Dynamic Reallocation of SLA Parameters in Passive Optical Network Based on Clustering Analysis

Nejm Eddine Frigui, Tayeb Lemlouma, Stéphane Gosselin, Benoit Radier, Renaud Le Meur, Jean-Marie Bonnin

Dynamic Reallocation of SLA Parameters in Passive Optical Network Based on Clustering Analysis

Nejm Eddine Frigui*, Tayeb Lemlouma†, Stéphane Gosselin*
Benoit Radier*, Renaud Le Meur*, and Jean-Marie Bonnin‡
*Orange Labs, Lannion, France
†University of Rennes 1 IRISA, Lannion, France
‡IMT Atlantique, Cesson Sévigné, France
nejm.frigui@orange.com, tayeb.lemlouma@irisa.fr, stephane.gosselin@orange.com
benoit.radier@orange.com, renaud.lemeur@orange.com, jm.bonnin@imt-atlantique.fr

***Abstract*—The introduction of new services as well as the growth in the number of communication terminals in the last years has led to an exponential growth of data traffic in both fixed and mobile networks. Passive Optical Networks (PONs) offer high bandwidth services to service providers customers. However, due to the dynamicity of users traffic patterns, PONs need to rely on an efficient upstream bandwidth allocation mechanism to define for each customer the amount of data that needs to be transmitted at a specific time. This mechanism is currently limited by the static nature of Service Level Agreement (SLA) parameters which can lead to an unoptimized bandwidth allocation in the network. In this paper, we propose a novel mechanism for optimizing the allocation of upstream Gigabit-capable Passive Optical Networks (GPON) resources based on the dynamic adjustment of some SLA parameters according to customer's estimated traffic patterns. Clustering analysis is used to differentiate customers according to their bandwidth utilization based on real-time and historical data. Three user classes are taken into account: heavy, light and flexible. Our work considers two fundamental clustering algorithms, namely K-means, a very well-known partitioning method and DBSCAN, one of the most common density-based clustering algorithms. An experimental study is conducted to evaluate the two algorithms and select which one can be the most suitable for the differentiation of user classes.**

***Index Terms*—Passive Optical Network (PON), Clustering Analysis, Service Level Agreement (SLA)**

I. Introduction

In the last years, the development of telecommunications networks and information technology has become increasingly fast. At the horizon 2020, we expect an exponential growth of data traffic in both fixed and mobile networks. This is mainly due to the widespread integration of Internet of Things, 5G networks, and the usage of high-speed services such as VoD, IPTV, video conferencing, and Ultra High Definition video services. Passive Optical Networks (PONs) offer high bandwidth services to operators' customers. The well known generation of ITU-T PON standards is the Gigabit-capable Passive Optical Networks (GPON) [1]. It was standardized for the first time in 2003 and provides 2.5 Gbit/s downstream and 1.25 Gbit/s upstream capacities while these rates can be shared up to 64 users depending on the splitter options used. As the demand for higher data speeds is increasing over the years, the control of the bandwidth allocation mechanism will be a key task in the network the capacity of which needs to be continiously utilized in a more efficient manner.

In our work, we propose a novel mechanism for optimizing the allocation of upstream GPON resources based on customer's estimated traffic patterns. Accordingly, we propose to dynamically adjust the Service Level Agreement (SLA) parameters as the only upstream resources allocation mechanism inputs that are allowed to be modified by the operator while maintaining the overall customer satisfaction. Traffic patterns will be identified through the use of clustering techniques based on real-time and historical data. We evaluate different clustering techniques in order to choose which one is more efficient for our case.

The paper is structured as follows. Section II describes the upstream bandwidth allocation mechanism in GPON. Section III presents some works related to the Dynamic Bandwidth allocation (DBA) mechanism. Section IV introduces the problem that needs to be adressed. In section V, we present our proposed model for optimizing the GPON upstream bandwidth allocation based on the dynamic adjustment of SLA parameters. In section VI, we present the clustering module in details as well as Python simulations to evaluate different clustering techniques applied to a real traffic trace captured on an operational FTTH network of Orange France. The obtained experimental results are then reported and analysed. Finally, we conclude our work in section VII.

II. Upstream Bandwidth Alloaction Mechanism

In PON networks, upstream bandwidth allocation is a control mechansim ensured by the Optical Line Terminal (OLT) which defines for each Optical Network Unit (ONU) the length and the time of transmission of each upstream burst. This mechanism can be also performed by the ONUs, so-called distributed bandwidth allocation. However, it is rarely known and not very common compared to the other one. The upstream bandwidth allocation can be achieved either in a static or dynamic way. Fig.1 shows the concepts of both Static Bandwidth Allocation (SBA) and Dynamic Bandwidth Allocation (DBA).

SBA has the advantage of being simple to implement at the OLT level. However, the risk of congestion or bottleneck in the

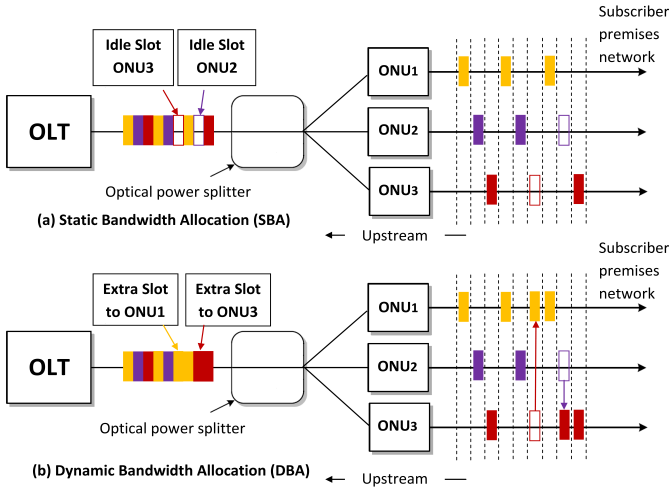

Fig. 1: Static Bandwidth Allocation (SBA) and Dynamic Bandwidth Allocation (DBA) mechanisms

network is high due to the possibility to have idle time slots that occupy the network. This can result in a degradation of the overall network performance and a customer dissatisfaction towards his service provider [2]. SBA can be classified as a QoS-unaware allocation mechanism and can be optimal only in the case where all network services require a constant bitrate. It should not be considered in the case of a shared-medium transmission system such as GPON.

