An Application-Oriented Fair Resource Allocation Scheme for EPON

Jingjing Zhang, *Student Member, IEEE*, and Nirwan Ansari, *Fellow, IEEE*

***Abstract*—This paper investigates an application-oriented bandwidth allocation scheme to ensure fairness among queues with diversified quality-of-serice (QoS) requirements in EPONs. Formerly, differentiated services (DiffServ) were suggested to be used in EPON so as to provision some queues with higher QoS over others. However, owing to the coarse granularity, DiffServ can hardly facilitate any particular QoS profile of an application in EPONs. In this paper, we define application utilities to quantify users' quality-of-experience (QoE) as a function of network layer QoS metrics. Then, we formulate the fair resource allocation issue into a utility max-min optimization problem, which is quasiconcave over queues' delayed traffic and dropped traffic. Utilizing the quasiconvex property, we propose to employ the bisection method to solve the optimization problem. The optimal value can be achieved by proper bandwidth allocation and queue management in EPONs. Detailed implementation of the proposed algorithm is discussed, and simulation results show that our proposed scheme can ensure fairness and guarantee QoS with fine granularity.**

***Index Terms*—EPON, fairness, optimization, quality-of-experience, quality-of-service, utility.**

I. INTRODUCTION

DIFFERENTIATED service (DiffServ) was proposed to be employed in access networks for quality-of-service (QoS) provisioning. Specifically, DiffServ classifies the incoming traffic into three classes: expedited forwarding (EF), assured forwarding (AF), and best effort (BE). EF is applicable to delay sensitive applications that require a bounded end-to-end delay and jitter specifications; AF is tailored for services that are not delay sensitive but require bandwidth guarantees; BE is not delay sensitive and has no minimum guaranteed bandwidth. The coarse granularity of DiffServ can hardly meet any particular QoS requirement imposed by various applications. However, the future access network will witness the sprouting of new applications, such as IPTV, video conference, telemedicine, immersing interactive learning, and large file transfer among computing and data-handling infrastructures (e-science). These newly emerging applications impose different QoS requirements as compared to those demanded by traditional video, voice, and data traffic. For example, large file transfer among e-science computing sites, on one hand, has strict throughput requirements, and hence possesses higher priority over traditional data traffic. On the other hand, traffic generated from these applications is not delay sensitive as compared to voice and video traffic. It is inappropriate to map these traffic into any of the three traffic classes in DiffServ. Inappropriate QoS mapping leads to either QoS over-provisioning or QoS under-provisioning. These diversified QoS requirements of applications pose great challenges on resource allocations in access networks.

This paper focuses on efficient QoS provisioning for queues with diversified QoS requirements in Ethernet Passive Optical Networks (EPONs), which have gained popularity among the access network technologies for their low costs, high bandwidth provisioning, and easy implementation. IEEE802.3ah standardized Multi-Point Control Protocol (MPCP) as a MAC layer control protocol for EPON. Specifically, MPCP defines two 64-byte control messages REPORT and GATE for the bandwidth arbitration in the upstream. Optical Network Units (ONUs) report their backlogged traffic to Optical Line Terminal (OLT) by sending REPORT. After collecting REPORT from ONUs, OLT dynamically allocates bandwidth to ONUs and informs its grant decisions to ONUs via GATE. Dynamic bandwidth allocation (DBA) has two major functions. One is to arbitrate bandwidth allocation among queues within the same ONU, referred to as intra-ONU scheduling. Another one is to arbitrate bandwidth allocation among different ONUs, referred to as inter-ONU scheduling. IEEE802.3ah does not specify any DBA algorithms for EPON. Formerly, many DBA algorithms have been proposed [2]–[7]; besides provisioning QoS guarantees, ensuring fairness among queues and ONUs is regarded as another important objective of DBA algorithms.

