Fragmentation in a Novel Implementation of Slotted GPON Segmentation and Reassembly

Yixuan Qin[1], Martin Reed[1], Zheng Lu[1], David Hunter[1],
Albert Rafel[2], and Justin Kang[2]

[1] Department of Computing and Electronic Systems, University of Essex,
Wivenhoe Park, Colchester, Essex CO4 3SQ, UK
{yqin,mjreed,zlu,dkhunter}@essex.ac.uk,
[2] BT, Adastral Park, Ipswich, IP5 3RE, UK
{albert.2.rafel,justin.kang}@bt.com

Abstract. Gigabit passive optical network (GPON) is likely to play an important role in future access networks and the current challenge is to increase the existing GPON bit-rate to 10 Gb/s to provide next generation access (NGA). However, implementing this in a cost-effective manner is difficult and an important research topic. One of the difficulties in implementation for the electronic part of high-speed GPON is the fragmentation feature as it requires multiple pipeline paths. This paper proposes a novel segmentation and reassembly (SAR) scheme, which simplifies the implementation of fragmentation in that it employs fewer FPGA resources and allows a faster hardware clock rate. Analysis confirms that the scheme does not suffer from reduced efficiency in a variety of conditions. It is also backward compatible and suitable for current 1.25 Gb/s and 2.5 Gb/s GPONs. The novel SAR is verified by both a hardware GPON emulator and a software OPNET simulation.

Key words: GPON, FPGA, SAR, fragmentation, pipeline, parallelism, emulator

1 Introduction

Gigabit passive optical network (GPON) is one of the prevailing Optical Access Network technologies, with large-scale deployment being expected soon. A few GPON trial networks are being carried out in the Asia-Pacific region, Europe and North America. GPON uses a point-to-multipoint network architecture and provides a single network topology to provide data, voice and video services with QoS by means of transmission containers (TCONTs) and dynamic bandwidth allocation (DBA).

GPON can support legacy TDM services as well as the growing demands of bandwidth-hungry applications such as high definition television over IP. It is also considered as a candidate for implementing the next generation access (NGA) network which is broadly agreed will supply 10 Gbit/s downstream and the same or lower rate upstream. The technical challenges which must be overcome to implement 10 Gbit/s have been addressed [1, 2]. In addition to supplying reliable and high quality access network service, cost-effective implementation is a big challenge faced by industry. The need for short time-to-market is a key driving force for today's manufacturers, enabling development cost reduction which would yield immediate returns. This paper will address a novel GPON segmentation and reassembly (SAR) method which will, without undesirable trade-offs, simplify implementation permitting higher hardware clock rates, minimize hardware resource usage and ease the development cycle . This should lead to a shorter time-to-market and make implementation more cost-effective, which are critical factors when implementing NGA. This novel GPON SAR has been verified by simulation using OPNET and demonstrated in a GPON emulator which was developed using a field programmable gate array (FPGA) using Handel-C (a proprietary hardware programming language provided by Agility Ltd). It is worth mentioning that this novel SAR method only applies to GPON rather than for example, EPON which does not need the SAR but with lower link utilization.

Pipelining and *parallelism* are commonly used techniques in hardware implementation to improve the efficiency and hardware running speed and are heavily used in the GPON implementations and the emulator described here. How the proposed SAR affects this heavily used *pipelining* and *parallelism* is addressed in the following section. The remainder of this paper is organized as follows. Section 2 describes the architecture of the FPGA-based GPON emulator. Section 3 describes the implementation of parallelism and pipelining. Section 4 addresses the issues of the current GPON SAR. Section 5 describes the novel GPON SAR which makes the implementation of pipelining and parallelism addressed in Section 3 much more efficient and easier; it also compares the new SAR based GPON implementation in terms of FPGA resource usage, timing constraints and design effort. Section 6 describes its advantages by comparing the system efficiency results from OPNET (an event based commercial simulator) by investigating the influence of different network traffic distributions, traffic loads, and GPON line speeds. Section 7 draws the conclusions.

