Adaptive Multi-Gate polling with Void filling for long-reach passive optical networks
— Source link 🔗

Abhishek Dixit, Goutam Das, Bart Lannoo, Didier Colle ...+2 more authors

Institutions: Ghent University

Related papers:

- Multi-thread polling: a dynamic bandwidth distribution scheme in long-reach PON

- Interleaved polling versus multi-thread polling for bandwidth allocation in long-reach PONs

- IPACT a dynamic protocol for an Ethernet PON (EPON)

- Synergized-adaptive multi-GATE polling with void filling: Overcoming performance degradation in LR-PONs

- Offline and Online Multi-Thread Polling in Long-Reach PONs: A Critical Evaluation

Share this paper: 📘 🐦 in ✉

View more about this paper here: https://typeset.io/papers/adaptive-multi-gate-polling-with-void-filling-for-long-reach-5de4unn7jv

Adaptive Multi-Gate Polling with Void filling for Long-Reach Passive Optical Networks

Abhishek Dixit, Goutam Das, Bart Lannoo, Didier Colle, Mario Pickavet, Piet Demeester
Dep. of Information Technology, Ghent University – IBBT, Gaston Crommenlaan 8/201, B-9050 Gent, Belgium
Tel: +32 9 331 49 77, Fax: +32 9 331 48 99, e-mail: abhishek.dixit@intec.ugent.be

ABSTRACT
In this paper, we introduce a new dynamic bandwidth allocation (DBA) algorithm for long-reach passive optical networks (LR-PON) based on the multi-threads (MT) polling algorithm. For the proposed new DBA algorithm, named as the Adaptive Multi-GAte polling with Void filling (AMGAV), the number of threads changes according to the traffic load of a PON in contrast to the static number of threads used in the MT polling algorithm. This makes the algorithm adaptive to load variations and it improves the delay performance for both higher and lower load scenarios. Moreover, the AMGAV algorithm requires no extra effort to enforce separation between threads which makes it simpler to implement than the MT polling algorithm. In the process, we also propose a novel bandwidth reporting scheme which does not have to keep track of the number of threads. Finally, through extensive simulations, we prove that for LR-PON, the AMGAV algorithm provides a better delay performance than the MT polling algorithm for a network load up to 90%, and a comparable performance at higher loads while improving the channel utilization.
Keywords: Long Reach PON (LR-PON), Void Filling, Multi-threads, Adaptive Multi-Gate Polling.

1. INTRODUCTION

With the advances in optical technology, the reach of passive optical networks (PONs) has increased from 20 km to 100 km or more. A PON with an increased reach is referred to as SuperPON or Long-Reach PON (LR-PON) in the literature [1]. There are various reasons to investigate LR-PON. The LR-PON can simplify the network by minimizing the number of equipment interfaces, network elements and nodes. Also, it is worth to increase the network span to cover more users to make full use of the increased optical capacity delivered by higher line rates (e.g., 10 Gbps) in the time-division multiplexing (TDM) domain and/or the usage of wavelength-division multiplexing (WDM) technology. We propose a TDM based dynamic bandwidth allocation (DBA) algorithm which can be used in LR-PON using either a TDM PON or a hybrid WDM/TDM PON technology [2].

In TDM PON, several DBA algorithms have been proposed of which IPACT is the most important example for Ethernet PON (EPON) [3]. In traditional TDM PON DBA algorithms, a new communication (thread) between an optical line terminal (OLT) in the central office and an optical network unit (ONU) at the user end is only started after the completion of the previous one, and these algorithms are referred to as single-thread (ST). The authors in [4] have shown that ST DBA algorithms lead to a degraded performance (i.e. an increase in the average packet delay) in LR-PON because of the large round-trip time (RTT) between an OLT and an ONU. To mitigate this large RTT effect, the authors in [4] have proposed the multi-threads (MT) polling algorithm where many parallel communications (or threads) are started between an OLT and an ONU.

In this paper, we propose the Adaptive Multi-GAte polling with Void filling (AMGAV) algorithm to overcome some problems of the current MT polling algorithm. In Section 2, we describe the problems of the MT polling algorithm and give a motivation for the proposal of a new DBA algorithm. In Section 3, we describe the AMGAV algorithm and in Section 4 we discuss the simulation results. Finally, Section 5 concludes the paper.