Generally, ensuring fairness among queues with diversified QoS requirements is equivalent to addressing the following problem: ***under the heavy-load scenario, which of the queues' performance should be sacrificed and at what degree?***

In order to facilitate QoS profiles for any application, we first adopt the concept of *application utility* to quantify users' quality-of-experience (QoE) as a function of received QoS of the specific application [1], [8], [9]. Specifically, application utility depends on the relationship between QoE and network-level QoS performances of the specific application. Large utility corresponds to high degree of user satisfaction at the user-level, which consequently translates to high QoS performances at the network-level.

To ensure fairness among queues, we treat maximizing the minimum application utility as the DBA objective, and formulate the problem of ensuring fairness among queues of

---

Manuscript received February 03, 2010; revised July 07, 2010; accepted July 07, 2010. Date of publication November 09, 2010; date of current version November 24, 2010. The preliminary version of this work was presented at ICST AccessNets 2009 [1].

The authors are with Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, University Heights, Newark, NJ 07102 USA (e-mail: jz58@njit.edu; nirwan.ansari@njit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSYST.2010.2082210

diversified QoS requirements as a utility max-min optimization problem. From the optimization point of view, the utility max-min problem with the single-objective of maximizing the minimum utility is a scalarization of the multi-objective optimization problem whose objective is to meet multiple QoS metrics, such as delay, loss ratio, and jitter. We also show that the utility max-min optimization problem is a quasilinear function over delayed traffic and dropped traffic of queues, and thus the optimal solution of the problem can be obtained by employing the bisection method. Proper bandwidth management and local queue management schemes are required to achieve the optimal value. The detailed implementation of the proposed algorithm in EPON is also included in the paper.

The rest of the paper is organized as follows. Section II discusses related works on inter-ONU and intra-ONU scheduling schemes in EPON. Section III defines utilities for various applications. Section IV presents the formal mathematical formulation of the utility max-min optimization problem. Section V details the proposed bandwidth allocation and queue management scheme. The implementation of the proposed scheme is discussed in Section VI. Simulation results are shown and analyzed in Section VII. The conclusion is included in Section VIII.

II. RELATED WORKS

In EPON, the bandwidth allocation usually includes two issues: intra-ONU scheduling which arbitrates bandwidth among queues in the same ONU, and inter-ONU bandwidth arbitration which arbitrates bandwidth among different ONUs.

For the intra-ONU scheduling issue, many DBA algorithms have been proposed to ensure fairness and guarantee QoS for queues in the same ONU. For example, the DiffServ framework was proposed to be incorporated into the DBA to provision QoS guarantees [3]–[5], [10]. However, the employed strict-priority discipline when incorporating the DiffServ framework into DBA raises the so-called *light-load penalty* problem [3]. To compensate for the light-load penalty, Kramer *et al.* [3] proposed a two-stage queueing system, where a proper local queue management scheme and a priority-based scheduling algorithm are employed. Kim *et al.* [11] adopted weighted fair queuing to give queues with different weights for their priorities. For the inter-ONU scheduling issue, the proposed DBA algorithms usually regards ensuring fairness among ONUs as the scheduling objective. For example, IPACT-LS [12] prevents ONUs from monopolizing the bandwidth by setting a predetermined maximum of the granted resources. Assi *et al.* [4] proposed to satisfy requests from light-load ONUs first, while penalizing heavily-loaded ONUs.

There are two schemes to realize inter-ONU and intra-ONU scheduling in EPON. In the first scheme, OLT arbitrates the bandwidth allocation among ONUs, and informs ONUs the specific time duration it can transmit packets. After receiving the grants from OLT, ONUs decide the bandwidth allocation among its queues. With this scheme, OLT performs inter-ONU scheduling, and ONUs perform intra-ONU scheduling. An alternative scheme is to perform joint inter- and intra-ONU scheduling at OLT. OLT decides bandwidth allocation for each queue at each ONU. With this scheme, the DBA function at OLT is more complicated, and more information needs to be delivered to ONUs. One example of the joint inter- and intra-ONU scheduling scheme is that proposed by Naser *et al.* [5]. They employed a credit pooling technique as well as a weighted-share policy to enable the OLT partition the upstream bandwidth among different classes in a fair fashion. In this paper, we also focus on the joint inter- and intra-ONU scheduling scheme for the sake of fairness and low complexity of ONUs.