2 FPGA Based GPON Emulator

Two MEMEC FPGA development boards featuring the high speed, high density Xilinx VirtexII Pro FF1152 FPGA are used to emulate the GPON shown in Figure 1. One is used as an OLT and another is used as an ONU. Both boards connect to Ethernet clients which act as the data, voice and video sources and sinks. The ONU generates upstream traffic with different interleave and different frame sizes in order to emulate a whole set of ONUs. The remaining upstream traffic will be contributed by an Ethernet client which is connected to the ONU. This emulator runs at a clock rate of 31.25 MHz and supports 1.25 Gb/s along with 40 bits data width (the generic width of RocketIO – a commercial high speed transceiver built inside the FPGA). Video streaming is demonstrated using this emulator. The OLT uses "hardcoded" grants (no DBA or status reporting) with a token buffer limiting the bit rate to each ONU to 64 Mb/s, emulating

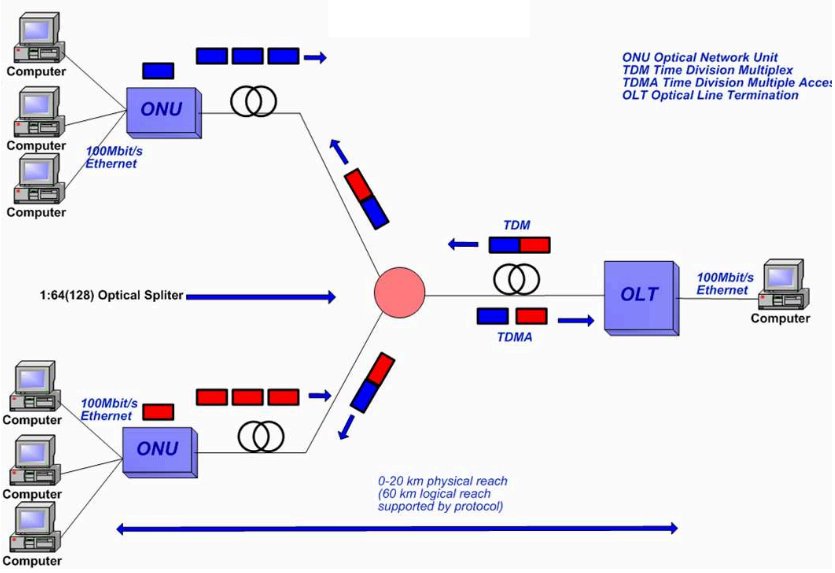


Fig. 1: GPON Emulator Setup

the limitation of competing ONUs (but allowing some peak beyond a 64-way typical maximum split). The emulator features full fragmentation support which is necessary for high-speed network transport. However, this desirable feature might become quite difficult to implement when aiming for very high speeds (over 1 Gb/s), making it very much more difficult to implement within the NGA (10 Gb/s). This GPON emulator demonstrates a novel SAR which we will show maintains high efficiency but is much more cost-effective.

3 Parallelism and Pipelines in the GPON Emulator

3.1 Common Method to Achieve Better Performance and Higher Throughput

Parallelism and pipelining are common methods which are widely used in FPGA design in order to achieve better performance and higher throughput. Hardware parallelism has significant advantages over loosely coupled software parallelism. Software parallelism relies upon a sequential set of statements that can be loosely aligned through operating system parallelism constructs and, unless the number of microprocessor cores is very large, only allows a limited level of parallelism. However hardware parallelism allows a large number of operations to be carried out in a tightly synchronous fashion with advantages in the number of operation that can truly be carried out at the same time and without the difficulties in aligning the operations compared to the software parallelism.

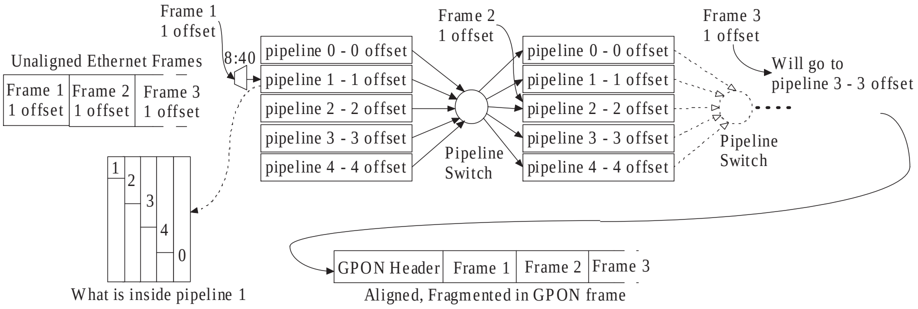


Fig. 2: Figure showing a simplified view of the use of pipelining and parallelism in GPON implementations. Pipeline needed to reduce the clock speed in the FPGA. Parallelism is needed in many parts, here the paralleled pipelines needed to align the output to one of five possible byte alignments is shown.

