Experimental Evaluation of TCP Performance over 10Gb/s Passive Optical Networks (XG-PON)

Jerome A. Arokkiam, Xiuchao Wu, Kenneth N. Brown and Cormac J. Sreenan
CTVR, Department of Computer Science, University College Cork, Ireland
emails: jerome.arokkiam@insight-centre.org, {xw2, k.brown, cjs} @cs.ucc.ie

***Abstract*—XG-PON is the next-generation standard for passive optical networks operating at 10Gb/s and TCP is the dominant transport protocol of the Internet. In this paper, we present the first performance evaluation of TCP over XG-PON, considering efficiency, fairness, responsiveness, and convergence. The impact of XG-PON's large delay-bandwidth product and asymmetric bandwidth provision are assessed, together with the dynamic bandwidth allocation mechanism. Our state-of-the-art NS3 simulation uses real implementations of three TCP variants (Reno, CUBIC and H-TCP) from the Network Simulation Cradle. Our results highlight several issues that arise for TCP over XG-PON, and emphasise the need for improved awareness of medium access control and scheduling in the context of specific TCP congestion control behaviour.**

I. Introduction

Passive optical networks (PON) have emerged as a highly attractive access network technology due to their very high bandwidth, low capital cost and relatively low operational and maintenance costs. The latest standard for PON from the ITU is for a 10 Gigabit-capable PON known as XG-PON, which is seen as a natural successor to the established Gigabit PON (GPON) technology. Once deployed, it can be expected that the majority of Internet traffic carried by such networks will be TCP-based since TCP is still the dominant Transport protocol of the Internet. Thus it is critically important to understand the behaviour of TCP over XG-PON, and the effect of XG-PON's large delay-bandwidth product and asymmetric bandwidth provision. Our paper is the first known study of the behaviour of TCP over XG-PON.

Using our state-of-the-art NS3 simulation for XG-PON, we provide a systematic evaluation of TCP, based on the established TCP Reno, and high-speed variants CUBIC and H-TCP. We expose the complex interaction between TCP based traffic and a dynamic bandwidth allocation (DBA) mechanism used in the Medium Access Control (MAC) layer of the XG-PON. We also demonstrate that TCP flows are not only incapable of fully utilising the XG-PON bandwidth for large round-trip times (RTT), but are also unable to share the bandwidth with other TCP/UDP flows fairly, even when served by a round-robin-based fair bandwidth allocation algorithm at the MAC layer of XG-PON.

This paper is organised as follows. Section II reviews background material while Section III presents the experimental set-up in detail. In Section IV we present and discuss the results. Finally, Section V explains the context for our contribution, and Section VI concludes.

II. Background

A. Passive Optical Networks (PON)

A PON is a point-to-multipoint passive optical technology for broadband Internet access and cellular backhaul. In terms of architecture, a PON comprises an Optical Line Terminal (OLT) which is located at the central office (CO) of a service provider and connected onwards to a core router/switch. On the other end of the PON are Optical Network Units (ONU), which are connected to the OLT via an optical distribution network composed of shared optical fibre, passive optical splitter/jointer, etc. Usually an ONU would be placed close to a building or customer premises, making it easier to connect the user/last mile devices. In the downstream (OLT to ONU), the OLT broadcasts frames to ONUs through the passive optical network, while in the upstream (ONU to OLT) direction, ONUs transmit frames to the OLT using a Time Division Multiple Access (TDMA) scheme. Henceforth we refer to upstream as US and downstream as DS.

In order to avoid upstream collisions between multiple ONUs transmitting at the same time, PON networks employ a bandwidth allocation mechanism on top of a polling mechanism. Ethernet PON (EPON), in IEEE 802.3ah standard, defined a multipoint control protocol (MPCP) in order to facilitate polling using GATE and REPORT messages. However the bandwidth allocation mechanism was left at the discretion of the vendors. Starting with IPACT [9], several dynamic bandwidth allocation (DBA) mechanisms have been proposed for EPON. In GPON, standardised in 2008 by ITU-T (G984.1 and G984.3), polling and bandwidth allocation mechanisms became an integral part of the standard, emphasising the need for a more precise DBA framework within the GPON standard itself.