III. APPLICATION UTILITY

DBA in EPON is desired to facilitate any QoS profile for queues and ensure fairness among queues. To achieve a finer granularity of QoS control, we first define application utility to describe QoS requirements of applications, and then make bandwidth allocation decisions based on application utilities.

Here, we introduce the concept of *application utility* to quantify the relationship between users' degree of satisfaction and received network layer QoS performances. Formerly, Tashaka *et al.* [13] specified QoS at each level of the Internet protocol stack: physical level QoS, node level QoS, network level QoS, end-to-end level QoS, application level QoS, and user level QoS (or perceptual QoS). Throughput, delay, delay jitter, and loss ratio are typical QoS parameters considered in a network. Mean opinion score (MOS) and subjective video quality are two subjective QoS measurements for voice and video at the user level [14]. Performances in these layers are interrelated. The QoS in the upper layer depends on the QoS in the lower layer. Both MOS and subjective video quality provide numerical indications of the perceived quality of received media after compression and/or transmission, and are related to the network layer QoS performances, such as throughput and delay. In this paper, we use application utility to describe the relationship between the user-level QoS and network-level QoS.

Determining the utility of an application needs to consider the application's specific QoS requirements; this is, however, beyond the scope of this paper. In this paper, we consider application utilities as a function of packet loss ratio, packet delay, and jitter. We further unify and normalize application utilities to the range from 0 to 1. Generally, application utility possesses the property that large utility implies small packet loss ratio, small packet delay, and low jitter. Mathematically,

$$\begin{cases} 0 \le f_{i,j} \le 1, \forall i, j \\ f_{i,j}(x_1 + \varepsilon, x_2, x_3) \le f_{i,j}(x_1, x_2, x_3), \forall \varepsilon > 0 \\ f_{i,j}(x_1, x_2 + \varepsilon, x_3) \le f_{i,j}(x_1, x_2, x_3), \forall \varepsilon > 0 \\ f_{i,j}(x_1, x_2, x_3 + \varepsilon) \le f_{i,j}(x_1, x_2, x_3), \forall \varepsilon > 0 \end{cases}$$

where $f_{i,j}(x_1, x_2, x_3)$ is the application utility of queue j at ONU i, x_1 is the packet loss ratio, x_2 is the delay, and x_3 is the jitter. The application utility is a monotonic function with respect to loss, delay, and jitter. Hence, it is quasilinear over these QoS metrics. Some particular applications may be modeled by convex functions. Cao *et al.* [8] used convex bandwidth utility function to model elastic delay-tolerant traditional data applications such as email, remote terminal access, and file transfer.

By virtue of application utility, the problem of ensuring fairness among queues with diversified QoS requirements can be formulated as a utility max-min fairness optimization problem. From the optimization point of view, the utility max-min problem with the single-objective of maximizing the minimum utility is a scalarization of the multi-objective optimization problem whose objective is to meet multiple QoS metrics, such as delay, loss ratio, and jitter [15].

IV. MATHEMATICAL FORMULATION OF THE PROBLEM

In this paper, we use the scheme proposed in [4] to compensate for the gap between two dynamic bandwidth allocation (DBA) cycles. That is, OLT allocates bandwidths at the beginning of a cycle to lightly-loaded ONUs without waiting for the arrival of reports from all ONUs, whereas OLT allocates the remaining bandwidth in a cycle to heavily-loaded ONUs after receiving all reports. We adopt the adaptive cycle length, but set an upper bound to the cycle length to avoid introducing significant waiting time of backlogged traffic. If the remaining bandwidth in a cycle with the maximum length cannot satisfy all queue requests, we delay some traffic transmission and drop some traffic of some queues, where utility max-min fairness is our objective in computing the delayed traffic and dropped traffic. The problem can be described as follows:

Given network resources as well as requests from queues at ONUs in a cycle, determine the granted bandwidth and dropped traffic to meet the objective of max-min utility.