As service providers need to adapt their network performances based on the dynamicity of users traffic patterns, a DBA mechanism is needed in order to make the best use of the unallocated bandwidth. DBA allows the bandwidth which is not consumed by some ONUs to be assigned to the other ones, resulting in a more flexible, efficient, and fair bandwidth allocation mechanism. ONUs can send their pending data only within the limits of the allocated grants which are specified in the Upstream Bandwidth Map (US-BWmap) field. These grants are transmitted by the OLT to the different ONUs in the downstream traffic frames [3]. They are updated regularly in each DBA cycle which represents the duration that covers all transmissions from all ONUs and may vary depending on the state of the ONUs packet queues [2]. GPON supports a mixture of services (e.g. VoIP, internet browsing, business services...) which do not have the same bandwidth requirements in the upstream (e.g. VoIP requires a constant bitrate). Thus, the DBA needs to take this into account in order to satisfy all customer needs. In this context, ONUs may support multiple classes of services. Each class has its own buffer with a specific quality of service (QoS) [4]. Therefore, the bandwidth allocation is no longer assured per ONU but rather per ONU per service. To guarantee the different QoS levels, GPON uses the Transmission Container (T-CONT) mechanism which allows different ONU's GPON Encapsulation Method (GEM) flows to be gathered in the same class of service [5][6][7]. Table I shows the five containers that can be assigned to one ONU depending on which services the latter had requested [8][9]. The activity status of each container can be obtained either through a status or non-status reporting mechanism. Fig.2 provides a simplified schematic of the Status Reporting (SR-DBA) mode [3][9].

TABLE I: T-CONT Types

T-CONT Type	Associated Services
Type 1	Fixed bandwidth services with a high priority Constant bit rate applications sensitive to jitter and delay
Type 2	Assured bandwidth services with a well defined bitrate Variable bit rate applications
Type 3	Assured bandwidth services similar to T-CONT2 Possibility to take advantage of non-assured bandwidth
Type 4	Best effort bandwidth services No specific requirement on QoS (no delay sensitivity) Served only when there is an available bandwidth
Type 5	Mixed type or a combination of two or more T-CONTs

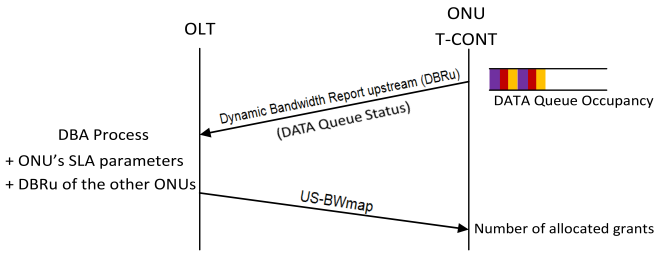

Fig. 2: Status Reporting (SR-DBA) mode

In the Non-Status Reporting (NSR-DBA) mode, the ONUs data queue occupancy information is not provided to the OLT. The latter should hinge on the actual transmission in the previous cycle and the information given by the GEM headers to estimate the ONUs queue status. In fact, when the ONU does not have data to send, idle frames are generated and sent during the allocation interval. These frames are observed by the OLT which understands that there is no transmission for the corresponding ONU and automatically decreases its bandwidth allocation in the following cycle [2][3][9].
The SR-DBA based algorithms are widely implemented in OLTs compared to the NSR-DBA ones. Thanks to their ability to reduce the latency (traffic can be granted for transmission once reported) and to avoid the overestimation or the underestimation of data queues, we will only consider the SR-DBA mechanism in the following.

SR-DBA algorithms are based on the combination of the queue occupancy information transmitted from each T-CONT depending on its activity status and the provisioned SLA of each T-CONT [9]. SLA parameters are defined in a service contract which represents a formal commitment between the provider and the subscriber. They define the specific objectives expected and the QoS that a client wishes to obtain from the service provider. The main parameters specified in the customer SLA are respectively the Committed Information Rate (CIR), the Excess Information Rate (EIR), and the Peak Information Rate (PIR) [3]. The CIR defines the guaranteed bandwidth provided by the network when delivering the data frames. The EIR specifies the additional bandwidth above that

of the CIR that a subscriber may use if available. The PIR represents the maximum bandwidth that can be provided. More details about SLA parameters are given in table II [5][8][10].

TABLE II: SLA Parameters

SLA Parameters			Significance
Peak Information Rate (PIR) PIR = CIR + EIR	Committed Information Rate (CIR) CIR = FIR + AIR	Fixed Information Rate (FIR)	∗Fixed bandwidth allocated to T-CONT type 1 ∗Untouchable parameter ∗Reserved cyclically regardless of demand
		Assured Information Rate (AIR)	∗Assured bandwidth allocated to T-CONTs type 2 and 3 ∗Not given without demand
	Excess Information Rate (EIR)		∗Allocated to T-CONTs type 3 and 4

III. Related Work to DBA Algorithms

As the DBA directly affects the performance of the network in the upstream, many works [11]–[15] have been proposed in the literature with the objective to enhance its operation. These works can be classified according to their ability to take or not into consideration the constraints agreed in the SLA contract, i.e. SLA awareness. In this paper, only the enhanced DBA proposals which respect the minimum and the maximum of bandwidth that must be provided to subscribers according to their SLA parameters are taken into account.