XG-PON, as the successor to GPON, is the latest PON standard. The XG-PON1 standard was finalised in October 2010 (G987.1, G987.3) by ITU-T, with asymmetrical downstream (DS) line rate of 10 Gb/s and upstream (US) line rate of 2.5Gb/s [6]. In XG-PON, in order to grant US ONU bandwidth to ONUs, OLT uses a section called *BWmap* in the 125us-periodic *DS PHY (physical)* frame. As illustrated in Figure 1, a $BWmap$ can have several $Allocation\ structures$, each addressed to the specific Alloc-ID of each ONU; the $StartTime$ refers to the synchronised time[1] that each Alloc-

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

978-1-4799-3512-3/14/$31.00 ©2014 IEEE

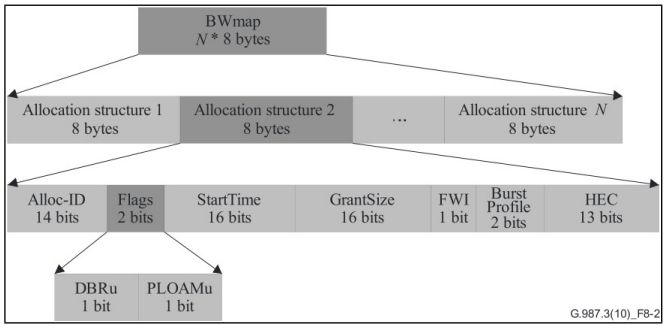


Fig. 1. XG-PON DS BWmap for US transmission opportunity [7]

ID (of the ONU) can start sending the data in the US and the $GrantSize$ refers to the amount of bandwidth granted for the particular Alloc-ID. Given a scenario that one Alloc-ID does not have any data allocations in this particular $BWmap$, the minimum $GrantSize$ should be 1 (4 bytes), to allow the particular Alloc-ID/ONU to request an US transmission opportunity in the next allocation cycle [7].

As part of the standard, the XG-PON defines bandwidth allocation schemes such as fixed bandwidth, assured bandwidth, maximum bandwidth and indication of additional bandwidth requirements (none, non-assured, best-effort). However, similar to EPON, XG-PON also allow the vendors/researchers to experiment with DBA policy, governed by the restrictions of the above schemes [7].

B. Transmission Control Protocol (TCP)

In our study we employ three widely known TCP congestion control algorithms, namely Reno, CUBIC and H-TCP. These are briefly reviewed here; the interested reader is refered to a recent comparison in [11].

1) Reno: Reno is one of the oldest, yet highly stable TCP variant used today. Its operation is well known, with four distinct phases: 1-Slow Start (SS), 2-Congestion Avoidance (CA), 3-Fast Retransmit, and 4-Fast Recovery.

2) CUBIC: CUBIC TCP was designed to take advantage of networks with large bandwidth-delay product (BDP) networks. It differs from TCP Reno mainly in CA (phase 2) and after Fast Recovery. Once segment loss is detected, the congestion window (cwnd) value is recorded as W_{max} and the window is decreased by a factor β in Fast Recovery. From that time a new window, $W(t)$ is calculated as:

$$W(t) = C(t-K)^3 + W_{max}$$

where C is a scaling factor, t is the elapsed time from the last window reduction, W_{max} is the window size just before the last window reduction, and $K = \sqrt[3]{Wmax \star (\beta/C)}$. The window grows as follows for each ACK received:

- If $W(t) < W_{max}$, it is in the concave region of the curve or stable mode. The window is adjusted according to $W(t)$.
- If $W(t) > W_{max}$, it is in the convex region of the curve or exploratory mode. The window is adjusted according to $W(t)$.