Pipelining allows operations to be performed on a fast data throughput in a manageable fashion. Data is stored in the pipeline so that it can be processed in parallel. This allows operations that take more than one clock cycle to be performed in a synchronous fashion one clock cycle at a time while the data moves through the pipeline. This requires an operation to be transformed from a straightforward software algorithm (as it is usually described and tested) into an implementation suitable for performing in a pipelined fashion.

While it is essential that both parallelism and pipelining are employed effectively in hardware implementations, they also cause a significant design complexity that increases the time-to-market. Furthermore, complex parallelism and pipelining consume more fabric resource and power. One example of the design penalty introduced by intensive parallelism and pipelining is that a far more complex fragmentation function is needed to align the parallel pipelines. Another example is the greater area used when implementing a high clock rate and high throughput that necessitates increasing the amount of pipelining used.

Thus it is crucial for NGA to retain parallelism and pipelining but implement the full functionality cost-effectively. Firstly, parallelism and pipelining within the GPON emulator, which widely employs the methods discussed above, are demonstrated in the following subsection. Then in Section 5 the improved implementation using the proposed SAR is discussed and compared with other options.

3.2 Common Techniques of Parallelism and Pipelining as used in the GPON Emulator

Instruction level parallelization and task level parallelization are used in the GPON Emulator. The former is based on space and lowest level parallelization, while the latter is based on logic where tasks communicate with one other through a channel or a first in first out (FIFO). Loop level pipelines which will initialize a new loop iteration before the current loop terminates and iterative modulo scheduling [3] are deployed intensively. In more detail, the pipeline has four clock cycles of delay, so that when the segregator prepares "current" data which needs to be sent, the RocketIO is actually sending the "past" data which has already been delayed inside the pipeline. From the other side, when the RocketIO sends the "current" data, the segregator must calculate and prepare the "future" data which will be sent four clock cycles later. Figure 2 shows how Ethernet frames are encapsulated into GPON frame and also shows the parallelism and pipelining deployed.

4 Issues Facing Current GPON SAR Implementations

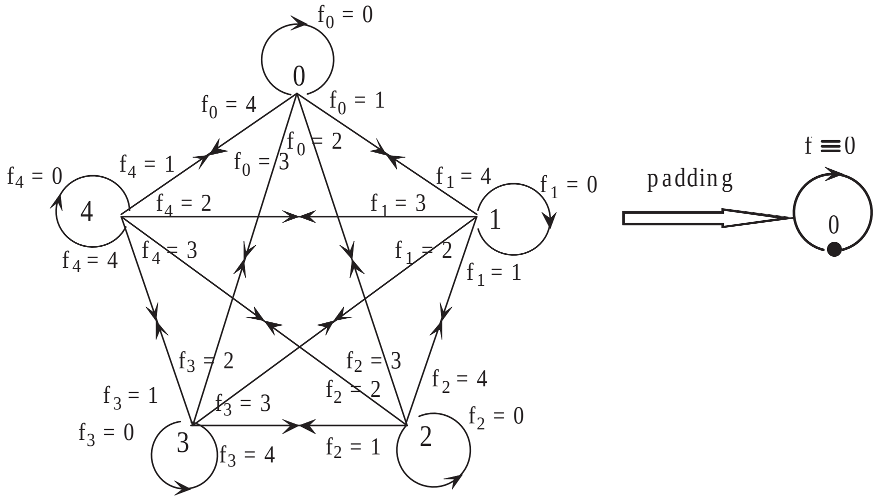
One of the contributions of this paper is the demonstration of a fully functional GPON emulator which can deal with different packet offsets while implementing fragmentation as specified by the GPON standards (G.984.3). This was required to compare with the proposed SAR scheme. The implementation of the standards compliant design demonstrated the substantial work required which may well be a major cost for commercial development, moreover, intensive pipelines and parallelism will cause area and power to impact more on the trade-off between area, power and clock rate. Finally this complexity creates difficulty in implementing a high-speed I/O system to support a 10Gb/s data stream while meeting the real-time constraints on the design.

In this work, the difficulties of implementing NGA are demonstrated and most importantly the major barrier is found in the development procedure. Depending on actual requirements, there are two major issues in the SAR procedure which affect the implementation of parallelism and pipelining, namely *offset* and *fragmentation*.