2. PROBLEM STATEMENT AND MOTIVATION

To understand how MT polling helps to reduce the average packet delay, we briefly discuss the components of the queuing delay (d) of a packet, as given in (1) [3]:

$$d = d_{poll} + d_{grant} + d_{trans} \qquad (1)$$

where:

- d_{poll}: time between a packet arrival at an ONU and the next "Request" sent by that ONU. On average, d_{poll} is half of the cycle time (the time interval between two successive Requests from the same ONU).
- d_{grant}: time interval between the ONU's Request for a transmission window for a packet till the grant from the OLT for that packet is received.
- d_{trans}: transmission delay after the appropriate grant from the OLT arrived and the packet is transmitted.

In Fig. 1 (a) and (b), we show the various components of the packet delay in the ST and MT polling algorithm, respectively. We see that in the MT polling algorithm, a new Gate message may be issued before the arrival of the previous Request from the ONU in contrast to the ST polling algorithm where a new Gate message is issued only after the arrival of the previous Request from the ONU. In the ST polling algorithm, due to the increased RTT in LR-PON, the scheduling cycle increases [4]. This eventually increases d_{poll} and results in more packets being buffered in the ONU between two successive Requests from that ONU. As a result, the number of bytes

transmitted by each ONU in one scheduling cycle will be increased, which in turn increases d_{grant} and d_{trans}, as shown in Fig. 1 (a). Although the MT polling algorithm improves the delay performance for LR-PON compared to the ST polling algorithm, we identify the following problems with the MT polling algorithm:

Static number of threads (Issue 1) — We see that as the number of threads or Gate messages increases, the ONUs are polled more frequently leading to reduced delays but it may lead to a large number of Gate and Request messages and thus it will waste bandwidth in control message transmission and guard band overhead. Obviously if the load is very high, the penalty of bandwidth wastage in control message transmission and guard band overhead is more severe. *This motivates us to make the number of threads transmitted in one cycle adaptive to the instantaneous network load and not static as in the MT polling algorithm.*

Thread spread and convergence (Issue 2) — The MT polling algorithm suffers from the problem of thread spread and convergence. Thread spread occurs whenever a thread data cycle (i.e. the time period of data transmission of all ONUs in one thread) becomes larger than the RTT of the most distant ONU, i.e. larger than Max(RTT). In this case, the MT polling algorithm reduces to the ST polling algorithm as the new Gate message will be transmitted only after reception of the previous Request message as shown in Fig. 1 (c). On the other hand, whenever a thread data cycle becomes large compared to the other threads, there is a problem of thread convergence as is shown in Fig. 1 (d). Also, due to thread convergence, the performance of the MT polling algorithm degrades to that of the ST polling algorithm. To prevent thread convergence in the MT polling algorithm, the threads are to be continuously tuned by shifting the load from some threads to the other threads. This leads to the complexity in the MT polling algorithm. Moreover, the packets in the moved bandwidth may encounter an increased delay. *This motivates us to make the cycle time fixed to overcome the problem of thread spread and convergence.*

Void formation (Issue 3) — The increase in the differential distances between ONUs and Max(RTT) in a LR-PON poses a problem of void formation. A void is the period in which the upstream channel is left unutilized. The MT polling algorithm is based on IPACT and thus, the thread scheduling cycle (i.e. the time period between issuing the same thread to an ONU, cf. Fig. 1 (e)) cannot be less than Max(RTT) of the PON. Thus, if the aggregation of all thread data cycles is less than Max(RTT), then it will lead to void formation. Fig. 1 (e) shows void formation in the MT polling algorithm. *This motivates us to determine the number of threads according to the network load which will minimize void formation to a large extent, and to over-grant an ONU whenever the number of used threads is smaller than the number of required threads.*