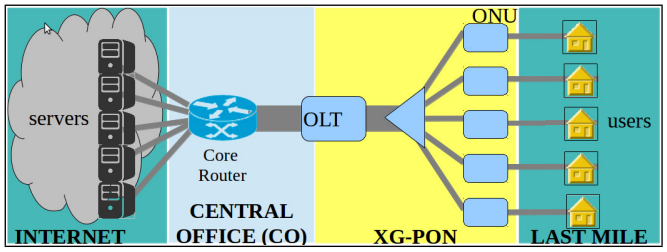


Fig. 2. Network set-up used for simulations

3) H-TCP: H-TCP [3], another high-speed variant of TCP, uses $\triangle$ (the elapsed time since the last congestion event) to adjust the speed of increasing congestion window. The window increase step is varied as a function of $\triangle$, which is also scaled with RTT to mitigate unfairness between competing flows with different RTTs. $\triangle$ is adjusted to improve link utilization based on an estimate of the queue provisioning [10]. In more detail,

$$cwnd \leftarrow cwnd + \frac{2 \star (1-\beta) \star f_\alpha(\triangle)}{cwnd}$$

$$cwnd \leftarrow g_\beta(B) \star cwnd$$

with,

$$f_\alpha(\triangle) = \begin{cases} 1, & \triangle \leq \triangle_L \\ max(f_{\alpha(\triangle)T_{min},1}), & \triangle > \triangle_L \end{cases}$$

$$g_\beta(B) = \begin{cases} 0.5, & |\frac{B(k+1)-B(k)}{B(k)}| > \triangle_B \\ min(\frac{T_{min}}{T_{max}}, 0.8), & otherwise \end{cases}$$

where $\triangle_L$ is a specified threshold such that the standard TCP update algorithm is used during a short period immediately after a congestion event. A quadratic increase function f_α is suggested in [3], namely $f_\alpha(\triangle) = 1 + 10(\triangle - \triangle_L) + 0.25(\triangle - \triangle_L)^2$. T_{min} and T_{max} are measurements of the minimum and maximum RTT experienced by a flow. $B(k+1)$ is a measurement of the maximum achieved throughput during the last congestion epoch.

III. Experimental Set-up and Methodology

The main goal of our experiments is to test the efficiency, fairness, responsiveness and convergence of TCP flows, with/without UDP flows, when XG-PON is used as an access network technology. Our network topology in Figure 2 is designed to reflect a typical PON access network connecting to the Internet.

A. Simulation Platform

*1) **XG-PON and DBA**:* We use the XG-PON module [15], designed for the network simulator ns-3 (version 3.19 [4]). The salient points of the XG-PON module are as follows.

- As the main goal of the XG-PON module was to analyse XGTC (XG-PON Transmission Convergence) layer and upper layer issues, the physical layer is designed in a

TABLE I
PSUEDO-CODE OF RR-DBA IN XG-PON SIMULATION MODULE

```
reset tot-no-of-served-tconts, no-of-scheduled-tconts, tot-allocated-size
save first-tcont served
tot-effective-allocation = us-frame-size - extra-allocation in last bwmap
initialise bw-map
do (while tot-allocation < tot-effective-allocation)
{
calculate size-to-assign for the tcont, subjected to
   max-allocation = 1.5 * us-frame-size - extra-allocation

if there is a valid size-to-assign
   { if this ONU already has an allocation
        add allocation with existing details
     if not
        add new allocation, with new details
        increment no-of-scheduled-tconts
     increment tot-allocation }

if there is no valid size-to-assign
   grant allocation for grant-request

if next-tcont to be served == first-tcont
   break the loop
}
produce bw-map
calculate extra-allocation in this bw-map
```

simple way by assuming a satisfactory power budget for the optical distribution network.

- Physical Layer Operations, Administration and Maintenance (PLOAM) and ONU Management and Control Interface (OMCI) channels are not implemented. Thus, the activation procedure by PLOAM, ranging procedure as well as dynamic configuration of XGEM (XG-PON Encapsulation Method) Port-ID and Traffic Containers (T-CONT) are not implemented.
- XG-PON channel has been modelled as a simple point-to-multipoint channel in DS (and multipoint-to-point in US) with propagation delays and line rates configured as per standards. However, the packets are assumed to arrive, without any XG-PON losses, at their corresponding recipients.
- DBA at the OLT is responsible to allocate the US bandwidth to T-CONT, and US scheduler at ONU is responsible to allocate the transmission opportunity of one T-CONT to its US XGEM Ports.
- OLT and ONU maintain sufficiently large and separated queue for each XGEM Port-ID.
- All ONUs are assumed to be at same distance from OLT.