Mathematically, the time allocation problem in any cycle k can be formulated as follows.

A. Given

1) Network Resources:

1) r: The data rate of the PON.
2) $cycle$: The upper bound of the cycle length.
3) t: The beginning of the time duration to be allocated. It depends on the beginning time stamp of cycle k and the bandwidth already allocated to lightly-loaded ONUs.
4) t': The ending of the time duration to be allocated. It depends on the beginning time stamp of cycle k and the maximum cycle length.

2) Queue Requests:

1) m: The number of queues in each ONU.
2) n: The number of heavily-loaded ONUs in the PON.
3) $f_{i,j}$: The application utility of queue j at ONU i.
4) $q_{i,j}^k$: The kth reported queue size of queue j at ONU i.
5) $a_{i,j}^k$: The time that the kth report from queue j at ONU i arrived at OLT.

Besides the network resource and current queue requests, some other historical information is needed to arbitrate bandwidth among queues.

3) Historical Information:

1) $\{a_{i,j}^1, a_{i,j}^2, \ldots, a_{i,j}^{k-1}\}$: The time that the $1\text{th}, 2\text{th}, \ldots, k-1\text{th}$ report from queue j at ONU i arrives at OLT, respectively.
2) $\{\Delta_{i,j}^1, \Delta_{i,j}^2, \ldots, \Delta_{i,j}^k\}$: The interval between two consecutive report arrival time of queue j at ONU i, e.g., $\Delta_{i,j}^k$ refers to the time interval between $a_{i,j}^{k-1}$ and $a_{i,j}^k$.
3) $\{q_{i,j}^1, q_{i,j}^2, \ldots, q_{i,j}^{k-1}\}$: The $1\text{th}, 2\text{th}, \ldots, k-1\text{th}$ reported traffic of queue j at ONU i, respectively.
4) $\{tr_{i,j}^1, tr_{i,j}^2, \ldots, tr_{i,j}^{k-1}\}$: The transmitted traffic during intervals $\Delta_{i,j}^1, \Delta_{i,j}^2, \ldots, \Delta_{i,j}^{k-1}$, respectively.
5) $\{dr_{i,j}^1, dr_{i,j}^2, \ldots, dr_{i,j}^{k-1}\}$: The dropped traffic during intervals $\Delta_{i,j}^1, \Delta_{i,j}^2, \ldots, \Delta_{i,j}^{k-1}$, respectively.
6) $\{ar_{i,j}^1, ar_{i,j}^2, \ldots, ar_{i,j}^{k-1}\}$: The arrival traffic during intervals $\Delta_{i,j}^1, \Delta_{i,j}^2, \ldots, \Delta_{i,j}^{k-1}$, respectively.

B. Determine

1) $tr_{i,j}^k$: The transmitted traffic during interval $\Delta_{i,j}^k$.
2) $dr_{i,j}^k$: The dropped traffic during interval $\Delta_{i,j}^k$.

C. Define

In EPON, OLT does not contain information with granularity as fine as the packet level. It does not know the arrival time and the departure time of every packet. So, we estimate the average loss, delay, and jitter of packets in a queue.

1) $l_{i,j}$: the average packet loss ratio of queue j at ONU i
2) $d_{i,j}$: the average packet delay of queue j at ONU i.
3) $v_{i,j}$: jitter is defined as the maximum difference among the delays of packets [16]. We assume the minimum delay is as low as zero. Then, jitter equals to the maximum delay of packets in queue j at ONU i.

D. Constraints

The time resource between t and t' is not oversubscribed.

$$\sum_{i,j} tr_{i,j}^k \leq (t'-t)\cdot r. \tag{1}$$

E. Objective

$$\text{maximize} \quad \min(f_{i,j}).$$

As shown before, $f_{i,j}$ depends on loss $l_{i,j}$, delay $d_{i,j}$, and jitter $v_{i,j}$. We next estimate $l_{i,j}$, $d_{i,j}$, and $v_{i,j}$ from the dropped traffic and transmitted traffic. Then, we continue to discuss the scheme of obtaining an optimal solution to the resource allocation problem.