The first issue (offset) is caused by differing byte offsets of consecutively received data when compared to the hardware data width as used in the serial to parallel conversion and subsequent pipeline. When an Ethernet frame which is considered to represent arriving client data, the segregator will allocate a time slot according to the length of each data field. If the allocated time slot is aligned with the data bus width, then the time slot offset will be always zero, otherwise it will be any one of x as shown in Equation 1. w_d is the data width in bits.

$$x = \begin{cases} \frac{w_d}{8} - 1 & \text{if unaligned} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The second issue (fragmentation) is caused by the demands of network efficiency, i.e. one cannot just waste the remaining space within the frame and wait until next frame rather than fragmenting. Therefore fragmentation is a desirable feature for high speed networks. In order to accommodate frames arriving in different time slots with varying length, the GPON implementation needs to have x replicated pipelines. In the case of the emulator $x = 5$ because the data bus width in the RocketIO is five bytes. Furthermore, the implementation must



(a) For the standard GPON fragmentation

(b) For the simplified state with the proposed implementation

Fig. 3: Time slot offset FSM, each node represents the current state i, $f_i = j$ represents moving to a new offset j.

take into account the merging of data from the five pipelines between any two segments. Note that each pipeline is a complete logic implementation which will run indefinitely. To demonstrate the complexity involved, the Figure 3a shows the finite state machine (FSM) of a time slot. As one can see, transitions exist from each state to each one of the five states (including itself – a fully meshed net). Each state corresponds to one pipeline; whenever new data arrives, a new offset needs to be calculated based upon the current one (the relation is shown in Figure 3). Consequently, the FPGA resource used expands dramatically; timing between logic gates becomes quite difficult to meet which is a barrier to the 10 Gb/s NGA.

5 Novel GPON SAR

The novel SAR reduces the five replicated pipelines to only one which implies one united time slot offset. The Figure 3b is the FSM for the new SAR. The middle part shows the key operation – padding applied to client data before being segregated into GEM payload.

Under the new SAR scheme, the length of the client data is monitored, and the segregator will always align (via padding) the Ethernet frame with the data bus width, consequently only one pipeline is needed to accommodate any length of Ethernet frame without slippage among different time slot offsets. This will reduce the number of large-width comparators which cause considerable delay in the FPGA fabric. Also it still implements fragmentation which makes the proposed SAR as efficient as the standard GPON [4], but greatly reduces the time-to-market and development cost. Moreover it reduces the FPGA resources required to fulfill the same function.