Table I shows the psuedo-code for the Round Robin DBA (RR-DBA), implemented by the function $GenerateBwMap()$ in the XG-PON module. These details of the RR-DBA are needed to understand the discussions in Section IV

*2) **Network Simulation Cradle (NSC) [8]:*** The NSC is a framework which allows real world TCP/IP network stacks to be used inside a simulation, thus providing highly-accurate models of the protocol performance. The linux-2.6.26 TCP code of NSC (version 0.5.3) was used to simulate the behaviour of Reno, CUBIC and H-TCP. The following options were also enabled for all TCP flows:

- Selective acknowledgement (SACK) and window scale (W-SCALE) options to achieve huge values of congestion window to effectively exploit the XG-PON network, even when a single TCP flow is used.
- Time stamp option to achieve more accurate round trip time (RTT) measurements at Transport layer.

B. General Assumptions and Associated Parameters

- The only bottleneck link in Figure 2 was the XG-PON network; therefore, all other links were configured with higher bandwidth than the XG-PON DS or US capacity.
- All nodes were configured with buffer values higher than the path BDP; additionally all TCP sources had sufficient (unlimited) amount of packets at the application layer, to transmit data for the full duration of the simulations.
- When calculating RTT between server(s) and user(s), processing delays (in microseconds range) between application and transport layers were ignored, as the RTT was calculated between peer transport layers.
- One way delay ($p2pDelay$) between all the nodes (except between OLT and ONU) were set manually. Thus, round trip time (RTT) between a server and the user (or vice versa) is given by,

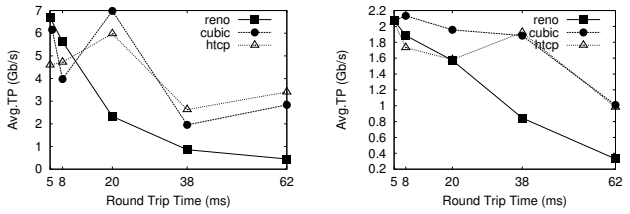
$$RTT = 6 \star p2pDelay + delay_{RR-DBA}$$

where $delay_{RR-DBA}$ is the two-way delay introduced by the RR-DBA.

- After running several experiments, we have decided to fix a value of 2ms for $delay_{RR-DBA}$ for representation of RTT, though $delay_{RR-DBA}$ oscillates around 1.5ms and 2.5ms, for light-loaded and heavy-loaded conditions respectively.
- We used *BulkSendApplication* to generate TCP based traffic in ns-3 while *OnOffApplication* was used to generate UDP based traffic; packet size and Maximum segment size (MSS) were fixed at 1024 bytes and 1500 segments respectively.
- $NscTcpL4Protocol$ in ns-3 was modified such that packet transfer from upper to lower layers was allowed only when the lower layer buffers had sufficient space.

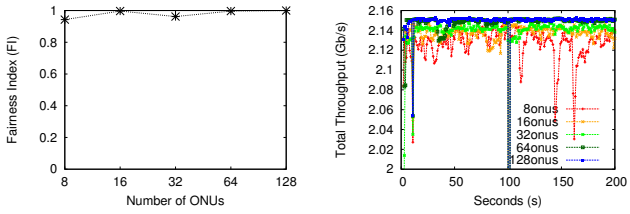
C. Metrics

- **Efficiency:** Efficiency refers to the utilisation of the effective XG-PON bandwidth in DS and US. So we simulated single flow TCP traffic, in both DS and US, for Reno, CUBIC, and H-TCP, with 5 different values of RTT in each scenario.
- **Fairness:** We evaluated fairness of multiple TCP flows (with and without UDP flows) in the US, when RR-DBA was the underlying bandwidth allocation algorithm in XG-PON. This is quantified using the well-known Jain Fairness Index (FI).
- **Responsiveness:** We tested the responsiveness of TCP traffic, when short pulses of UDP traffic were introduced while the TCP traffic is in steady-state. We introduced 10%, 40% and 70% of UDP traffic (percentage of total