In [11], an SLA aware dynamic bandwidth allocation algorithm (SLA-DBA) was presented. In this algorithm, the OLT allocates in a first stage a number of bytes to an ONU queue depending on its bandwidth report. Then taking into account the SLA constraints, an adjustment will be performed in such a way that the number of assigned bytes never exceeds the maximum guaranteed rate provided by the SLA.

In [12], the proposal introduces an approach called cyclic-polling-based dynamic bandwidth allocation with service level agreements (CPBA-SLA) based on a two-layer allocation scheme: SLA-layer and ONU-layer. In the SLA-layer allocation, ONUs are distributed on two or more groups based on their SLA parameters. Then, the OLT assigns the bandwidth for each class of service in each group in a way that high-priority traffic can be served earlier. In the ONU-layer allocation, bandwidth is assigned to ONUs which belong to the same group where a group can be defined as a set of ONUs that require the same average packet delay of a specific class of service (e.g. high-priority, delay-sensitive traffic, etc.) for a specific cycle time.

Authors in [13] propose a fair bandwidth allocation algorithm based on network utility maximization. In addition to the requested bandwidth by each ONU in its report message and the guaranteed bandwidth coming from the SLA contract, the algorithm takes into account an additional factor that corresponds to the importance of an ONU regarding the conditions specified in its QoS requirements. Then, it distributes the extra bandwidth available in the network in a fair manner according to the definition of fairness in the network utility maximization model.

Although the majority of these works contributes to the optimization of the DBA mechanism, they do not take into account the fact that the DBA is a closed control protocol between the OLT and the ONUs in the GPON network [16][17]. Thus, modifying its operation is a hard task for an operator or a service provider who doesn't have the total control of this mechanism due to the dependency on a specific vendor. To resolve this issue and allow operators to optimize their networks without being bound to the constraints of specific vendor, it's recommended to focus only on the input parameters of the DBA algorithm which can be managed by the operator, i.e. SLA parameters. The idea is so, to try to optimize the management of these parameters in order to exploit in a more efficient manner the bandwidth available in the network, without modifying the implementation of the DBA mechanism itself as it was done in the majority of the reviewed works.

IV. Problem Statement

In our work, we propose a novel management procedure for optimizing the allocation of GPON resources based on the dynamic adjustment of the SLA parameters according to estimated customer traffic patterns. The latter will be identified through the use of clustering techniques based on real-time and historical data. The idea came from the fact that during the day, customers are not connected to the network in the same way. There are times when some of them need more than their PIRs, but they can not afford an extra bandwidth by the DBA due to the SLA parameters constraints. Thus, adjusting dynamically these parameters during the day can improve the QoS and maximize the overall satisfaction of users. To summarize, the problem can be formulated as follows:

- **Given:** The initial SLA parameters and the activity status of each ONU based on real-time and historical transmission data
- **Objective:** Optimize the upstream resource allocation by adjusting the SLA parameters depending on the customer bandwidth usage
- **Output:** New SLA parameters for each ONU at a specific period of time

V. The Proposed Model

To achieve the main objective introduced in the previous section, we present a new architecture for optimizing the upstream bandwidth allocation in Fig.3. To the best of our knowledge, this is the first time, an optimization approach based on the dynamic adjustment of SLA parameters and the customer traffic behaviour during the day is applied to PON networks. The proposed architecture consists of four main modules: *Monitored Data Collector*, *Clustering Module*, *Assignment Index Calculator*, and a mechanism for *Reallocating SLA Parameters*. The operation of each module is explained in the following.

A. Monitored Data Collector

In this module, the Initial SLA parameters are retrieved from the Management Information Base (MIB) only once for each ONU and then stored in a specific table. Monitored traffic data

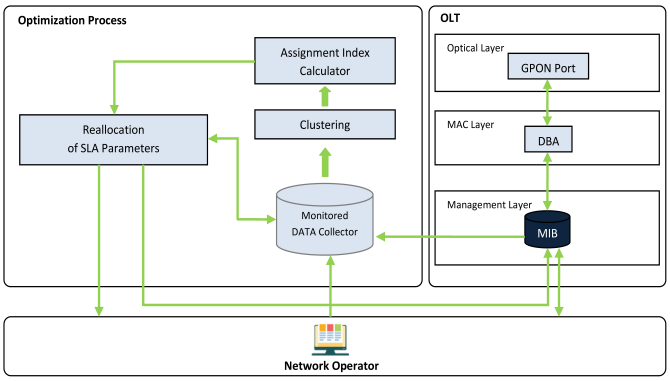

Fig. 3: Proposed architecture for the dynamic adjustment of SLA parameters

for each ONU is collected from the MIB at regular intervals that can be parameterized by the network operator. It will be then stored as well as the historical transmission data which will be recovered from the network operator side.

B. Clustering Module

As the traffic utilization of GPON users may change several times per day since traffic patterns are linked to daily life, defining different time periods of the day when users have a similar traffic utilization can be very useful in the optimization of PON resources allocation. In this case we will refer to a clustering module to classify subscribers according to their online behaviour during the day. The main objective is to identify which users consume a lot of bandwidth (heavy users) and try to maximize their satisfaction in terms of QoS by giving them a surplus of bitrate on their EIRs. Consequently, for all day periods, the clustering will identify clusters of heavy users with a high upstream transmission bitrate and clusters which contain light users with a low upstream bitrate. The input of this module is the historical transmission data collected over some time horizon. The collection is achieved using a time series format with the average bitrates for each ONU over a parameterizable monitoring interval (e.g. 5min). The clustering process will be executed for the whole OLT in order to have as many as possible of data points. Table III presents an example of decomposing each day into four time slots, in each time slot, three classes of users can be distinguished: heavy, light, and flexible users.

