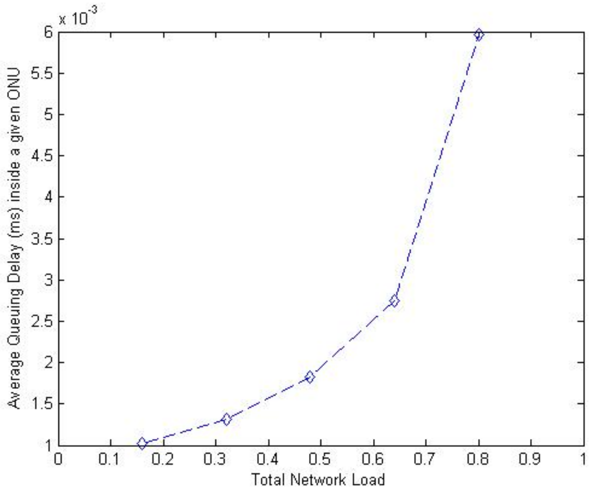

Fig.9 Average queuing delay (ms) inside a given ONU vs. different total network load

Conclusion and Future Work

In this paper, we employed the OPNET Modeler [5] in order to implement the Interleaved Polling with Adaptive Cycle Time (IPACT) algorithm [1]. IPACT is addressed as the first standard DBA algorithm which was proposed for TDM-based PONs (EPONs). It is also regarded as the comparison benchmark for the majority of the existing DBA algorithms for TDM-based PONs. The IPACT functionalities along with the step by step implementations had been fully detailed in this paper. With the aim of evaluating our implementations, the initial simulation results were also captured. The results show the strong relations between the number of the available ONUs, the amount of loads on single ONU and total network load vs. Average Queuing Delay (sec) inside a given ONU. In our OPNET implementation, we only considered one class of service, Best Effort (BE), as the traffic type generating by the front end customers. Our future work is to extend the implemented scenario in order to support diverse Class of Services (CoS) such as Expedited Forwarding (EF) like Voice over IP (VoIP) traffic and Assured Forwarding (AF) like video traffic.

Acknowledgements

The authors would like to acknowledge the support of University of Ulster for providing VCRS scholarship and IU-ATC for funding the joint work with BT.

References

[1] G. Kramer, B. Mukherjee, G. Pesavento, “IPACT a dynamic protocol for an Ethernet PON (EPON),” IEEE Comm. Mgz.,Vo1.40, Issue2, pp.74-80,Feb.2002.
[2] M. McGarry, M. Reisslein, M. Maier, “WDM Ethernet passive optical networks,“ IEEE Optical Communications, Vo1.44, Issue2, S18-S25, Feb.2006.G.Kramer and G.Pesavento, EPON: Building a Next-Generation Optical Access Network, IEEE Communications Magazine, Feb 2002, p: 66-73.
[3] Y. Luo, S. Yin, N. Ansari and T. Wang, “Resource Management for Broadband Access over Time-Division Multiplexed Passive Optical Networks”, IEEE Network, September-October 2007.
[4] N. Moradpoor , G. Parr, S. McClean, B. Scotney and K. Sivalingam, “Simulation and Performance Evaluation of Bandwidth Allocation Algorithms for EPONs”, OPNETWORK2010.
[5] OPNET Modeler 16.0, available at: www.opnet.com
[6] S. Hussain, X.Fernando, “EPON: an extensive review of up-to-date DBA Schemes”, in Canadian Conf. on Electrical and Computer Eng, 2008, Niagara Falls, ON, Canada, pp. 511–516, May, 2008.