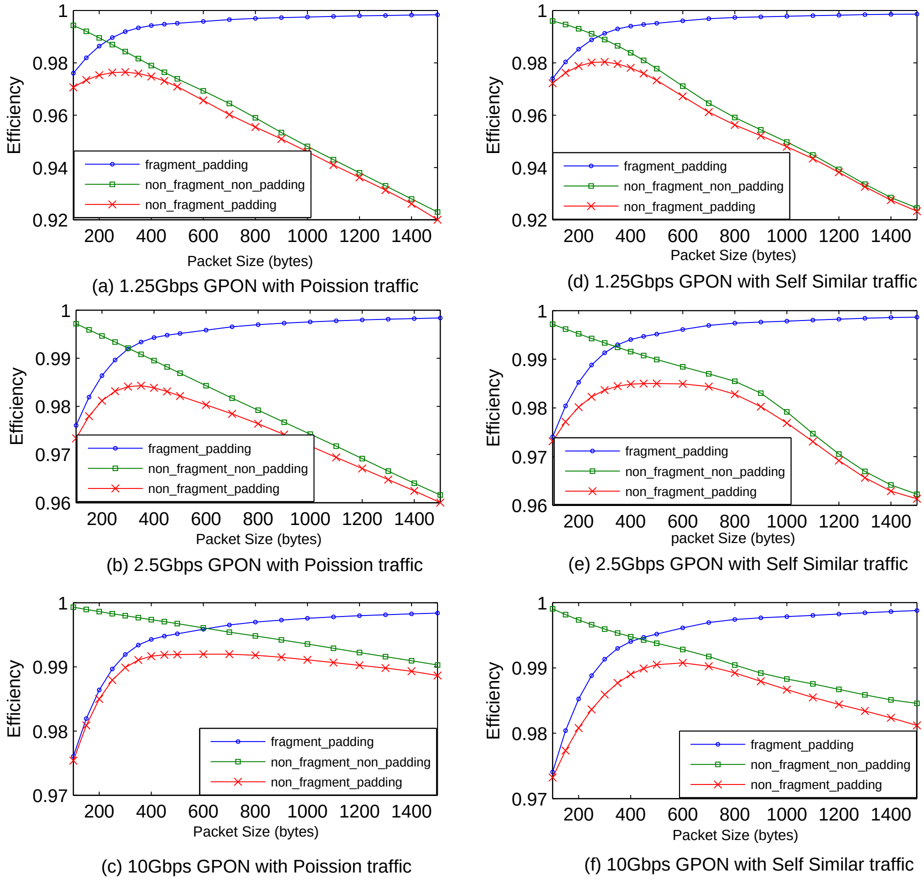
Table 1: GPON SAR Comparison

Standard GPON SAR		NEW GPON SAR	
Logic Utilization	*Used*	*Logic Utilization*	*Used*
Slice Flip Flops	5515	Slice Flip Flops	1205
4 input LUTs	17977	4 input LUTs	4009
Timing Constraints	*Achieved*	*Timing Constraints*	*Achieved*
Shortest Clock Cycle	31.89ns	Shortest Clock Cycle	7.923ns
Design Working hours	*Used*	*Design Working hours*	*Used*
Hours x Person	> 1000hp	Hours x Person	< 300hp

The novel and standard SAR implementation when Place-and-Routed using Xilinx ISE and the key performance attributes of the design are shown in Table 1. The flip flops and look up table (LUT) used are reduced by almost 80%. The timing achieved improves four times; as known in practice, when the timing constraint approaches the limit, to improve it even by 1 ns is very difficult. However the new SAR improves it by 24 ns. This approach is unlike many solutions which trade off area for better clock speed. (e.g. pipeline stages can be inserted in order to increase the clock rate because reducing the number of independent operations in one clock cycle increases the achievable speed). Thus, this is a good solution which uses a much smaller area and has a much faster clock speed while keeping the same functionality. Consequently the power consumed should reduce too (however this is outside the scope of this paper). The most important benefit is the significant reduction in development time (estimated to be of the order of a 70% reduction using the proposed SAR), this is a key driver for manufacturers to deploy this technology widely in practice. All these benefits make the improved GPON a strong candidate for 10 Gb/s NGA.

The same benefits apply to the receiver part because reassembly does not have to take into account the different time slot offsets to maintain five duplicate pipelines; it just treats any unequal length Ethernet frames equally and will only need to have one unique pipeline to reassemble data. The user network interface (UNI) will filter out the time slot information and recover the received data according to the time slot information, then transmit it to clients.

One might argue that if the fragmentation function were omitted, then there would be no need to calculate the data offset; the offset would always be 0 as well, simplifying the SAR. Also one might argue that the offset alignment (padding) will lead to poor efficiency. The following sections will show that fragmentation



(a) 1.25Gbps GPON with Poission traffic

(d) 1.25Gbps GPON with Self Similar traffic

(b) 2.5Gbps GPON with Poission traffic

(e) 2.5Gbps GPON with Self Similar traffic

(c) 10Gbps GPON with Poission traffic

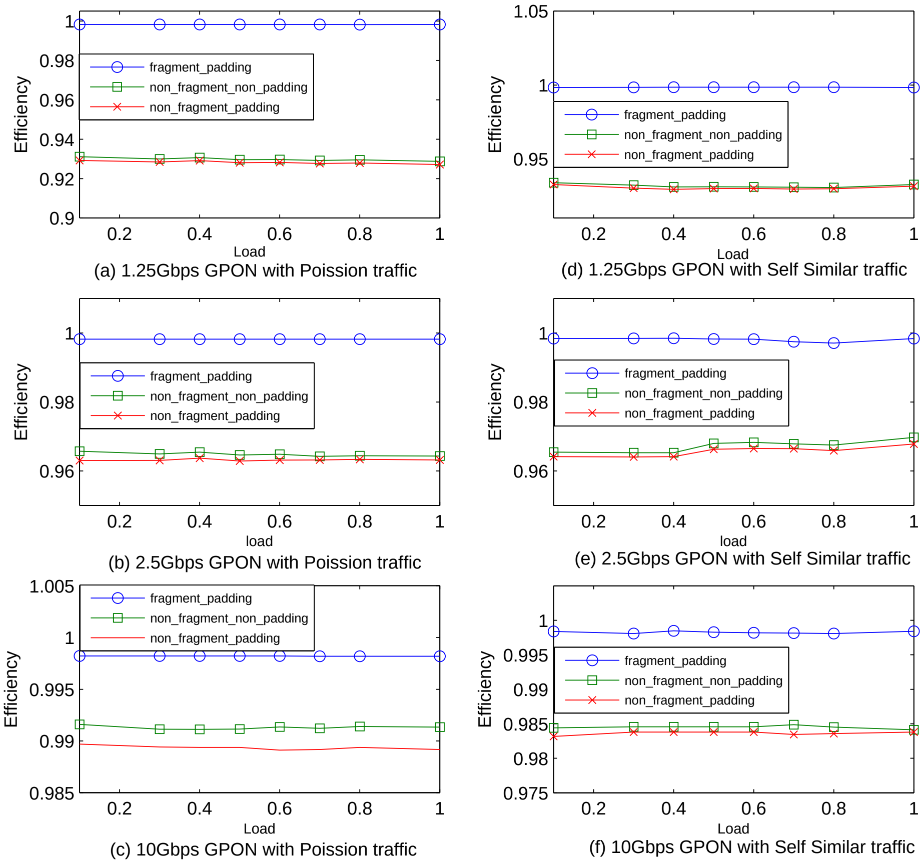
(f) 10Gbps GPON with Self Similar traffic