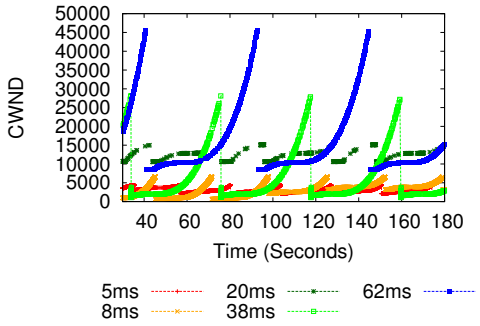
(a) DS (b) US

Fig. 3. Avg.TP for all three algorithms in DS and US

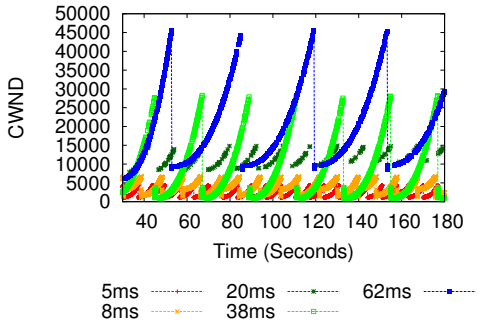


(a) Fairness Index (b) Total Throughput

Fig. 5. FI and Throughput for multiple TCP flows in US



(a) CUBIC



(b) H-TCP

Fig. 4. CWND growth for CUBIC and H-TCP in for all RTT in DS

XG-PON US bandwidth) to represent light, medium and heavy loaded access network conditions. For each load, we simulated pulse durations of 10s, 20s and 40s, to represent short, medium and long pulse durations.

- **Convergence:** Convergence was evaluated by introducing additional TCP flows to an existing steady-state system, in the US, to analyse the convergence time of multiple TCP flows to reach steady-state.

Fairness, responsiveness and convergence was tested only in the US, when a DBA is employed in the XG-PON, owning to broadcast packets not using any bandwidth allocation mechanism in the DS. We adopted the guildelines from [10] to ensure proper analysis of the behaviour of TCP across large BDP networks.

IV. Results and Discussion

A. Efficiency

A single TCP flow was simulated for Reno, CUBIC and H-TCP, both in DS and US, for $p2pDelay$ values of 0.5ms, 1ms, 3ms, 6ms and 10ms (RTT values of 5ms, 8ms, 20ms, 38ms and 62ms respectively). The simulation ran for 200s in each instance. Figure 3 shows the average throughput ($Avg.TP$) of all three algorithms, for all RTT values, where $Avg.TP = \frac{TotalSentBytes_{t2} - TotalSentBytes_{t1}}{t2-t1}$. Here, t1 and t2 are equal to 30s and 180s respectively, corresponding to steady-state behaviour of congestion control algorithms.

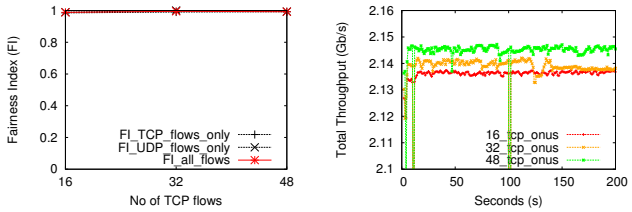
In the DS (Figure 3a), Reno, CUBIC and H-TCP show poor performance for $Avg.TP$ as RTT was increased from 5ms to 62ms. Firstly, Reno shows a linear decrease of $Avg.TP$, owing to slow, linear cwnd increment, as RTT increases. Secondly, even though both CUBIC and H-TCP is able to perform better than Reno in higher RTT/BDP values, they are also unable to maintain an $Avg.TP$ closer 8Gb/s, which is the effective maximum throughput achievable by a UDP traffic through XG-PON in DS. Figure 4 presents the reasons behind such poor performance of CUBIC and H-TCP. As in Figure 4a, for RTT=5ms and 20ms, since CUBIC are able to demonstrate less variation between the highest and lowest cwnd achieved in each congestion epoch, it is able to have an $Avg.TP$ value close to 7Gb/s. The same is observable for H-TCP for RTT=20ms as in Figure 4b. However, with RTT=8ms and 38ms, both CUBIC and H-TCP experience retransmission time-out (RTO) at the end of every congestion epoch, forcing them to restart their cwnd growth from SS phase. For the highest RTT=62ms, both CUBIC and H-TCP achieve higher $Avg.TP$ than for RTT=38ms, by not experiencing RTO, due to having a comparatively less exponential increment towards the end of each congestion epoch.

