CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Refining the GIANT dynamic bandwidth allocation mechanism for XG-PON
Authors	Arokkiam, Jerome A.;Brown, Kenneth N.;Sreenan, Cormac J.
Publication date	2015-06
Original Citation	Arokkiam, J. A., Brown, K. N. and Sreenan, C. J. (2015) 'Refining the GIANT dynamic bandwidth allocation mechanism for XG-PON', IEEE International Conference on Communications (ICC), London, UK, 8 -12 June, pp. 1006-1011. doi: 10.1109/ICC.2015.7248454
Type of publication	Conference item
Link to publisher's version	https://ieeexplore.ieee.org/document/7248454 - 10.1109/ICC.2015.7248454
Rights	© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Download date	2026-11-07 12:21:01
Item downloaded from	https://hdl.handle.net/10468/8987



University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Refining the GIANT dynamic bandwidth allocation mechanism for XG-PON

Jerome A. Arokkiam
Dept. of Computer Science
CTVR, UCC, Ireland
jerome.arokkiam@insight-centre.org

Kenneth N. Brown
Dept. of Computer Science
CTVR, UCC, Ireland
k.brown@cs.ucc.ie

Cormac J. Sreenan
Dept. of Computer Science
CTVR, UCC, Ireland
cjs@cs.ucc.ie

***Abstract*—XG-PON requires an effective dynamic bandwidth allocation (DBA) mechanism for upstream traffic to support quality of service for different classes of traffic. We propose X-GIANT, which extends GPON based GigaPON Access Network (GIANT) DBA, with validated optimisations to the originally proposed key parameters - service timers and assured vs non-assured ratio of medium priority traffic. We implement X-GIANT in a standard-compliant XG-PON module designed for the state-of-the-art ns-3 simulator, tune the above key parameters and show that mean-delay and throughput for different classes of traffic obey the XG-PON requirements and respect priorities at both light and heavy upstream loads. We also show that X-GIANT shows better mean-delay performance than Efficient Bandwidth Utilisation (EBU), a recently proposed, GIANT-derived, priority-based DBA mechanism for XG-PON, for all three classes of traffic simulated.**

I. Introduction

Passive optical networks (PON) have emerged as an attractive access network technology due to their very high bandwidth, low capital cost and relatively low operational and maintenance costs. One of the remaining challenges for PON, as an access network, is to incorporate an efficient Quality of Service (QoS) framework to support different classes of traffic. An abstract QoS structure to be used by the DBA was proposed both in the Ethernet PON (EPON) [5] and Gigabit PON (GPON) [6] standards by IEEE/FSAN and ITU-T respectively. However, very few standard-compliant QoS based DBA mechanisms have been implemented for the Gigabit PON (GPON) standard and its successor, 10 Gigabit-capable PON (XG-PON) [7], due to their restricted definitions. The first such implementation, the GIANT DBA [10], was published in 2006, benchmarking QoS based DBA implementation for GPON. More recently, Han, et.al. have published the EBU DBA [3], which, derived from GIANT, shows significant improvement over the mean-delay performance of GIANT.

In this paper, we refine the concepts of GIANT to comply with the standard of XG-PON and implement the refined DBA, known as X-GIANT, in the state-of-the-art ns-3 simulator. We also implement the EBU, in order to validate the performance improvements achieved by X-GIANT. Our analysis shows that X-GIANT satisfies the mean delay and throughput based requirements for different classes of traffic in XG-PON and surpasses the mean-delay performance of EBU for all classes of traffic. More specifically, X-GIANT ensures constant and minimum mean-delay for highest priority (equivalent to real-time) traffic while ensuring the order of priority among all classes of traffic, when the upstream load is varied from 0.5 to 1.8 times the capacity of XG-PON.

In the rest of the paper, Section II presents the necessary background, Section III discusses related work, Section IV describes the experimental set-up and our refinements to GIANT, Section V contains our experimental evaluation of X-GIANT, and we conclude with directions for future work in Section VI.

II. Background

A. Passive Optical Networks (PON)

The architecture of PON includes an Optical Line Terminal (OLT) - located at the service provider/central office(CO) and connected to a core router - and Optical network Unit (ONU), connected to the OLT via an optical distribution network (ODN) using shared optical fibre and passive optical splitter/jointer. An ONU would usually be placed close to a building or customer premises, making it easier to connect the user/last-mile devices. In the downstream (DS) direction (OLT to ONU), the OLT broadcasts frames to ONUs through the passive optical network using Time Division Duplexing, while in the upstream (US), ONUs transmit frames to the OLT using a Time Division Multiple Access.