TABLE III: Clustering Explanation

Day Period	Time Slot	User Traffic Classes	Cluster Index
Morning	6 a.m. to 12 p.m.	Heavy Users	C_{M_H}
		Light Users	C_{M_L}
		Flexible Users	C_{M_F}
Afternoon	12 p.m. to 6 p.m.	Heavy Users	C_{A_H}
		Light Users	C_{A_L}
		Flexible Users	C_{A_F}
Evening	6 p.m. to 11 p.m.	Heavy Users	C_{E_H}
		Light Users	C_{E_L}
		Flexible Users	C_{E_F}
Night	11 p.m. to 6 a.m.	Heavy Users	C_{N_H}
		Light Users	C_{N_L}
		Flexible Users	C_{N_F}

C. Assignment Index Calculator

As the results of the clustering module are provided according to the chosen monitored time interval, we introduce an Assignment Index (AI) to classify the user's traffic by day period. This index ranges from 0 to 1 and represents the degree of attachment of one ONU to a certain traffic class in a specific day period. For instance, if during a day period, a user has been classified as heavy 5 times over 10 time intervals, then its AI to the heavy users class will be 0.5. The mathematical formulation for this calculation process is detailed in Algorithm 1 while the notations are provided in Table IV.

TABLE IV: Table of Notations

Notation	Meaning
U_i	User index, $i \in [1, n]$, with n:total number of users
t	Interval of time in a day period, e.g. with a 5 minutes step
D	Number of supervised days
$Days$	The set of supervised days
DP	The set of day periods (e.g. morning, afternoon, evening, and night)
C_{t_H}	The cluster which contains heavy users for a specific interval t
C_{t_L}	The cluster which contains light users for a specific interval t
$AI^d_{C_{j\_H}}(U_i)$	AI of U_i to the heavy users cluster in the day period j of the day d
$AI^d_{C_{j\_L}}(U_i)$	AI of U_i to the light users cluster in the day period j of the day d
$CIR(U_i)$	Committed Information Rate of U_i
$EIR(U_i)$	Excess Information Rate of U_i
$BR^{avg}_t(U_i)$	Mean upstream bitrate of U_i over a specific interval t
$Sum^L_{EIR_T}$	The sum of theoretical EIRs of light users
$Sum^H_{EIR_P}$	The sum of provided EIRs to heavy users
$ExBw^{avg}_{j-d}$	The average of the extra bandwidth by day period for a specific day

Algorithm 1 Assignment Index Calculation

for d **in** $Days$ **do**
  for j **in** DP **do**
    for each U_i **do**
      $nbItr \leftarrow 0$ # $nbItr$: number of time intervals in a day period j
      $x \leftarrow 0$ # x: number of times the U_i was considered as heavy
      $y \leftarrow 0$ # y: number of times the U_i was considered as light
      for t **in** j **do**
        if $U_i \in C_{t\_H}$ **then**
          $x \leftarrow x + 1$
        end if
        if $U_i \in C_{t\_L}$ **then**
          $y \leftarrow y + 1$
        end if
        $nbItr \leftarrow nbItr + 1$
      end for
      $AI^d_{C_{j\_H}}(U_i) \leftarrow \frac{x}{nbItr}$
      $AI^d_{C_{j\_L}}(U_i) \leftarrow \frac{y}{nbItr}$
    end for
  end for
end for

In a first phase, the AI is computed upon the results of the clustering analysis. The latter is performed on the historical data provided by the service provider over a specified time horizon. To get a more accurate index, it is preferable to have a longer monitoring period as well as time intervals with smaller steps in a day period. As the index is calculated per day and per day period, we calculate the average of all daily AIs per day period for each user:

$$\begin{cases} AI^{avg}_{C_{j\_H}}(U_i) = \frac{\sum_{d \in Days} AI^{d}_{C_{j\_H}}(U_i)}{D} \; \forall j \in DP, \forall i \in [1,n] \\ AI^{avg}_{C_{j\_L}}(U_i) = \frac{\sum_{d \in Days} AI^{d}_{C_{j\_L}}(U_i)}{D} \; \forall j \in DP, \forall i \in [1,n] \end{cases}$$

Since the $AI^{avg}_{C_{j\_H}}(U_i)$ and $AI^{avg}_{C_{j\_L}}(U_i)$ may not be too meaningful in the case where the different indexes are very dispersed, an additional step consisting in calculating the standard deviation will be strongly recommended to decide whether or not the mean value can be taken into account:

$$\begin{cases} AI^{sd}_{C_{j\_H}}(U_i) = \sqrt{\frac{1}{D} \sum_{d \in Days} (AI^{d}_{C_{j\_H}}(U_i) - AI^{avg}_{C_{j\_H}}(U_i))^2} \\ AI^{sd}_{C_{j\_L}}(U_i) = \sqrt{\frac{1}{D} \sum_{d \in Days} (AI^{d}_{C_{j\_L}}(U_i) - AI^{avg}_{C_{j\_L}}(U_i))^2} \\ \forall j \in DP, \forall i \in [1,n] \end{cases}$$