Computational Complexity (Issue 4) — In addition to the complexity added by thread tuning in the MT polling algorithm, the bandwidth reporting (BR) scheme also makes the algorithm complex. The traditional queue size reporting will lead to the problem of over-reporting and thus an ONU has to report its bandwidth according to the thread number on which the grant has come. Thus, there is more computational power needed at both the OLT and ONU as there is an added dimension associated with the thread number on which the Gate and Request message depend. The computational complexity of the protocol is increased by *n* times the complexity of IPACT where *n* is the number of threads. Inter-thread scheduling will further increase the complexity of the protocol [4]. *This motivates us to propose a bandwidth reporting scheme which solves the problem of over-reporting and in which the OLT and ONU do not have to keep track of the number of threads.*
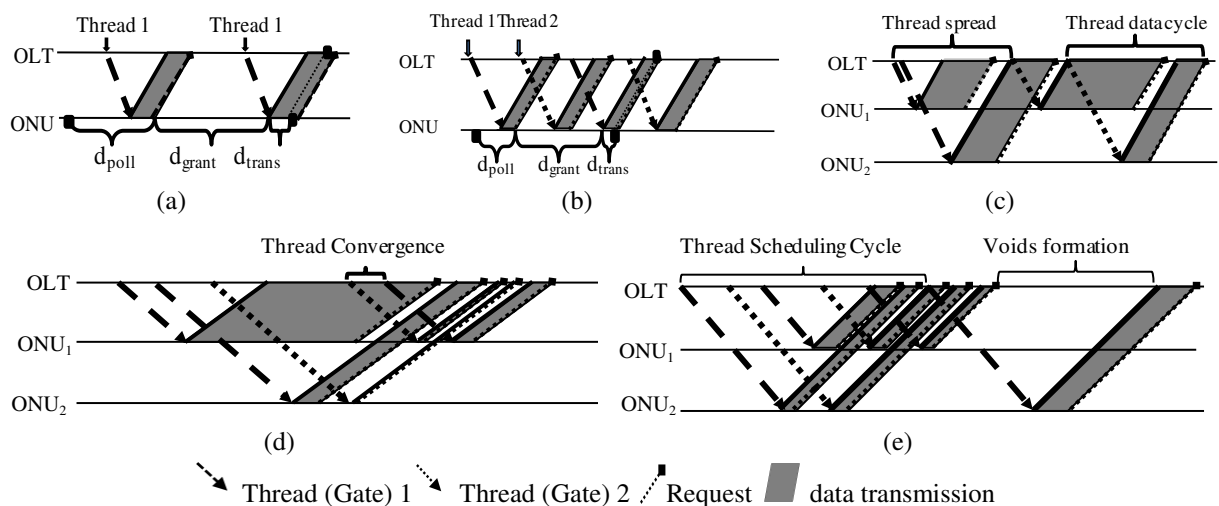

Figure 1: (a) Various delay components in traditional ST polling (b) Delay components in MT polling (with two threads) (c) The problem of thread spread in MT polling (d) The problem of thread convergence in MT polling (e) Formation of voids in MT polling.

3. ADAPTIVE MULTI-GATE POLLING WITH VOID-FILLING ALGORITHM

In our proposed AMGAV algorithm, we do adaptive multi-threading as more than one Gate may be issued to an ONU before the receipt of the acknowledgement of the previous Gate message and the number of Gate messages depends on the instantaneous network load.

Fixed cycle time (solves issue 2): The number of Gate messages is determined after a time interval referred to as a cycle. The time interval of a cycle depends on Max(RTT) and the number of Gate messages issued in the cycle. Generally,

$$T_{cycle}(i) = S_f \, \text{Max}(RTT) + N_o \, N_g(i) \, B \tag{2}$$

where $T_{cycle}(i)$ is the time period of the ith cycle, S_f is the scaling factor, N_o is the number of active ONUs, $N_g(i)$ is the number of Gate messages per ONU in the ith cycle, and B is the guard time between two adjacent time-slots. The scaling factor (S_f) depends on the reach of the PON and is a key parameter of the AMGAV algorithm. In a 20 km reach PON, where Max(RTT) is very small, the gain of threading is much reduced as compared to the longer reach scenario. Thus, with the reach of the PON, both the cycle time and the number of threads have to be properly scaled. For a given PON reach, S_f is fixed leading to the fixed cycle time independent of the load.

Void-filling and adaptive number of threads (solves issue 3 and 1, respectively): The algorithm adopts the void filling approach in which the size of the grants issued to the ONUs is such that the transmission slot of a complete cycle is utilized. For simplicity, we first consider a single-thread process between an ONU and OLT. Let us consider an example of how voids are created in traditional ST DBAs like IPACT. In Fig. 2 (a), two ONUs are considered for clarity. Let us assume that ONU_2 is having a RTT of 1 ms and ONU_1 and ONU_2 have requested for a transmission window of 200 µs each in the previous cycle. Since the next grant message for ONU_2 can only be issued after the receipt of the acknowledgement of the previous one, this leads to a void period of around 800 µs. In the first method of void-filling (cf. Fig. 2 (b)), we distribute the unrequested bandwidth between all ONUs (using constant or linear credit schemes [3]). Here, we try to utilize the otherwise wasted bandwidth and this gives us a significant improvement in the average packet delay. First, the report from the ONU is transmitted at the beginning of the transmission slot, already reducing the void formation with 200 µs in this example. The OLT then calculates the bandwidth wastage and allocates the unutilized bandwidth to both ONUs. Instead of only granting a 200 µs transmission window to each ONU, an extra transmission window of 300 µs will be allocated to each ONU so that the bandwidth wastage is minimized. However due to the bursty nature of the traffic, the bandwidth demand of each ONU may be entirely different. A better method of void-filling is to poll ONUs multiple times in a cycle. Thus, on receipt of a Request for a transmission window of 200 µs from both ONU_1 and ONU_2, the OLT will divide the cycle into smaller time-slots, called sub-cycles. E.g. in the present case, the cycle will be divided in three sub-cycles where each sub-cycle will be equal to a time-slot of 333 µs. The OLT will poll each ONU once in a sub-cycle. The number of Gate messages (or threads) issued to each ONU is equal to the number of sub-cycles and is formulated by (3),