As part of the QoS specification, GPON and XG-PON assume different classes of traffic, which must be handled appropriately in using an efficient DBA

B. Restrictions for QoS in XG-PON DBA structure

In XG-PON, OLT uses a DS header section called the $BWmap$ in the DS physical frame to allocate US data transfer opportunities for each ONU. A $BWmap$ can have several $Allocation\ Structures$, each consisting the following fields:

- $Alloc\text{-}ID$ (queue identifier): To identify a traffic flow in each ONU
- Start Time: To denote the synchronised time[1] that each $Alloc\text{-}ID$ can start sending the data in the US
- Grant Size: To refer to the amount of data granted for the particular $Alloc\text{-}ID$ for US transmission

[1]Due to the differences in the distances of ONUs from OLT, this is a synchronised time, achieved by the ranging procedure of XG-PON

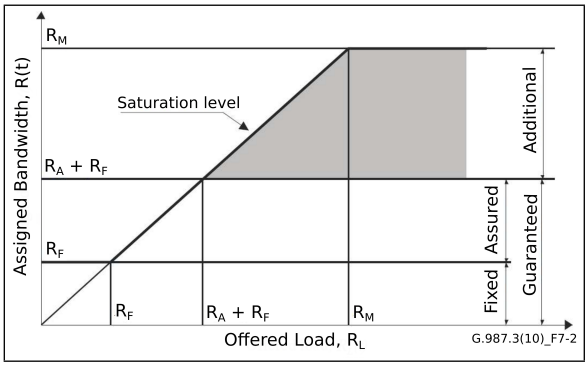


Fig. 1. Distribution of traffic types in XG-PON Vs offered load [8]

- $DBRu$: To allow an $Alloc\text{-}ID$ to request for data in the next US frame
- Fields for other operations and maintenance

Given a scenario that one $Alloc\text{-}ID$ does not have any data allocation in this particular $BWmap$, the minimum $GrantSize$ should be 1 (4 bytes), to allow the particular $Alloc\text{-}ID$ to request an US transmission opportunity in the next allocation cycle[2] [8].

QoS, a feature left at the discretion of vendors by EPON, was also duly recognised by DBA in XG-PON, by defining the following four classes of traffic [8]:

- **Fixed Bandwidth** : Reserved portion of the link capacity, regardless demand and overall traffic load.
- **Assured Bandwidth**: Portion of the link capacity that is allocated to a flow that has unsatisfied traffic demand, regardless of the overall traffic conditions
- **Non-Assured Bandwidth**: Additional bandwidth the OLT dynamically assigns to an $Alloc\text{-}ID$ in proportion to the ONU's total guaranteed bandwidth
- **Best Effort Bandwidth**: Additional bandwidth the OLT dynamically assigns to an $Alloc\text{-}ID$ in proportion to the ONU's total non-guaranteed bandwidth.

Fixed Bandwidth has the highest priority and Best Effort has the least. Additionally, XG-PON also defines an optional class of traffic to facilitate decentralised/ONU-based bandwidth allocation, where the OLT provisions US data transmission based on the aggregated queue status for all four classes of traffic above, such that the bandwidth allocation to each traffic class is at the discretion of ONU.

Additionally, XG-PON classifies both the Fixed and Assured Bandwidth as Guaranteed Bandwidth and the rest as non-guaranteed. The standard commands the DBA to provide the guaranteed portion of bandwidth to all ONUs before allocating the non-guaranteed portion. An illustration of the order of distribution of different bandwidth types against the offered load ($R_L(t)$) can be seen in Figure 1

In order to allocate the both the above portions, XG-PON standard allows status reporting (SR) and traffic monitoring (TM) methods, which can be used at the discretion of the DBA, as long as the basic objectives of the DBA given in the standard are fulfilled. Since the SR mechanism is simple to implement and avoids over-provisioning of upstream bandwidth, most published DBAs use SR based DBA, without making use of TM.

[2]An allocation cycle = filling of 1 BWmap with Allocation structures

III. Related Work

EPON, standardised in 2004, defined a very simple DBA and QoS structure. The GPON standard introduced more restrictive definitions, stressing the need for a concise DBA framework within the standard itself. Hence, though several DBA mechanisms have been proposed for EPON [11], [14], [12] and 10G-EPON in the literature, there are only a few standard-compliant DBA mechanisms for GPON [10], [4], [9], [1], and XG-PON [3].