Between CUBIC and H-TCP, the former shows higher $Avg.TP$, only when its congestion epoch demonstrates an equal concave and convex cwnd growth(RTT=5ms and 20ms), in contrast to a fully exponential growth in the latter. But, when the convex region takes most part of each congestion epoch, H-TCP outperforms CUBIC.

Just as in DS, we also observed degrading $Avg.TP$ performance for all three algorithms in the US direction (Figure 3b). Since the same above explanations apply for Reno, CUBIC and H-TCP in the US and due to space limitation, we have omitted the respective cwnd growth plots.

B. Fairness

1) Fairness Among multiple TCP flows with no background traffic: To evaluate fairness among multiple TCP flows, we simulated multiple ONUs (8, 16, 32, 64 and 128), each using



(a) Fairness Index (b) Total Throughput

Fig. 6. FI and Throughput for multiple TCP and UDP flows in US

a single US TCP flow, for 200s, with CUBIC as congestion control algorithm and RTT=62ms (Figure 5). The FI for all five scenarios are almost equal to 1 (Figure 5a). Additionally, as the number of ONUs increase, total throughput through XG-PON has also increased. That is, as the number of TCP flows increased, per-flow throughput becomes smaller; thus, each flow can only reach a smaller value for the maximum cwnd; as a result, variation between the cwnd of each flow is less at the time of congestion. Since RR-DBA maintains fairness at almost equal to 1 in all the 5 scenarios, the total throughput for all the TCP flows, is also higher when the number of flows are increased.

2) Fairness among multiple TCP flows with multiple UDP flows: To evaluate how the RR-DBA would treat different types of traffic, we simulated a fixed number of 64 flows in the 3 scenarios below, with each ONU having either a single TCP or UDP flow:

- S1: 16 TCP flows and 48 UDP flows
- S2: 32 TCP flows and 32 UDP flows
- S3: 48 TCP flows and 16 UDP flows

As can be seen in Figure 6a, again the RR-DBA is able to provide a very high fairness to all the flows. However, in the case of total throughput, we can see (Figure 6b) an improvement from S1 to S3, even though the total number of flows through XG-PON remains the same at 64. This is for the same reason given in Section IV-B1, as the number of TCP flows served by RR-DBA increases, so does the total throughput through XG-PON. However, we can also see that the increase is not as large as in Figure 5b, since here, the total number of flows through XG-PON remains fixed at 64.

C. Responsiveness

In order to test the responsiveness, we simulated single CUBIC TCP flow along with single UDP flow, for 200s with RTT of 62ms. Even though we have simulated 9 combinations of low, medium and high loads of UDP pulses, with small, medium and long pulse duration, we only present in Figure 7, a single scenario (high load, high pulse duration), as the results from all other combinations are similar. When we introduced a UDP flow with 40s pulse duration and 1.5Gb/s datarate, we did not see UDP taking over the allocated full bandwidth (1.5Gb/s). That is, while the TCP continued to behave as usual, the UDP only takes advantage of the idle XG-PON bandwidth. This is mainly due to the nature of traffic flow input, where TCP asks for a higher amount of data in the current request

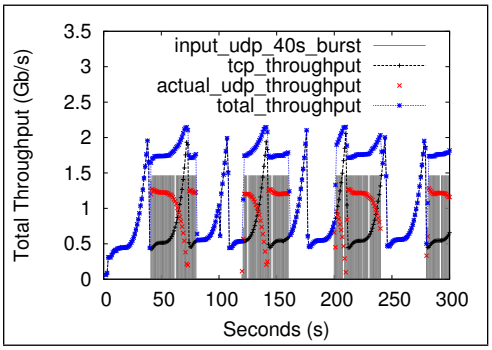
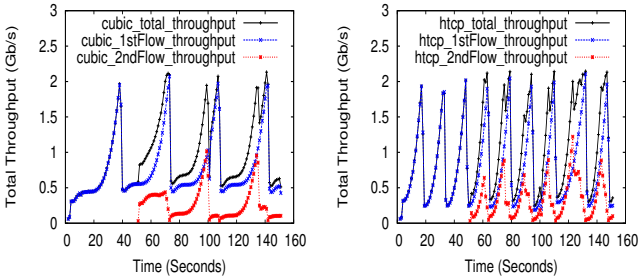