$$N_g(i) = \min \left[\left\lceil \frac{S_f \, \text{Max}(RTT) R_u}{B_l(i)} \right\rceil , \text{Max}(N_g) \right] \tag{3}$$

where R_u is the upstream channel data rate, $B_l(i)$ is the sum of the Requests (in bytes) of all ONUs, and Max(N_g) is the maximum value of $N_g(i)$ for all cycles. Especially note that the number of Gate messages depends on the instantaneous network load which makes the algorithm adaptive. As at low load, however, the number of Gate messages may become very high, adversely affecting the downstream traffic, Max(N_g) is introduced. In a cycle, the time-slot for a sub-cycle is fixed. As illustrated in Fig. 2 (c), three sub-cycles are initiated at time t_o, t_1 and t_2, respectively. The time t_o denotes both the beginning of the new cycle and the first sub-cycle. At time-interval t_1, the second sub-cycle will be distributed fairly according to the latest demands of the ONU. In the present case, since there is no new arrival of a Request message between t_o and t_1, the time-slots are distributed as in the previous sub-cycle. During the start of the third sub-cycle, the Request message from ONU_2 has already arrived and now the OLT will distribute the time-slots window of the third sub-cycle according to the new Request statistics. With this algorithm, there is no void formation and at all time the upstream channel is better utilized. Moreover, by using multiple Gates in a cycle we achieve fairer statistical multiplexing.
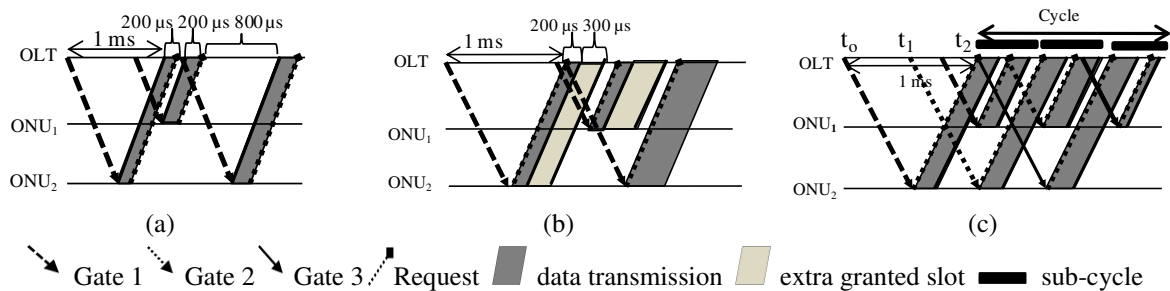

Figure 2: (a) Void formation in the ST polling algorithm (b) Removal of void formation by applying a void-filling algorithm (c) An example of application of the AMGAV algorithm

Bandwidth reporting scheme (solves issue 4): In the AMGAV algorithm, an ONU generates a Request message with multiple Requests to report the sum of the untransmitted bytes in the present cycle and the size (in bytes) of multiple fractions of newly arrived packets. The OLT will grant the window size choosing one of the multiple Requests such that it almost completely fills the transmission slot and there are no Request truncations (allocating smaller transmission slots than Requests). The use of multiple Requests is important for the AMGAV algorithm because the transmission slot of all ONUs has to be adjusted in a sub-cycle and thus otherwise there may be large Request truncations leading to bandwidth wastage in unused slot remainders (USRs). On the other hand, the OLT maintains a request variable (RV) for each ONU. The OLT increments the RV by the number of requested bytes in each thread and thus maintains track of all requested bytes. Similarly, on the issue of every Gate message, the OLT decrement the corresponding RV by the number of granted bytes. The proposed bandwidth reporting scheme requires the OLT to inform an ONU about any lost Request message.