GIANT [10] DBA was proposed in 2006 to counter the bursty nature of the upstream traffic. It had the following fixed mapping between the classes of traffic and traffic container (T-CONT) types in its implementation:

- T-CONT T1 : fixed bandwidth only
- T-CONT T2 : assured bandwidth only
- T-CONT T3 : assured & non-assured bandwidth
- T-CONT T4 : best-effort bandwidth only

Each T-CONT was given a higher size of US data (allocation bytes or AB in [10]) to be transferred at a fixed interval (service interval or SI in [10], equivalent to integer multiples of the periodicity of a single US frame in GPON). GIANT used minimum AB values ABmin and maximum service intervals SImax for T1, T2, the polling of T4 and the assured portion of T3; the values ABsur and SImin were used for T4 and the non-assured portion of T3. A down counter (service timer in [10]) kept track of each T-CONT type's SI; the expiry of each service timer invoked the allocation of AB for the corresponding T-CONT. A brief description of the choice of AB for each T-CONT type is given in Table I. Further details on calculation of SI and AB can be found in Table I of [10].

An extension of GIANT, the Immediate Allocation (IA) [4] DBA was proposed in 2008, for GPON. IA used an additional parameter ($available\ byte\ counter$ or $VB(j)$) within the algorithm of GIANT, to increase the grant allocation frequency, which purely relied on the service timer expiry of each T-CONT.

EBU [3] was proposed in 2013 for XG-PON, as an improved version of IA. EBU added modification for $VB(j)$ to accommodate negative values so that unused US transmission opportunity from one T-CONT can be utilised by a T-CONT of similar T-CONT type. EBU also introduced S_k, where,

$$S_k = \Sigma_{j \epsilon Z}\ VB(j)$$
$$with\ Z = \{\ i \quad | \quad VB(i) > 0, service\,timer(i) = 0, queue(i)\ \epsilon\ TCONT\ type\ k\}$$

Since S_k is recalculated at the end of each allocation cycle, for each T-CONT in each ONU, EBU injects a high processing overhead for large number of ONUs supported in XG-PON. On the other hand, EBU employed more frequent polling compared to IA, to grant bandwidth to each T-CONT, based on

TABLE I
BRIEF DESCRIPTION OF THE PARAMETERS USED IN GIANT

	Allocation mechanism and calculation of AB
T1	Receives fixed periodic grants, restricted only by PIR ABmin = PIR * SImax
T2	Reservation method ABmin = min{Request Rate,PIR}*SImax
T3	Reservation method First Round: ABmin = min{min{Request Rate, GIR}*SImax, GBS} If excess bandwidth available, then, in the second round: ABsur = min{min{Request Rate, SR}*SImin, PBS-GBS} with SR = Surplus Rate = PIR-GIR, GBS $<$ PBS
T4	Reservation method, but no throughput guaranteed available First Round: ABmin = DBRu length, SImax = polling period If excess bandwidth available, then, in the second round, ABmin = min{min{Available BW, PIR}*SImax, PBS}
	With: GIR,PIR = Guaranteed, Peak Information Rates **GBS,PBS = Guaranteed, Peak Burst Sizes**

more recent T-CONT queue status at ONU. Thus, simulation results presented in [3], at the cost of complexity, showed better mean-delay performance for EBU, compared to IA when the US load was increased from 0.1 to 1.0 times the US capacity of XG-PON. Yet, similar to [4], the authors provided little explanation for the choice of AB, SI and non-assured ratio in T3 and did not test the efficiency of EBU for over-loaded US traffic conditions.

A closer look at GIANT, IA and EBU shows that:

- All three DBAs are built upon the the same principles and algorithmic architecture.
- Modifications in EBU and IA focused on assigning as much bandwidth as possible to demanding T-CONTs over a period of SI allocation cycles, even at the expense of violating the provisioned AB. These are aimed at curtailing the bursty nature of PON's traffic, which was the sole purpose of introducing service timers in GIANT. Yet, IA and EBU showed moderate improvement over GIANT, at the expense of a complex implementation.

We believe that it is possible to achieve the desired performance of an XG-PON DBA for different classes of traffic using a simple and optimised design, naturally resulting in our standard-compliant X-GIANT DBA. Our evaluation also uses the state-of-the-art ns-3 simulator, so as to offer conclusions based on a widely used and validated simulation platform.