V. UTILITY MAX-MIN FAIR BANDWIDTH ALLOCATION AND QUEUE MANAGEMENT

In EPON, after collecting reports from ONUs, OLT estimates the real-time QoS performances of queues at ONUs, and then tries to maximize the minimum utility received by queues. In this section, we estimate QoS performances of ONUs and present the scheme to address the utility max-min fair resource allocation problem. Before the QoS estimation, we first discuss the queue management scheme which is employed to maximize application utilities to the best.

A. Drop Head Queue Management

After a queue obtains the information of the amount of traffic of its queues to be dropped, it selects packets to be dropped if necessarily. Drop Tail is a typical queue management algorithm used by Internet routers. It drops the newly arrived packets when the buffer is filled to its maximum capacity.

Instead of dropping packets from the tail of the queue, we drop packets from the head of the queue in this paper. For packets at the head of the queue, they experience a longer waiting time in the queue as compared to those at the tail of the queue. Rather than allocating the channel resource to those packets with larger delay, we drop packets from the head to allocate the precious channel resources to packets which have smaller delay, thus achieving high utility of the queue. So, in this paper, the backlogged traffic is dropped with higher priority over the newly arrived traffic for higher utility.

B. Estimating QoS Metric of Queues

For each particular application, the application utility depends on the end-to-end delay, jitter, and packet loss ratio, which are affected by not only the access network, but also all the other network elements. As packets travel all the way to the access network, they have already experienced certain delay and loss ratio. The resource allocation scheme in access networks is desired to take these already experienced delay and packet loss ratio into account. However, the estimation of already experienced delay and packet loss ratio requires a proper real-time network measurement scheme, which is beyond the scope of this article. In this article, we only consider the delay, jitter, and packet loss ratio caused by the access network with the assumption that delay, jitter, and packet loss ratio affected by other networks are negligible.

For the packet loss ratio, it can be estimated as the ratio between the dropped traffic and the requested traffic, i.e.,

$$\frac{dr_{i,j}^{k}}{q_{i,j}^{k}}.$$

However, for delay and jitter, the estimation is more complicated. In order to estimate the delay and jitter performance of queues, we estimate packet arrival time and departure time first.

For the downstream transmission, it is possible for OLT to track the arrival time of each downstream packet. However, for the upstream transmission, OLT does not possess the information with granularity as fine as the packet level, but can only estimate the arrival time based on the sizes of queue requests in each resource allocation cycle. In addition, it is hard to predict the future network traffic, and estimate the time that the delayed traffic will be transmitted. In this paper, we make optimistic assumption that the delayed packets in the current cycle can be successfully transmitted in the next cycle. This assumption is made only to facilitate the estimation of the traffic delay and jitter, though it may not be held in reality. The following first estimate the packet arrival time, and then address the issue of estimating packet loss ratio, delay, and jitter.

The total arrival traffic between time 0 and $a_{i,j}^{k}$ equals to $\sum_{l=1}^{k} ar_{i,j}^{l}$. Among the total arrival traffic $\sum_{l=1}^{k} ar_{i,j}^{l}$, $\sum_{l=1}^{k} dr_{i,j}^{l}$ has been dropped, and $\sum_{l=1}^{k} tr_{i,j}^{l}$ has been successfully transmitted. Therefore, the request $q_{i,j}^{k}$ of queue j of user i at time $a_{i,j}^{k}$, equals to $\sum_{l=1}^{k} (ar_{i,j}^{l} - tr_{i,j}^{l} - dr_{i,j}^{l})$. We can further obtain that

$$ar_{i,j}^{k} = q_{i,j}^{k} + \sum_{l=1}^{k} (tr_{i,j}^{l} + dr_{i,j}^{l}) - \sum_{l=1}^{k-1} ar_{i,j}^{l}. \tag{2}$$

At the right side of (2), $q_{i,j}^{k}$ is the request reported to OLT. Both $tr_{i,j}^{l}$ and $dr_{i,j}^{l}$ are decided by OLT. Hence, OLT can infer the arrival traffic $ar_{i,j}^{k}$ during time interval $\Delta_{i,j}^{k}$ by recursion.