We propose that once the $AI^{sd}_{C_{j\_H}}(U_i)$ and $AI^{sd}_{C_{j\_L}}(U_i)$ reach a certain threshold AI^{sd}_{TH}, the $AI^{avg}_{C_{j\_H}}(U_i)$ and $AI^{avg}_{C_{j\_L}}(U_i)$ will not be taken into account and hence the U_i will be assigned to flexible user clusters (i.e. C_{j_F}). The AI^{sd}_{TH} value should be as close as possible to 0 to assert that the calculated averages are significant. Hence, the assignment index of a user during a long period of time can be a good criterion to classify it as heavy or light for a specific day period. Finally, to decide to which cluster the user belongs based on its mean AI, we suppose that:

$$\begin{cases} U_i \in C_{j\_H} \;\; if \; AI^{avg}_{C_{j\_H}}(U_i) \geq 0.75 \; \forall j \in DP, \forall i \in [1,n] \\ U_i \in C_{j\_L} \;\; if \; AI^{avg}_{C_{j\_L}}(U_i) \geq 0.75 \; \forall j \in DP, \forall i \in [1,n] \\ U_i \in C_{j\_F} \;\; Otherwise \; \forall j \in DP, \forall i \in [1,n] \end{cases}$$

To improve the accuracy of the AI process, we continuously consider the results of the clustering module. This is achieved based on the new data collected by the *Monitored Data Collector* module which is regularly updated. As a first step, we consider a weekly update. The AI is used as a reference in the reallocation of the SLA parameters in the network and the determination of the additional bandwidth that can be provided to heavy users.

D. Reallocation of SLA Parameters

The main objective of the reallocation of SLA parameters in the network is to try to maximize as well as possible the satisfaction of all users in a specific day period. Ordinarily, the DBA assigns to each ONU the amount of data to be transferred regarding its packet queue status and the whole state of the network. The problem is when there is a majority of light or offline users in the PON, the excess available bandwidth to be assigned by the DBA to heavy users (i.e. to their EIRs) may exceed their PIRs. Thus, heavy users do not take advantage of a part of the additional bandwidth due to their SLA parameters constraints. The idea is therefore to extend the EIRs for this class of users in such a way they can benefit from a bandwidth supplement whenever possible. We propose a calculation process of this extra bandwidth in Algorithm 2. This algorithm is performed by PON port, i.e. the considered users (whether light or heavy) are all part of the same PON port. We consider that offline users are part of light users cluster. Notations are provided in Table IV.

Algorithm 2 Extra Bandwidth Calculation

for d **in** $Days$ **do**
  for j **in** DP **do**
    $Sum^{L}_{EIR_T} \leftarrow 0, ExBw \leftarrow 0$
    for U_i **in** C_{j_L} **do**
      $Sum^{L}_{EIR_T} \leftarrow Sum^{L}_{EIR_T} + EIR(U_i)$
    end for
    $nbItr \leftarrow 0$ # $nbItr$: number of time intervals in a day period j
    for t **in** j **do**
      $Sum^{H}_{EIR_P} \leftarrow 0$
      for U_i **in** C_{j_H} **do**
        $Sum^{H}_{EIR_P} \leftarrow Sum^{H}_{EIR_P} + (BR^{avg}_{t}(U_i) - CIR(U_i))$
      end for
      if $Sum^{L}_{EIR_T} > Sum^{H}_{EIR_P}$ **then**
        $ExBw \leftarrow ExBw + (Sum^{L}_{EIR_T} - Sum^{H}_{EIR_P})$
      end if
      $nbItr \leftarrow nbItr + 1$
    end for
    $ExBw^{avg}_{j\_d} \leftarrow \frac{ExBw}{nbItr}$
  end for
end for
for j **in** DP **do**
  $ExBw^{avg}_{j} \leftarrow \frac{\sum_{d \in Days} ExBw^{avg}_{j\_d}}{D}$
end for

The resulted extra bandwidth $ExBw^{avg}_{j}$ is calculated by day period and represents the average of all extra bandwidths over a specific number of days. It will be shared between the heavy users according two possible options: either a fair sharing where the additional bandwidth is added to all ONUs in the same way, or a prioritization of some ONUs on others. In this work, we apply a fair sharing where all heavy users will have the same amount of $ExBw^{avg}_{j}$ in a specified day period and so, the share of each user will be added to its EIR parameter. The new configuration of the SLA parameters will be then planned to be executed at a specific time. The *Monitored DATA Collector* is informed by the new SLA parameters and stored them in the same table as the initial parameters. In the event that a problem has occurred, a rapid return to the initial configuration can be performed. In the remainder of this paper, we will focus on the clustering module. The objective is to study and evaluate the different clustering mechanisms that exist in the literature in order to select which one can be the most suitable for our usecase.

VI. Clustering Module

The easiest way to classify subscribers according to their online traffic behaviour over a time period is to look at the differences between the traffic data of each user in order to create common groups based on the traffic similarity. If we specify for heavy and light users the minimum (or maximum) threshold that each subscriber shall reach to be assigned to a specific class, then this approach can be the most suitable to be adopted. However, in PON networks, the usage segments can be very varied since all heavy or light users do not have the same behavior in terms of upstream data bitrate. Thus,

there is no exact definition for these two classes of users. To overcome these limitations, we will refer to the clustering analysis approach which identifies the attachement of a user to a specific group without any prior assumption. Clustering analysis is a very common approach in the field of data analysis and related disciplines and can be very useful in the case of grouping customers into different classes regarding their traffic data. Given a specific data set, the objective of clustering is to find segments where objects are similar based on a measure of similarity (e.g. size, density, shape, etc.) [18].

A. Clustering Algorithms

As many clustering algorithms exist in the literature and several categorization criteria can be taken into account, it is difficult to consider all of them in this work. For this, we will focus on two major fundamental clustering approaches that can be applied in our usecase which are respectively, partitioning methods and density-based ones [19]–[22].