Fig. 4: Packet Size v.s. Efficiency. Choose average load of 0.7

is an important feature which improves efficiency considerably, and padding only affects efficiency to a small, quite acceptable, extent.

6 Efficiency Advantages Verified by OPNET Simulation Results

There has been considerable debate about what statistical distribution is most suitable when modeling network traffic. The Poission process was widely used when modeling traffic characteristics [5], then researchers argued that network traffic is self-similar [6]. Currently there is again a body of opinion arguing that network traffic obeys the exponential distribution when many self-similar traffic streams aggregate together [7]. Debating Internet traffic models is not the purpose of this paper, however. Regardless of this question, the proposed



(a) 1.25Gbps GPON with Poission traffic

(d) 1.25Gbps GPON with Self Similar traffic

(b) 2.5Gbps GPON with Poission traffic

(e) 2.5Gbps GPON with Self Similar traffic

(c) 10Gbps GPON with Poission traffic

(f) 10Gbps GPON with Self Similar traffic

Fig. 5: Load v.s. Efficiency. Choose average packet size of 1370 byes

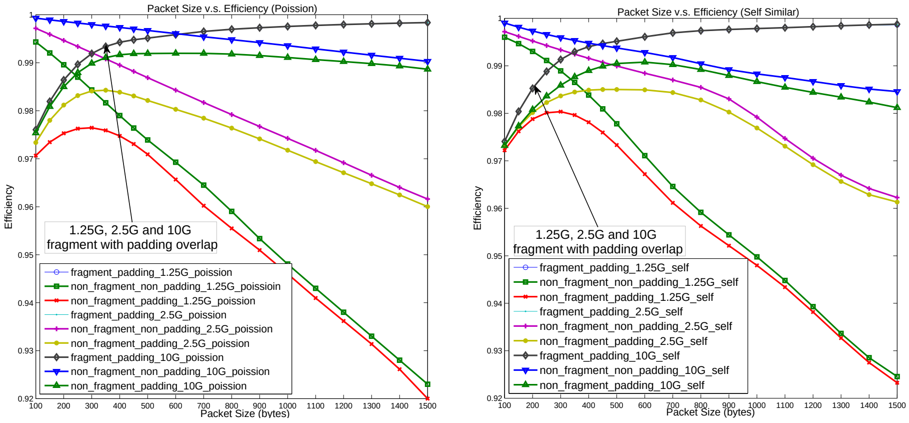
GPON SAR is shown to have superior performance with various of traffic type. The simulation scenarios are shown in Table 2.

The network is assumed in normal usage to have a load of 0.7. The GPON generic overhead factor of 0.06 is not considered, only the proposed SAR efficiency is taken into account to make its influence clear. The packet size obeys an exponential distribution. The mean packet size varies from 100 to 1500 bytes with step size of 200 bytes. The Hurst Parameter of the packet size is 0.7, and the Fractal Onset Time Scale is 1.0 for a self-similar distribution. In order to verify the scalability of the novel SAR for the NGA, the GPON capacity is simulated from 1.25 Gb/s up to 10 Gb/s. Assuming that the GPON employing the standard SAR (fragmentation but no padding) has efficiency of unity, the efficiency of the GPON using the new SAR (fragmentation with padding) is studied against packet size and load via simulation. Also the other two possible options, i.e. “no fragmentation and no padding”, and “no fragmentation but with padding”, are simulated as well in order to make a comparison. Finally, real Internet trace

Table 2: Simulation Scenarios

Traffic Load = 0.7		
	Traffic Distribution	
GPON Capacity	Poission	Self similar
1.25Gbps	Efficiency *v.s.* Packet size	Efficiency *v.s.* Packet size
2.5Gbps	Efficiency *v.s.* Packet size	Efficiency *v.s.* Packet size
10Gbps	Efficiency *v.s.* Packet size	Efficiency *v.s.* Packet size