IV. DESIGN AND IMPLEMENTATION OF X-GIANT

A. Platform

X-GIANT is implemented as the DBA mechanism for the XG-PON module [13], designed for the network simulator ns-3 (version 3.19 [2]). The salient points of this XG-PON module are as follows.

- As the main goal of the module was to analyse XGTC (XG-PON Transmission Convergence) layer and upper layer issues, the physical layer is designed in a simple way by assuming a satisfactory power budget for ODN.
- Physical Layer Operations, Administration and Maintenance and ONU Management and Control Interface channels are not implemented.
- In the XG-PON channel, packets are assumed to travel without any XG-PON losses, but propagation delays and line rates are configured as per standard.
- Centralised DBA at the OLT is responsible for allocating the US bandwidth to each $Alloc\text{-}ID$, and US scheduler at ONU is responsible for allocating the transmission slot of each $Alloc\text{-}ID$ to its US XGEM (XG-PON Encapsulation Method) Ports.
- OLT and ONU maintain sufficiently large and separate queues for each XGEM Port-ID.
- All ONUs are at the same distance from OLT.

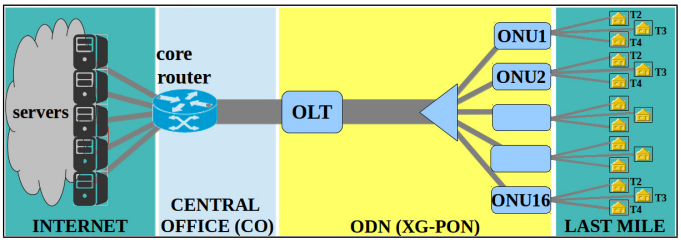


Fig. 2. Network set-up used for simulations

B. Network Architecture And Experimental Environment

Our network topology in Figure 2 is designed to reflect a typical PON access network connected to the Internet, for estimating the mean delay and throughput of different T-CONT based traffic arriving in each ONU. A total of 16 ONUs were used in our experiments, with each ONU, having a set of 3 flows for T2, T3 and T4 T-CONT type each. We did not use T1 traffic flow in any experiment as T1 has a fixed allocation in GIANT [10]. In all the results, mean delay represents the mean value of the difference in time between the arrival time of a certain packet in the T-CONT buffer in each ONU and the departure time of the same packet from the buffer, to be transmitted in the upstream; the reader may add a fixed propagation delay on top of the mean delay for each T-CONT type to obtain the delay for a packet arriving in T-CONT buffer in ONU to reach the OLT buffer. We also used a one-way propagation delay of 400us, to represent 60km distance between OLT and every ONU

We ran each experiment for 20 seconds ($>10^6$ packets when US is fully-loaded), with the in and out times at the T-CONT buffer (fixed at 15KB) being recorded for each packet in each T-CONT flow; packet size was fixed at 1472 Bytes[3] and each T-CONT flow was generated with a fixed and equal arrival rate, such that total incoming traffic from all three T-CONT flows in all 16 ONUs would be between 0.5 and 1.8 times the maximum effective US capacity of XG-PON.

[3]maximum packet size at application layer to avoid segmentation of 1500 Bytes-long XGEM frame

C. Refinements to GIANT

Here, we present the refinements made to XG-PON, with regard to GIANT DBA. We chose GIANT as the base algorithm for refinement, due to the simplicity of GIANT and its algorithmic similarities with IA and EBU.

1. Utilisation of XG-PON US capacity

GIANT allocates bytes to any T-CONT only at the expiry of its service timer. However, when the service timer is set to more than 1 allocation cycle (SImax $>$ 1), there will always be a cycle where the service timer for a given T-CONT is merely updated, without providing any allocation bytes, even though the T-CONT has requested US transmission slots.

Decision: In X-GIANT, we calculate AB, as a percentage of (SImax * US frame size), instead of a percentage of a single US physical frame size. This ensures that, even when service timers of T-CONTs are not expired, sufficient US transmission slots are provisioned by X-GIANT so that US load is utilised to minimum of $\{R_L(t)$, Total US capacity$\}$

2. When total US request $>$ total US capacity

GIANT visits a particular T-CONT only when the allocation cycle has space for the T-CONT to transmit data in the next US frame. But, since the service timer is updated (reduced by 1) only when a T-CONT is visited in an allocation cycle, lower priority T-CONTs, which are not visited at over-loaded US conditions, merely update their service timers when they are eventually visited at a later allocation-cycle. This will lead to higher than required delays for lower priority T-CONT types when the US load increases, resulting in heavy starvation of best effort traffic.