1) Partitioning Methods: Partitioning methods represent the most simplest and fundamental way to group points in clustering analysis. They take a set of n data points and try to construct k clusters ($k \leq n$) while each data point belongs to exactly one cluster. The metric used to differentiate clusters in most of partitioning methods is the distance between the objects. Therefore, objects that belong to the same cluster are probably close [19]. The most well-known partitioning algorithm in the literature is the K-means. Even if it was proposed a long time ago, K-means [23] remains one of the most used partitioning algorithms by researchers thanks to its simplicity and ease of implementation. The K-means algorithm performs as follows. Given a data set D containing n objects and k the number of clusters, the algorithm selects arbitrarily k points from D called cluster centroids. Then, each remaining point in D will be assigned to one of the k clusters in such a way that the Euclidean Distance between the points and the centroids is minimized. This process will be repeated for several times and in each time, the centroids values will be updated based on the objects that belong to the corresponding cluster. Then, all objects will be re-assigned taken into account the new cluster centers. Once the centroids values and the objects assignments are immutable, the algorithm process is stopped and clusters are formed [18].

2) Density-Based Methods: Density based methods represent clustering approaches that divide a set of data points into dense regions separated by scattered ones. The capacity of a given cluster continues to increase as long as the number of objects, i.e. density, in the "neighborhood" reachs some threshold [19]. One of the most common and cited density-based clustering algorithms is DBSCAN (Density Based Spatial Clustering of Applications with Noise). It was proposed for the first time in 1996 [24]. The idea of DBSCAN is to find core data points which have dense neighborhoods in order to connect them and form the corresponding clusters. Unlike partitioning algorithms, DBSCAN does not require any specification regarding the number of clusters. Only two parameters need to be taken into account: the maximum radius of each cluster "ϵ" and the smallest number of cluster points "$MinPts$" which represents the density threshold of a dense region [25]. The process of the DBSCAN algorithm is as follows. Given a data set D, ϵ and $MinPts$, all points are marked as unvisited. The algorithm selects a randomly unvisited point P and marks it as visited. Then it calculates the number of points within the $\epsilon - neighborhood$ of P ($\epsilon - neighborhood = \{Q \in D/dist(P,Q) \leq \epsilon\}$). If the number of points is larger than $MinPts$, a new cluster is created containing all identified points. Else, P is considered as noise point and does not belong to any cluster. The same process will be repeated until all points are assigned to a cluster or considered as noise [24][26].

B. Experimental Results

In order to analyse and evaluate the clustering algorithms presented in section VI and choose which one is more suitable for our use case, we rely on a real traffic trace collected based on a tool developed within Orange France called *OTARIE*. This probe is based on a software written in C that works on various PC-based hardware architectures equipped with a DAG (Data Acquisition and Generation) traffic capture card which has an API that allows to read the packets as they arrive on the network interface. The traffic traces that we dispose are collected for 3447 ONUs belonging to the same OLT equipment over a period of two weeks between the 7th and the 20th of November 2016. However, they do not cover the whole day. For this, we have chosen to work on a specific period of the day between 9p.m and 12a.m with a 5 minutes interval since the clustering approach becomes very advantageous in the case of busy hours when a significant number of subscribers are online. Due to space constraints, we present the results only for the day of November 14, 2016. Fig.4 shows the distribution of the different users by common slice of mean upstream bitrate. For a better representation, we limited the time axis scale between 9:05p.m and 10:05p.m.

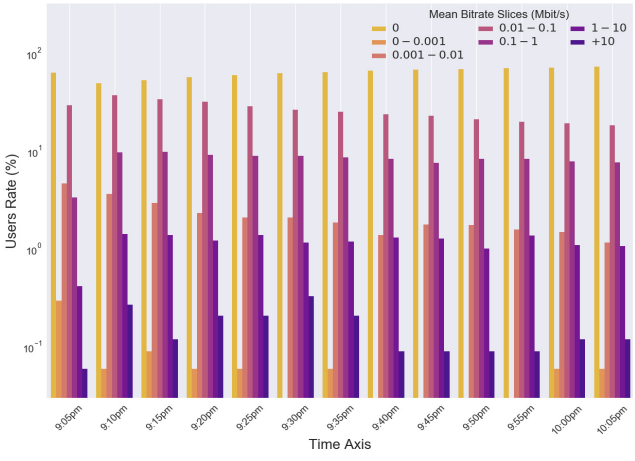

Fig. 4: Distribution of users per mean upstream bitrate slices

In the following subsections, the results of the application of the K-means and DBSCAN algorithms to our dataset are

presented. The accomplishment of the experiment relies on the use of the sklearn library [27] provided for Python developers. Matplotlib and seaborn modules have been used for plotting the different points and clusters in the most convenient way.

1) K-means: As mentionned in section VI-A1, the K-means algorithm takes as input the number of possible clusters k. In our dataset, k represents the number of user classes per a parameterized interval of time t (see Table IV). We would expect that for each time interval t, a maximum of 3 user classes may exist: heavy, light and the rest of users. Therefore, we evaluated the K-means algorithm with $k = 3$ for each time interval. Two metrics were used to calculate the Euclidean Distance used by K-means to differentiate user classes: the first consists in directly calculating the difference between two ONUs mean upstream bitrates. The second applies the decimal logarithm on the ONUs mean upstream bitrates. The results are presented respectively in Fig.5 and Fig.6. In all intervals, the three requested clusters are visualized. In Fig.5, the K-means algorithm has assigned the majority of users with a very low mean upstream bitrate to the same cluster. In Fig.6, the use of the decimal logarithm metric allows to have a more balanced distribution between the different clusters.