Packet size = 1370 byte		
	Traffic Distribution	
GPON Capacity	Poission	Self similar
1.25Gbps	Efficiency *v.s.* Traffic load	Efficiency *v.s.* Traffic load
2.5Gbps	Efficiency *v.s.* Traffic load	Efficiency *v.s.* Traffic load
10Gbps	Efficiency *v.s.* Traffic load	Efficiency *v.s.* Traffic load



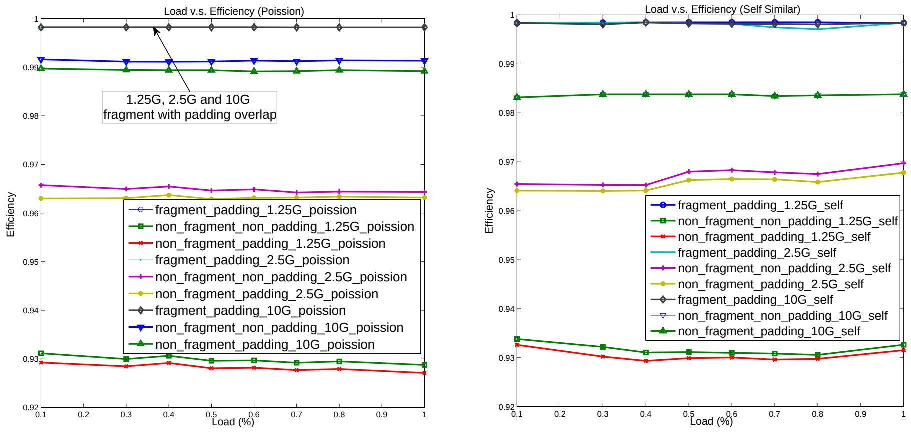
(a) Poission Traffic (b) Self-similar Traffic

Fig. 6: Packet Size v.s. Efficiency. Assume 0.7 load of the full capacity

files [8] are used as source data to verify the correctness of the aforementioned simulation. In this paper the bandwidth efficiency E is defined as follows:

$$E = \begin{cases} \frac{lt_e}{lt_e} \equiv 1 & \text{if (fragment with padding, standard)} \\ \frac{lt_e}{lt_e + pt} & \text{if (fragment with padding, proposed)} \\ \frac{lt_e}{lt_e + nf_w} & \text{if (without fragment without padding)} \\ \frac{lt_e}{lt_e + pt + nf_w} & \text{if (padding without fragment)} \end{cases} \tag{2}$$

lt_e is the total length of an Ethernet frame, pt is the total length of the padding, nf_w is the total wasted time slots in bits due to non-fragmentation. The results shown in Figure 4 correspond to the scenarios shown in the upper part of Table 2 and show no matter what type of traffic the efficiency against packet size with different types of traffic (Poisson and self-similar) and with



(a) Poission Traffic

(b) Self-similar Traffic

Fig. 7: Load v.s. Efficiency. Assume 0.7 load of the full capacity. Choose average packet size of 1370 bytes

different GPON capacities (1.25 Gb/s, 2.5 Gb/s and 10 Gb/s). The proposed GPON SAR which is depicted by the curves with circles, dots and diamonds respectively (fragment with padding) always increases while package size increases no matter what traffic type and what GPON capacity is employed. The lowest efficiency of the proposed solution is 0.97 and it approaches unity as the packet size increases. If fragmentation is omitted as mentioned at the end of Section 5, with no padding (the curves with squares, crosses and triangle-downs respectively, non fragment non padding), the whole trend is of dropping efficiency. Obviously with no fragmentation but with padding (depicted by the curves with times, asterisks and triangle-ups respectively) efficiency is lowest. The proposed SAR solution out-performs the other two options when the packet size is greater than 200 bytes. In Figure 4 (a), when the packet size is less than 200 bytes, the curves with circles, dots and diamonds (non fragment non padding) are higher than the one we proposed (fragment with padding) namely the curve with squares, crosses and triangle-downs. That is because when the packet size is small, the number of padded bytes is comparable to the original length, hence *padding* dominates the influence of efficiency. When packet size increases, in this case bigger than 200 bytes, the fragmentation dominates and affects efficiency to a greater extent. When each Ethernet frame size is small, then one GPON frame contains more Ethernet frames and is padded more often. When the Ethernet frame size is sufficiently large, the number of Ethernet frame in a GPON frame becomes smaller, then they have a lower probability of being padded and the last part will waste time slots if there is no fragmentation applied. Also the crossing point of the curve with circles can be found with the curve with squares – this moves to the right part of the x-axis when the GPON capacity increases. This is because when GPON capacity increases, the GPON frame size increases linearly (19440 bytes for 1.25 Gb/s, 38880 bytes for 2.5 Gb/s and 155520 bytes for 10 Gb/s), then the padding effect will increasingly dominate.