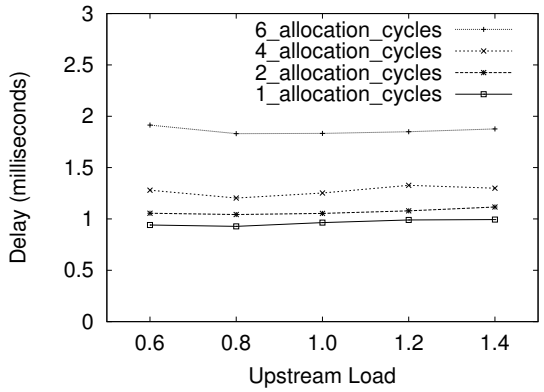
Decision: In X-GIANT we allocate slots to a T-CONT with expired service timer, in the next possible US frame, without waiting for another round of service timer updates, and we update service timers of all the T-CONTs, regardless of whether they are visited by DBA in an allocation cycle.

D. Key Parameters Justified

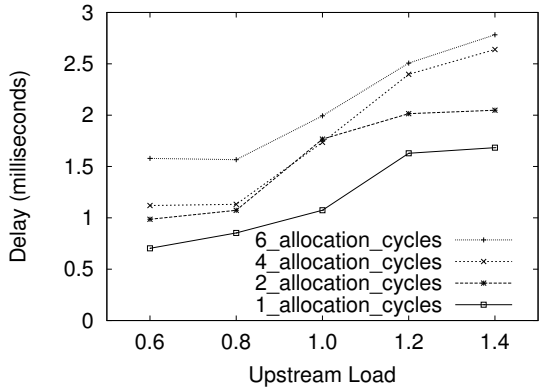
Some parameter values which were left unspecified in GIANT may have a significant impact on performance. We used the experimental set-up in Section IV-B to choose optimised values for the SI and the assured portion of T3, as discussed below.

1. SImax Vs mean delay

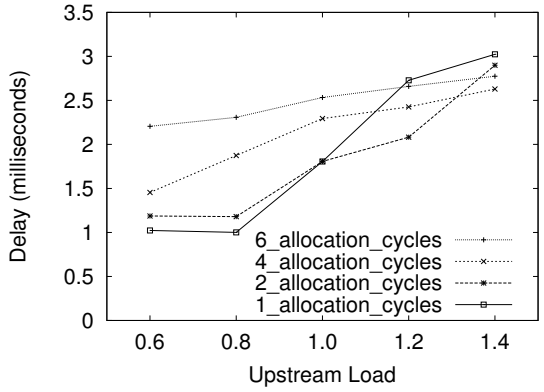
GIANT, IA and EBU use a fixed SImax value of 5 allocation cycles and SImin value of 10 allocation cycles in all the experiments. To try to exploit the higher US bandwidth of XG-PON, we experimented with different SImax values of 1,2,4 and 6 allocation cycles (with SImin = 2*SImax) to determine the impact on mean delay, as shown in Figure 3. For T2 and T3 (Figure 3a, 3b respectively), the delay is minimum when SImax=1; as SImax increases, so does the delay. For T4 (Figure 3c), when US load$<$1.0, delay is minimum when SImax is lowest (=1); however when US load$>$1.0, delay is compromised quickly, as expected, for lower SImax values, at the expense of prioritising, first T2, then T3, traffic.



(a) T-CONT type T2



(b) T-CONT type T3



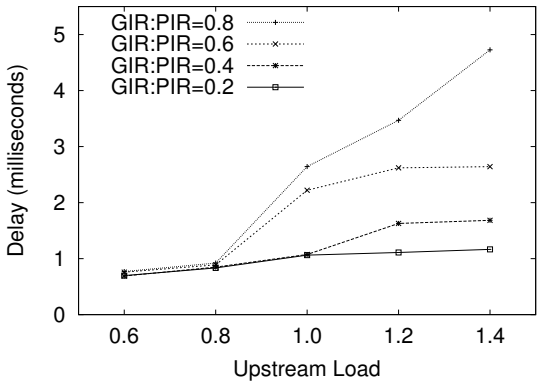
(c) T-CONT type T4

Fig. 3. Mean Delay of each T-CONT type for different SImax values

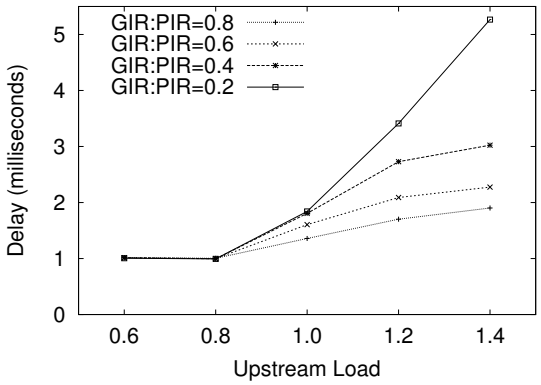
Decision: We fixed SImax at single allocation cycle for T2, assured portion of T3, and polling rate for T4, while maintaining SImax = 2*SImin.