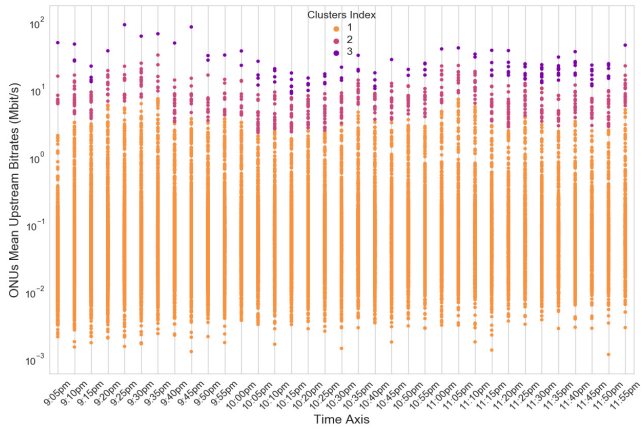

Fig. 5: K-means results (k=3, metric: ONUs mean upstream bitrates)

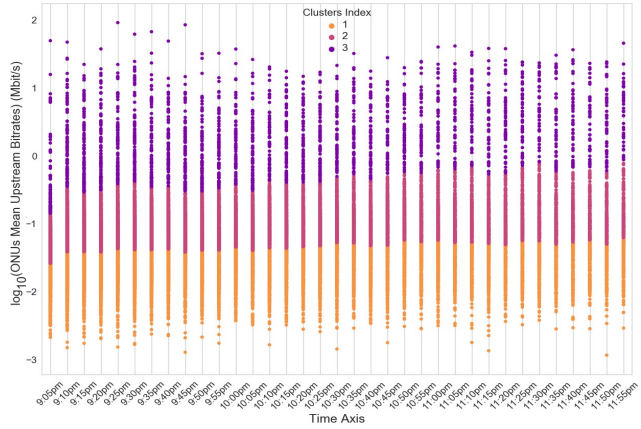

Fig. 6: K-means results (k=3, metric: decimal logarithm of ONUs mean upstream bitrates)

2) DBSCAN: Unlike K-means, DBSCAN does not have any requirements regarding the specification of the number of clusters before analyzing the dataset. However, the most difficult task lies in how to choose the ϵ and $MinPts$ parameters discussed in section VI-A2. We assume that a user class can contain at least one user (heavy or light), that is, even if only one user has been assigned to a group, the DBSCAN algorithm must take it into account and not consider it as noise. It is possible to have a minority of users who are online during a day period. If we add that the final assignment of a user to a specific cluster depends on the calculation of his AI (i.e. the $MinPts$ value does not reflect the final percentage of heavy or light users), we can propose as a first hypothesis that $MinPts$=1. Regarding ϵ, we tried to vary it in order to choose which value can produce the best clustering results. In our case, ϵ can be understood as the gap between two user classes. Thus, the wider the gap, the easier the interpretation of the clusters. The tested values range from 0.1 Mbit/s to 20 Mbit/s. When the ϵ value decreases, more clusters are formed by the DBSCAN algorithm. The number of resulted clusters is a very important criterion to take into consideration in traffic patterns classification. Indeed, the higher it is, the more complicated the interpretation will be. Between the different evaluated ϵ values, we have chosen to show the results for ϵ=10 Mbit/s (see Fig.7) since the number of the identified clusters is the same as the number of clusters initialized to K-means (i.e. $k = 3$).

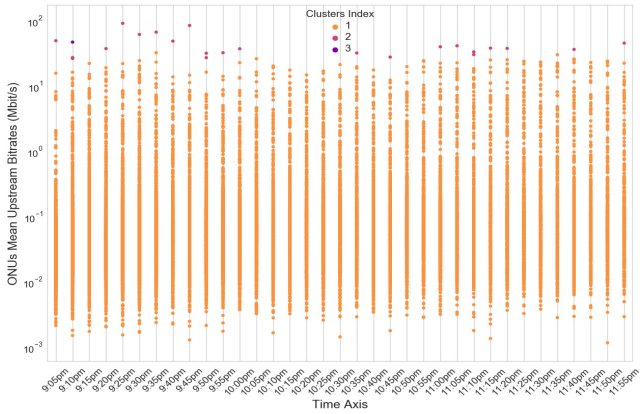

Fig. 7: DBSCAN results (ϵ=10, $MinPts = 1$)

C. Discussion

The evaluation of the K-means and DBSCAN algorithms allows us to visualize and differentiate users according to their traffic patterns. DBSCAN, thanks to its concept based on the differentiation of points according to their density, makes it possible to determine heavy and light user classes. However, it has a strong tendency to identify only a very small number of heavy users compared to the number of light ones using a 5 minutes time interval. This may seem as a major drawback for this algorithm in a policy of maximizing satisfaction for all network users. The other problem with DBSCAN is that we can fall in a case where one of the identified clusters

contains heavy and light users in the same time. This case can occur when there are no differences between the densities of points in one time interval in such a way that the epsilon parameter has no effect (e.g. the interval of 10:05 pm in Fig.7). Concerning K-Means, this algorithm gives a much more balanced distribution in terms of users number between heavy and light. The use of the decimal logarithm metric allowed to have a better classification results compared to the other metric. Nonetheless, K-means has the disadvantage of depending on the initial positions of different centroids and the need to specify for it the possible number of clusters before each process.