The graphs in Figure 7, corresponding to the lower part of Table 2, show the efficiency against traffic load with different types of traffic (Poission and self-similar) with different GPON capacities (1.25 Gb/s, 2.5 Gb/s and 10 Gb/s), from which the proposed GPON SAR (the curve with circles, dots and diamonds respectively, fragment with padding) always performs better than the other two options (more than 0.995) and the efficiency is not affected by the load variation no matter what the traffic distribution and GPON capacity are. As listed in Table 2, the packet size has an exponential distribution with mean size the same as the video streaming data size, i.e. 1370 bytes. The numerical relationship of the three curves corresponds to the one in Figure 6 when the packet size is 1370 bytes. From this it can be found that the efficiency is mainly dependent on packet size rather than the traffic load, traffic distribution and GPON capacity. Figure 6 and 7 also compare the influence of GPON capacity; all the curves move to the top part of the y-axis when GPON capacity increases. All the values can be verified with reference to Figure 4 and 5. Because of the very high efficiency of the proposed SAR, the three curves (1.25 Gb/s, 2.5 Gb/s and 10 Gb/s) almost coincide (approaching the standard efficiency of unity). Finally, the results are verified by using an Internet trace file as the source [8]; the results are shown in Table 3 which are all very close to the other simulation curves.

Table 3: Simulation Results on Internet Trace

	Efficiency		
GPON Capacity *Gb/s*	Fragment with Padding	No Fragment No Padding	Padding without Fragment
1.25	0.978140084	0.996378372	0.976086154
2.5	0.978149578	0.996614944	0.976207582
10	0.978153429	0.996893023	0.976520454

7 Conclusions

From the co-verification of a hardware GPON emulator and a software OPNET simulation, the proposed GPON SAR dramatically simplifies implementation compared to that proposed in the GPON standard. In particular it should be noted that it occupies almost 80% less FPGA resource, achieves a hardware clock rate which is four times faster and in the experience of the authors reduces development time by 70%. Thus this scheme is a practical candidate for 10 Gb/s NGA. Using simulation it was demonstrated that the proposed SAR retains the important fragmentation feature with almost the same efficiency as standard GPON. It is robust to different traffic distributions, packet sizes and traffic loads, and is suitable for GPONs with different speeds between 1.25 Gb/s and 10 Gb/s.

References

1. D. Nesset, R. Davey, D. Shea, P. Kirkpatrick, S. Shang, M. Lobel, and B. Christensen, "10 Gbit/s bidirectional transmission in 1024-way split, 110 km reach, PON system using commercial transceiver modules, super FEC and EDC," in *ECOC 2005*, Sep. 2005, pp. 25–29.
2. S. Kimura, M. Nogawa, K. Nishimura, T. Yoshida, K. Kumozaki, S. Nishihara, and Y. Ohtomo, "A 10Gb/s CMOS-Burst-Mode Clock and Data Recovery IC for a WDM/TDM-PON Access Network," Tech. Rep., nov. 2004.
3. M. Lam, "Software pipelining: An effective scheduling technique for VLIW machines," in *Proceedings of ACM SIGPLAN Conference on Programming Language Design and Implementation*, 1988, pp. 318–328.
4. I.-T. Recommendation, "G.984.3: Transmission convergence layer specification," 2006, with amendment.
5. H. Heffes and D. Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance," *Selected Areas in Communications, IEEE Journal on*, vol. 4, no. 6, pp. 856– 868, Sep 1986.
6. W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Netw.*, vol. 2, no. 1, pp. 1–15, 1994.
7. J. Cao, W. Cleveland, D. Lin, and D. Sun, "Internet traffic tends toward Poisson and independent as the load increases," *Nonlinear Estimation and Classification*, 2002.
8. Four million-packet traces of LAN and WAN traffic seen on an Ethernet, "The Internet Traffic Archieve sited at the Lawrence Berkeley National Laboratory." [Online]. Available:`http://ita.ee.lbl.gov/html/contrib/BC.html`