2. Ratio of assured portion of T3 Vs. mean delay

For the T3 traffic, GIANT, IA and EBU used fixed values for the ratio between assured portion (GIR in [10]) and total allocation bytes (PIR in [10]) . In Figure 4, we vary this GIR:PIR ratio, and observe that while the mean delay for overall traffic in T3 increases as we increase the GIR:PIR ratio, mean delay for T4 decreases. For T3, since the serving rate is fixed, increasing the assured portion places more traffic into the assured buffer, hence the higher delay; since T3 traffic is



(a) Mean Delay for T3



(b) Mean Delay for T4

Fig. 4. Mean Delay of each T-CONT type at different GIR:PIR ratios

delayed, T4 traffic is given more data transfer opportunities, showing less mean-delay. The corresponding results for T2 (not included) showed no difference in mean delay as the ratio of GIR:PIR in T3 is varied, since T2 traffic is served before that of T3.

Decision: In order to maintain the low mean delay for the overall T3 traffic, compared to T4, we fixed the GIR:PIR ratio at 0.4 in X-GIANT. This ensured that T4 was always treated at lower priority than T3 traffic at light-loaded, fully-loaded and overloaded upstream conditions.

Table II presents the pseudo-code for X-GIANT, implemented with the above key decisions; function allocate-bytes-to-tcontX(k, timer=SImax) calculates AB, for each T-CONT, based on [10] and Section IV-C.1.

V. Results: Mean Delay and Throughput

We evaluated the impact of light and heavy loading of an XG-PON US using GIANT-derived EBU and X-GIANT, for mean delay of different traffic classes (Figure 5). To provide a fair comparison, we used the same network set-up and packet-recording process (as briefed in Section IV-B) between X-GIANT and EBU. However, EBU was implemented with the same algorithm as in [3] and simulated using its original values for AB ($ABmin_{T2}$ = $ABmin_{T3}$ = $ABsur_{T3}$ = 7812 Bytes, AB_{T4} = 15624 Bytes), SI ($SImax_{T2}$ = 5, $SImax_{T3}$ = $SImin_{T3}$ = SI_{T4} = 10) and GIR:PIR ratio of T3 (= 0.5). X-GIANT, after

TABLE II
Psuedo-code of X-GIANT simulation module

```
GetFirstTcontServed()
for k= 1:4{
(BLOCK-X)
if (iterator.type == k)   {
 if (type-k-already-served)
   last-served-tcont-k-index = allTcontsArray.getIndex(iterator)
 else
    first-served-tcont-k-index = allTcontsArray.getIndex(iterator)
    type-k-already-served = true   }   }
```

```
CheckAllTcontsServed(tcontX)
tcont-type-k = tcontX.type
execute BLOCK-X to find the next tcont in allTcontsArray with type k
for k = 1:4 {
if (timerX of tcontX is expired)
   if (assured-round = true)
      allocate-bytes-to-tcontX (k, timer = SImax)
      reset timerX
   else
      allocate-bytes-to-tcontX (k, timer = SImin)
      reset timerX
   tot-alloc-bytes = tot-alloc-bytes + alloc-bytes-to-tcontX
   prepare the necessary fields in BWmap in XG-PON DS frame
else
   if (lower priority tcont types are not available)
      reset index-of-next-tcont-to-be-served
      exit allocation-cycle
   if (all tconts in allTcontsArray with type-k is served
      if ((k = 4) AND (assured-allocation-round = true))
         next tcont type to be served = 3
         assured-allocation-round = false
      else if ((k = 4) AND (assured-allocation-round = false))
         reset index-of-next-tcont-to-be-served
      else
         tcontX = first tcont in usAllTconts with type k+1
         update index-of-next-tcont-to-be-served
   FinaliseBwAllocation() }
```

```
FinaliseBwAllocation()
update timers for all tconts if (timer != expiry-value-of-timer)
if (total-allocated-bytes > SImax * upstream-frame-size)
   reset index-of-next-tcont-to-be-served, total-allocated-bytes
```