VII. Conclusion And Future Work

A new mechanism for reallocating SLA parameters based on users traffic patterns in a PON network has been proposed in this paper. A particular attention has been granted to the clustering module which allows to classify subscribers according to their hourly online behaviour during the day in order to have clusters that contain heavy users who have a high usptream transmission bitrate and clusters which contain light users with a low transmission upstream bitrate. Two clustering approaches have been evaluated in this paper, namely K-means and DBSCAN algorithms. The analysis that we have conducted is based on the ability of each algorithm to differentiate or not the PON users according to their mean upstream bitrates by time interval in a day period. The results showed that the two algorithms can meet this requirement. However, K-means based on the decimal logarithm metric shows a better classification results compared to DBSCAN in terms of equitable distribution between heavy and light users. This can be an important point in terms of maximizing the satisfaction of all users. In a future work, we expect to continue the investigations on the presented and the other clustering algorithms (e.g., hierarchical clustering) in order to select the most efficient tool for classifying PON users. Given that the traffic traces that we dispose are collected over a limited period of two weeks, we plan to complete the analysis of our approach using a network simulator. This will allow us to have more visibility and to evaluate the whole mechanism while taking into account several QoS parameters.

References

[1] ITU-T G.984.1, "Gigabit-capable Passive Optical Networks (G-PON): General characteristics," March 2008.

[2] O. Haran and A. Sheffer, "The importance of dynamic bandwidth allocation in GPON networks," *PMC-Sierra inc., White paper*, no. 1, 2008.

[3] M. Feknous, A. Gravey, B. L. Guyader, and S. Gosselin, "Status reporting versus non status reporting dynamic bandwidth allocation," in *2015 6th International Conference on the Network of the Future (NOF)*, Sept 2015, pp. 1–7.

[4] A. Walid and A. Chen, "Self-adaptive dynamic bandwidth allocation for GPON," *Bell Labs Technical Journal*, vol. 15, no. 3, pp. 131–139, Dec 2010.

[5] ITU-T G.984.3, "Gigabit-capable Passive Optical Networks (G-PON): Transmission convergence layer specification," January 2014.

[6] J. Ozimkiewicz, S. Ruepp, L. Dittmann, H. Wessing, and S. Smolorz, "Evaluation of Dynamic Bandwidth Allocation Algorithms in GPON Networks," *WSEAS Trans. Cir. and Sys.*, vol. 9, no. 2, pp. 111–120, Feb. 2010.

[7] J. Ozimkiewiez, S. R. Ruepp, L. Dittmann, H. Wessing, and S. Smolorz, "Dynamic bandwidth allocation in GPON networks," in *4th WSEAS international conference on Circuits, systems, signal and telecommunications*, 2010, pp. 182–187.

[8] M. Feknous, B. Le Guyader, P. Varga, A. Gravey, S. Gosselin, and J. A. T. Gijon, "Multi-criteria comparison between legacy and next generation point of presence broadband network architectures," *Advances in Computer Science: an International Journal*, vol. 4, no. 3, pp. 126–140, 2015.

[9] H. Bang, S. Kim, D.-S. Lee, and C.-S. Park, "Dynamic bandwidth allocation method for high link utilization to support NSR ONUs in GPON," in *2010 The 12th International Conference on Advanced Communication Technology (ICACT)*, vol. 1, Feb 2010, pp. 884–889.

[10] I. Loumiotis, E. Adamopoulou, K. Demestichas, and M. Theologou, *Optimal Backhaul Resource Management in Wireless-Optical Converged Networks*. Cham: Springer International Publishing, 2015, pp. 254–261.

[11] D. Nowak, P. Perry, and J. Murphy, "Bandwidth allocation for service level agreement aware Ethernet passive optical networks," in *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, vol. 3, Nov 2004, pp. 1953–1957 Vol.3.

[12] S. i. Choi and J. Park, "SLA-Aware Dynamic Bandwidth Allocation for QoS in EPONs," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 2, no. 9, pp. 773–781, September 2010.

[13] N. Merayo, P. Pavon-Marino, J. C. Aguado, R. J. Durn, F. Burrull, and V. Bueno-Delgado, "Fair bandwidth allocation algorithm for PONS based on network utility maximization," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 1, pp. 75–86, Jan 2017.

[14] B. Skubic, J. Chen, J. Ahmed, L. Wosinska, and B. Mukherjee, "A comparison of dynamic bandwidth allocation for EPON, GPON, and next-generation TDM PON," *IEEE Communications Magazine*, vol. 47, no. 3, pp. S40–S48, March 2009.

[15] C. M. Assi, Y. Ye, S. Dixit, and M. A. Ali, "Dynamic bandwidth allocation for quality-of-service over Ethernet PONs," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 9, pp. 1467–1477, Nov 2003.

[16] E. Dai and W. Dai, "Towards SDN for optical access networks," in *Spring Technical Forum Proceedings*, 2016, pp. 1–6.

[17] H. Woesner and D. Fritzsche, "SDN and OpenFlow for converged access/aggregation networks," in *2013 Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, March 2013, pp. 1–3.

[18] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[19] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[20] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[21] M. Zaït and H. Messatfa, "A comparative study of clustering methods," *Future Generation Computer Systems*, vol. 13, no. 2-3, pp. 149–159, 1997.

[22] I. A. Venkatkumar and S. J. K. Shardaben, "Comparative study of data mining clustering algorithms," in *2016 International Conference on Data Science and Engineering (ICDSE)*, Aug 2016, pp. 1–7.

[23] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA., 1967, pp. 281–297.

[24] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[25] C. Lu, Y. Shi, Y. Chen, S. Bao, and L. Tang, "Data mining applied to oil well using K-Means and DBSCAN," in *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, Nov 2016, pp. 37–40.

[26] I. V. Anikin and R. M. Gazimov, "Privacy preserving DBSCAN clustering algorithm for vertically partitioned data in distributed systems," in *2017 International Siberian Conference on Control and Communications (SIBCON)*, June 2017, pp. 1–4.

[27] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.