Fig. 7. Co-existence of single TCP and UDP flows in US



(a) CUBIC (b) H-TCP

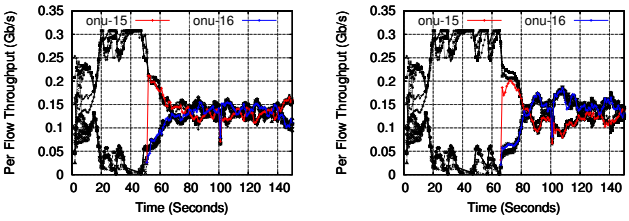
Fig. 8. Convergence of 2 TCP flows in US, using CUBIC and H-TCP

than in the previous request during steady state, while UDP asks for the same amount of data at every request. It is also notable, that since UDP is taking advantage of the varying TCP flow, the total throughput has reached a much higher value during the UDP pulse.

D. Convergence

Figure 8 shows the convergence of two TCP flows (each ONU with only a single US flow, and both flows with same congestion control algorithm and RTT=62ms), when a second flow was introduced after 50s of the first flow's existence in XG-PON. Figures 8a and 8b show that while the first TCP flow is able to achieve higher instantaneous throughput in each epoch, the second flow is only able to have low utilisation in the same congestion epoch as the first. The is because the first TCP flow was asking for a higher amount of US capacity by the time the second flow was introduced. As a result the first flow ends up receiving proportionally higher US bandwidth granted by several grant cycles, due to the fair nature of the RR-DBA, compared to the second flow receiving a lower amount of bandwidth. Due to this, while the first flow continues to grow its cwnd without any disturbance, the second flow experiences an implicit congestion in US, triggering the respective congestion control algorithms to fall back to slow start for the second TCP flow, every time the total capacity reaches its maximum throughput value.

However, we see a completely different scenario when two new TCP flows are introduced after 50s (Figure 9a) and 65s (Figure 9b), for an existing 14 TCP flows (all flows employ CUBIC congestion control algorithm and RTT=62ms). As two



(a) 2 Flows after 50s (b) 2 Flows after 65s

Fig. 9. Convergence of 14 CUBIC TCP flows in US with 2 new flows

flows are introduced at a point, when there is are two constant sets of TCP flows with two different bandwidth allocation patterns, similar to the behaviour of two TCP flows in Figure 8, after 50s. As the total number of allocated flows/ONUs have changed, the RR-DBA and the scheduling mechanism of XG-PON have re-calibrated their grant allocation pattern to reflect the bandwidth request pattern of the particular moment. As a result, only a fluctuating convergence is observed between all the flows, in two sets of bandwidth allocation, with each flow joining a different set in the two scenarios. We also observed similar behaviour for when H-TCP was used instead of CUBIC, as the congestion control algorithm. These results also indicate a poor interaction between XG-PON DBA, scheduling mechanism and the independent cwnd growth of high-speed TCP variants, as far as the XG-PON US bandwidth allocation is concerned.

V. Related Work

There have been several studies of TCP on high BDP networks in the past [14], [1], [5], [10]. Specifically, in [2], the authors have investigated throughput and fairness for TCP traffic over EPON. However, they focused on the MAC without considering the impact of variations in TCP congestion control algorithms and millisecond-range propagation delays and jitter. Interaction between TCP based traffic and the DBA employed in MAC layer of 10GE-PON was also studied in [12]; yet the scope of the analysis was restricted to single congestion control based tests, with more focus placed on TCP based traffic only in the presence of multiple ONUs, without provisioning sufficient buffer at relevant nodes. In [13], the authors analysed the effect of ONU buffers and TDMA Frame duration, for GPON, using single ONU and generic TCP; however, the same authors had also agreed that further analysis should be done with high-speed TCP variants for GPON MAC layer, using multiple competing TCP flows to better understand TCP performance over GPON.