refined from GIANT as identified in Sections IV-C and IV-D, contained different values for AB ($ABmin_{T2}$ = 1564 Bytes, $ABmin_{T3}$ = 624 Bytes, $ABsur_{T3}$ = 1876 Bytes, AB_{T4} = 3128 Bytes), SI ($SImax_{T2}$ = $SImax_{T3}$ = 1, $SImin_{T3}$ = SI_{T4} = 2) and GIR:PIR ratio of T3 (= 0.4). Our results for EBU closely match the results in [3], showing notable increment of mean-delay between 0.7 and 0.9 US loads, for all three T-CONT types. As the load increases from 0.5 to 1.8, the mean delay of T2 traffic remains constant and low (1ms) for X-GIANT, while in EBU mean delay for T2 shows a sudden hike at 0.9 load (due to the buffer being utilised fully) and keeps increasing at a slower gradient as load further increases. The delay for T4 in X-GIANT also show a significant improvement over that of EBU beyond 0.8 load. This is because, while X-GIANT

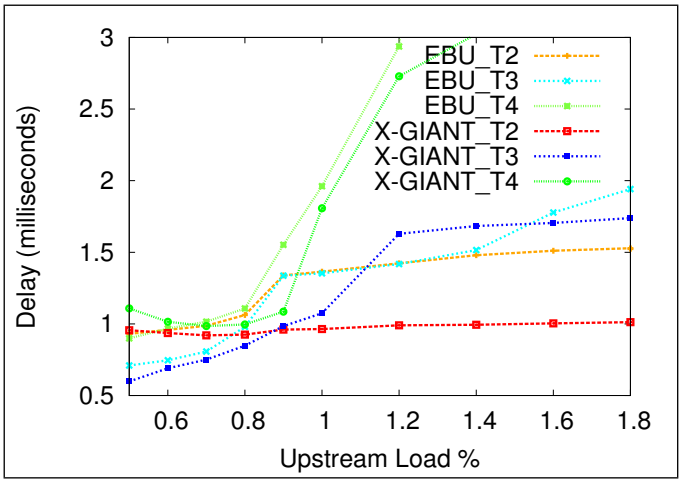


Fig. 5. Mean Delay of T2, T3 and T4 at different loads

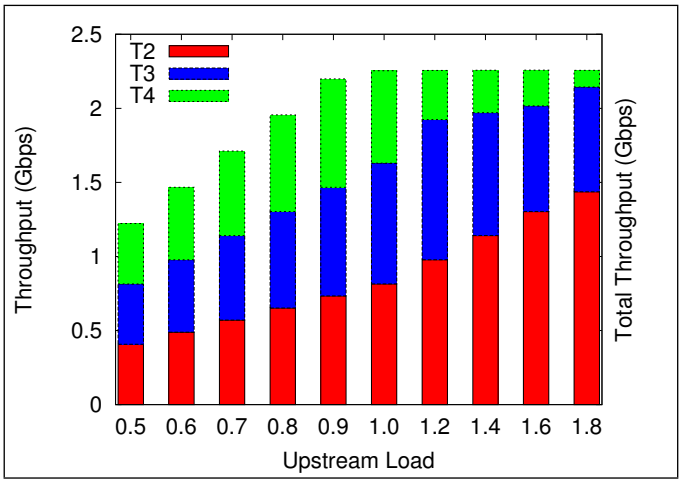


Fig. 6. Throughput of T2, T3 and T4 for X-GIANT at different loads

makes use of early expiry of service timers, even when the upstream load is greater than 0.9, the impact of aggregation of VB(j) in EBU is minimal under similar scenarios.

For T3, both X-GIANT and EBU are able to achieve lower mean-delays, compared to higher priority T2, under load $<$ 1; this is because X-GIANT gives grant to T3 at a higher frequency than for T2 within a single round-robin[4] cycle; EBU, however, provides more grant to T3 than to T2, at equal frequencies. However, when load is increased beyond 0.9, T3 of X-GIANT and EBU takes its priority between T2 and T4 as intended, with X-GIANT achieving lower mean-delay overall.