4. SIMULATION RESULTS

We simulate an LR-PON with 16 ONUs where the distance between the OLT and ONUs varies from 10 to 100 km. We have chosen R_u as 1 Gbps, the maximum user data rate at an ONU as 100 Mbps, S_f as 2, Max(N_g) as 8, and B as 1 µs. We generate packets in the form of Ethernet frames (64 to1518 bytes). The synthetic user traffic is self-similar with a Hurst Parameter of 0.8 [5]. The buffer size at each ONU is limited to 1 MB. In Fig. 3 (a) and (b), the load is normalized to the upstream channel rate R_u (1 Gbps). When the load is not very high, such as 0-0.9, our algorithm shows a much better delay performance than ST or MT polling as shown in Fig. 3 (a). At high load, the delay is comparable to that of the MT polling algorithm. Fig. 3 (b) shows the channel utilization vs. network load. The improved channel utilization is due to the reduced USR. At a network load of 1, the channel utilization is 97.5% compared to 95% in the MT polling algorithm. 0.8% of bandwidth wastage is due to guard band overhead, about 0.4% due to USR. This is an improvement over the MT polling algorithm where the bandwidth wastage is about 2.9% due to USR formation.
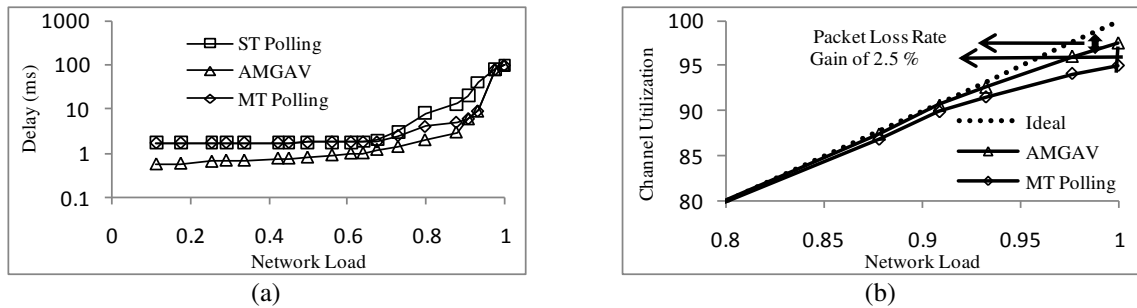

(a)   (b)

Figure 3: (a) Average packet delay for OLT-ONU distance of 100 km (b) Channel utilization vs. Network Load

5. CONCLUSIONS

In this paper, we proposed the Adaptive Multi-GAte polling with Void filling (AMGAV) algorithm, which explores the benefits of having multiple polling processes to schedule the upstream traffic simultaneously. The AMGAV algorithm not only improves the delay at high loads, but has a better performance over the whole range of loads. The average packet delay is significantly improved until a network load of 90%. At a higher network load, the average packet delays are comparable to the ones that were achieved in MT polling. The AMGAV algorithm is less complex and more efficient than the MT polling algorithm and is adaptive to the network load.

REFERENCES

[1]   G. Talli and P. D. Townsend: Feasibility demonstration of 100 km reach DWDM SuperPON with upstream bitrates of 2.5 Gb/s and 10Gb/s, in *Proc. OFC 2005*, Anaheim, CA, USA, Mar. 2005.
[2]   G. Das, *et al.*, "A New Architecture and MAC Protocol for Fully Flexible Hybrid WDM/TDM PON", in *Proc. ECOC 2009*, Vienna, Austria, Sep. 2009, paper P6.28.
[3]   G. Kramer, B. Mukherjee, and G. Pesavento: IPACT: A Dynamic Protocol for an Ethernet PON (EPON*), IEEE Communications Magazine*, vol. 40, no. 2, pp. 74-80, Feb. 2002.
[4]   H. Song, B.-W. Kim, and B. Mukherjee: Multi-thread polling: a dynamic bandwidth distribution scheme in long-reach PON, *IEEE JSAC*, vol. 27, no. 2, pp. 134-142, Feb. 2009.
[5]   W. Willinger, *et al.*: Self-similarity through high variability: statistical analysis of Ethernet LAN traffic at the source level, in *Proc. ACM SIGCOMM '95*, pp. 100-113, Cambridge, MA, USA, Aug. 1995.