VI. Conclusions

Having systematically analysed the performance of TCP over XG-PON, we conclude that:

- Even though a single TCP flow is able to achieve very high average throughput, close to the effective bandwidth of XG-PON at very small RTT/BDP values, it is not able to achieve such high utilisation at higher RTT/BDP values, even when using high-speed TCP variants like CUBIC and H-TCP, in both DS and US.
- When several TCP flows were introduced, we were able to achieve a very high total throughput, as the fluctuations between the congestion window of each TCP flow is very small, compared to a single flow scenario.
- When two TCP flows were simulated to pass through the XG-PON network, with different starting time, we were unable to achieve a proper convergence between the TCP flows; we also saw a fluctuating convergence when two new TCP flows were introduced to an existing set of 14 flows..

Thus, there is a clear need for further analysis on the interaction between Transport and MAC layers and the need for an improved DBA and scheduling mechanism, to prevent the issues of single layer protection introduced by TCP traffic traversing through different layers of different technologies. In the future, we will focus on designing a DBA mechanism, which can incorporate different scheduling algorithms and quality of service, to achieve better TCP performance through XG-PON.

References

[1] H. Bang, J. Kim, Y. Shin, and C.-S. Park, "Analysis of ONT buffer and power management performances for XG-PON cyclic sleep mode," in *2012 IEEE Glob. Commun. Conf.* IEEE, Dec. 2012, pp. 3116–3121.

[2] K.-C. Chang and W. Liao, "On the throughput and fairness performance of TCP over ethernet passive optical networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 12, pp. 3–12, Dec. 2006.

[3] R. D.Leith, Shorten, and H. Ins, "H-TCP protocol for high-speed long-distance networkstcp3.pdf," in *Proc. PFLDnet, Argonne*, 2004.

[4] GNU GPLv2, "ns-3.19," 2013. [Online]. Available: http://www.nsnam.org/ns-3-19/

[5] H. Ikeda and K. Kitayama, "Dynamic Bandwidth Allocation With Adaptive Polling Cycle for Maximized TCP Throughput in 10G-EPON," *J. Light. Technol.*, vol. 27, no. 23, pp. 5508–5516, Dec. 2009.

[6] ITU-T, "G.987.1: XGPON General Characteristics," 2010.

[7] ITU-T, "G.987.3: XGPON Transmission Convergence layer specification," 2010.

[8] S. Jansen and A. McGregor, "Simulation with real world network stacks," in *Winter Simulation Conference*, 2005.

[9] G. Kramer, B. Mukherjee, and G. Pesavento, "IPACT a dynamic protocol for an Ethernet PON (EPON)," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 74–80, 2002.

[10] Y.-T. Li, D. Leith, and R. N. Shorten, "Experimental Evaluation of TCP Protocols for High-Speed Networks," *IEEE/ACM Trans. Netw.*, vol. 15, no. 5, pp. 1109–1122, Oct. 2007.

[11] L. Marrone, A. Barbieri, and M. Robles, "TCP Performance - CUBIC, Vegas & Reno," *J. Comput. Sci. Technol.*, vol. 13, no. 1, pp. 1–8, 2013.

[12] H. Nishiyama, Z. M. Fadlullah, and N. Kato, "Inter-Layer Fairness Problem in TCP Bandwidth Sharing in 10G-EPON," *IEEE Syst. J.*, vol. 4, no. 4, pp. 432–439, Dec. 2010.

[13] J. Orozco and D. Ros, "TCP Performance over Gigabit-Capable Passive Optical," in *Third International Conference on Access Networks*, vol. 6, 2009, pp. 264–279.

[14] M. Sosic and V. Stojanovic, "Resolving poor TCP performance on high-speed long distance links Overview and comparison of BIC, CUBIC and Hybla," in *2013 IEEE 11th Int. Symp. Intell. Syst. Informatics*. IEEE, Sep. 2013, pp. 325–330.

[15] 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 (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Mar. 2013, pp. 195–202.