In Figure 6, we show the breakdown of the throughput for the three classes (when each ONU is loaded with equal and uniform traffic) in X-GIANT. As desired, at light-loaded conditions all traffic classes achieve the same throughput, while in overloaded conditions, T4 throughput is quickly sacrificed in favour of T2, while T3 declines more gracefully because of the assured (40%) portion in T3. X-GIANT is also able to utilise the upstream capacity upto 2.25 Gbps at fully/over loaded conditions, proving the network utilisation capability of the DBA.

[4]A round-robin cycle refers single ABmin to every T2, T3 and T4 and single allocation of ABsur for every T3 and T4 queue in XG-PON

VI. Conclusion and Future Work

GIANT has been used as a DBA mechanism for GPON. We have refined its specification to account for the requirements and opportunities offered by XG-PON, creating X-GIANT. We have implemented X-GIANT in ns3 simulation, and demonstrate that X-GIANT behave as expected for different classes of traffic under a range of US loads. X-GIANT, a simpler DBA compared to EBU, is also able to outperform the latter for all classes of traffic at light and heavy loads.

In future, we will: include SR and TM mechanisms in a single DBA mechanism to better predict of T-CONT queue status and avoid bandwidth over-provisioning; analyse the overhead in US frame when large number of T-CONTs are served in a single allocation cycle; evaluate the complexity of the DBA mechanism against the need for two stage bandwidth allocation for T3; assign weights to unallocated data in each T-CONT queue, based on T-CONT type and waiting time in queue, to ensure intra-T-CONT and inter-ONU fairness

Acknowledgements

This work is supported by the CTVR Grant (SFI 10/CE/I 1853) from Science Foundation Ireland.

References

[1] N. Alvarez, P. Kourtessis, R. Lorenzo, and J. Senior, "Full-Service MAC Protocol for Metro-Reach GPONs," *J. Light. Technol.*, vol. 28, no. 7, pp. 1016–1022, Apr. 2010.
[2] GNU GPLv2, "ns-3.19," 2013. [Online]. Available: http://www.nsnam.org/ns-3-19/
[3] M. S. Han, H. Yoo, and D. S. Lee, "Development of Efficient Dynamic Bandwidth Allocation Algorithm for XGPON," *ETRI J.*, vol. 35, no. 1, pp. 18–26, Feb. 2013.
[4] M.-S. Han, H. Yoo, B.-Y. Yoon, B. Kim, and J.-S. Koh, "Efficient dynamic bandwidth allocation for FSAN-compliant GPON," *J. Opt. Netw.*, vol. 7, no. 8, p. 783, Jul. 2008.
[5] IEEE, "802.3ah: Ethernet in the First Mile," 2004.
[6] ITU-T, "G.984.x: Gigabit-Capable Passive Optical Networks (G-PON)," Rec. G.984.x, Oct. 2008.
[7] ——, "G.987.1: XG-PON General Characteristics," 2010.
[8] ——, "G.987.3: XG-PON Transmission Convergence layer specification," 2010.
[9] K. Kanonakis and I. Tomkos, "Offset-Based Scheduling With Flexible Intervals for Evolving GPON Networks," *J. Light. Technol.*, vol. 27, no. 15, pp. 3259–3268, Aug. 2009.
[10] H. C. Leligou, C. Linardakis, K. Kanonakis, J. D. Angelopoulos, and T. Orphanoudakis, "Efficient medium arbitration of FSAN-compliant GPONs," *Int. J. Commun. Syst.*, vol. 19, no. 5, pp. 603–617, Jun. 2006.
[11] M. Mcgarry, M. Reisslein, and M. Maier, "Ethernet passive optical network architectures and dynamic bandwidth allocation algorithms," *IEEE Commun. Surv. Tutorials*, vol. 10, no. 3, pp. 46–60, 2008.
[12] A. Nikoukar, S. Hwang, A. T. Liem, C. J. Wang, Y. H. Lin, and V. Golderzehi, "A new QoS-aware Green Dynamic Bandwidth Allocation in Ethernet Passive Optical Network," in *2014 Int. Conf. Intel. Green Build. Smart Grid*. IEEE, Apr. 2014.
[13] X. Wu, K. N. Brown, C. J. Sreenan, P. Alvarez, M. Ruffini, N. Marchetti, D. Payne, and L. Doyle, "An XG-PON module for the NS-3 network simulator," in *SimuTools '13*. ICST, Mar. 2013, pp. 195–202.
[14] J. Zheng and H. T. Mouftah, "A survey of dynamic bandwidth allocation algorithms for Ethernet Passive Optical Networks," *Opt. Switch. Netw.*, vol. 6, no. 3, pp. 151–162, Jul. 2009.