Besides the arrival traffic $ar_{i,j}^{k}$ during interval $\Delta_{i,j}^{k}$, OLT can estimate the arrival time of all traffic contained in $q_{i,j}^{k}$. As explained earlier, both the dropping and transmitting are from the head of the queue with the earliest arrival so as to let precious resources be used for transmitting traffic with smaller delay, and hence larger utility. Then, among the total $\sum_{l=1}^{k} ar_{i,j}^{l}$ arrival traffic before time $a_{i,j}^{k}$, the first $\sum_{l=1}^{k} (tr_{i,j}^{l} + dr_{i,j}^{l})$ arrival traffic is either transmitted or dropped, and the latest arrival $\sum_{l=1}^{k} (ar_{i,j}^{l} - tr_{i,j}^{l} - dr_{i,j}^{l})$ traffic remains in the queue and is reported to OLT. Among the $q_{i,j}^{k}$ request traffic, assume $x_{i,j}^{k,l}$ traffic arrives during time interval $\Delta_{i,j}^{l}$. Then, the following can be derived.

$$x_{i,j}^{k,l} = \begin{cases} \min\{ar_{i,j}^{k}, q_{i,j}^{k}\}, & \text{if } l = k \\ \min\left\{ ar_{i,j}^{l}, \left(q_{i,j}^{k} - \sum\limits_{m=l+1}^{k} x_{i,j}^{k,m} \right)^{+} \right\}, & \text{otherwise} \end{cases}. \tag{3}$$

For traffic $x_{i,j}^{k,l}$ which arrived during time interval $\Delta_{i,j}^{l}$, the earliest arrival time and the average arrival time equal to $a_{i,j}^{l-1}$ and $(a_{i,j}^{l} + a_{i,j}^{l-1})/2$, respectively. For any given τ, assume $x_{i,j}^{l,\tau} > 0$ and $x_{i,j}^{l,\tau-1} = 0$. Then, among the traffic contained in request $q_{i,j}^{k}$:

- the earliest arrival time is $a_{i,j}^{\tau-1}$;
- the average arrival time is $\sum_{l=\tau}^{k} (a_{i,j}^{l} + a_{i,j}^{l-1})/2 \cdot x_{i,j}^{k,l}/q_{i,j}^{k}$.

Regarding the departure time estimation, $tr_{i,j}^{k}$ among $q_{i,j}^{k}$ traffic is transmitted between time t and t'. We optimistically assumed that the delayed traffic $q_{i,j}^{k} - tr_{i,j}^{k} - dr_{i,j}^{k}$ in the current cycle can be successfully transmitted in the next cycle. To facilitate estimation, we further assume that the length of the next cycle equals to the cycle duration upper bound $cycle$. Then, among the traffic contained in request $q_{i,j}^{k}$,

- the average departure time of $tr_{i,j}^{k}$ traffic is $(t + t')/2$;
- the average departure time of $q_{i,j}^{k} - tr_{i,j}^{k} - dr_{i,j}^{k}$ traffic is $t' + cycle/2$;
- the largest departure time is $t' + cycle$.

Based on the estimation of average arrival and departure time, the average delay can be estimated as follows.

$$d_{i,j} = \frac{\frac{(t+t')}{2} \cdot tr_{i,j}^{k} + \left(t' + \frac{cycle}{2}\right) \cdot (q_{i,j}^{k} - tr_{i,j}^{k} - dr_{i,j}^{k})}{(q_{i,j}^{k} - dr_{i,j}^{k})} - \sum_{l=\tau}^{k} (a_{i,j}^{l} + a_{i,j}^{l-1}) \cdot \frac{x_{i,j}^{k,l}}{2q_{i,j}^{k}}.$$

We assume the minimum delay is as low as zero. The jitter then equals to the maximum delay of packets.

$$v_{i,j} = t' + cycle - a_{i,j}^{\tau-1} \ \text{ if } \ q_{i,j}^{k} - dr_{i,j}^{k} - tr_{i,j}^{k} > 0.$$

As we can see, the average loss is a linear function over the dropping traffic $dr_{i,j}^{k}$; the average delay is a linear fractional function over the dropping traffic $dr_{i,j}^{k}$ and transmitting traffic $tr_{i,j}^{k}$; the jitter is a step function. Then, the sublevel sets of all the three functions are convex. We also know that $-f_{i,j}$ is a

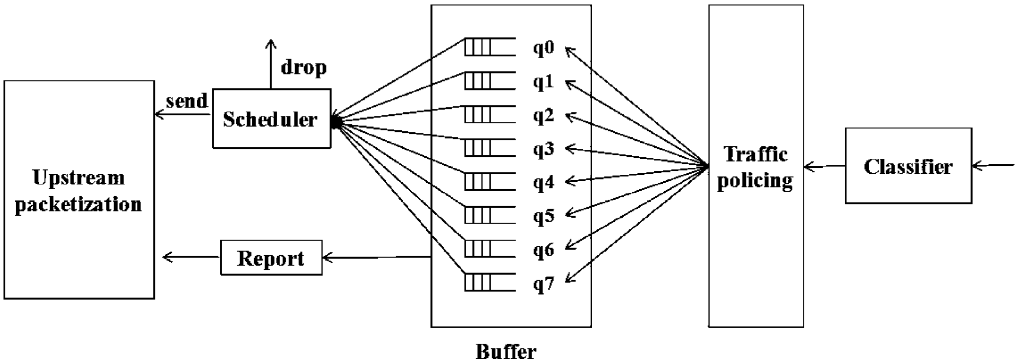


Fig. 1. Implementation at ONUs.

non-decreasing function with respect to loss $l_{i,j}$, delay $d_{i,j}$, and jitter $v_{i,j}$. Accordingly, it can be derived that the sublevel set of $-f_{i,j}$ as a function of $dr^k_{i,j}$ and $tr^k_{i,j}$ is convex. Therefore, $-f_{i,j}$ is a quasiconvex function over $dr^k_{i,j}$ and $tr^k_{i,j}$, and $f_{i,j}$ is a quasiconcave function.

C. Utility Max-Min Fair Bandwidth Allocation

With the estimation of QoS performances, OLT can perform bandwidth allocation for utility max-min fairness. Owing to the quasiconcave property of the utility function $f_{i,j}$, we herein employ the bisection method to obtain the optimal solution to the utility max-min optimization problem [17]. The main idea is as follows: Let a be the lower bound of the utility, b be the upper bound of the utility, and x be the utility to be achieved. Since we assume the application utility is normalized between 0 and 1, initially, a is set as 0, b is set as 1, and x is set as 1. We calculate the maximum dropped traffic $dr^k_{i,j}$ and delayed traffic $tr^k_{i,j} - dr^k_{i,j}$ which can guarantee x. If the sum of the minimum required bandwidth $tr^k_{i,j}$ is less than the available bandwidth $cycle$, the upper bound b is updated to be x, and x is decreased to the midpoint between a and b; otherwise the lower bound a is increased to x, and x is increased to the midpoint between a and b. The above process is performed recursively until a and b are close enough to each other. The pseudocode of the algorithm is presented as follows.

Algorithm 1 Determine $tr^k_{i,j}$ and $dr^k_{i,j}$

```
1: Let $a = 0, b = 1, x = 1$
2: while $b - a < \varepsilon$ do
3:     calculate the maximum allowed loss ratio of each queue to ensure its corresponding utility to be above $x$
4:     calculate the maximum $dr^k_{i,j}$ for each queue
5:     calculate the maximum delay and jitter of each queue to ensure its corresponding utility to be above $x$
6:     calculate the maximum $tr^k_{i,j} - dr^k_{i,j}$ for each queue
7:     calculate the minimum required $dr^k_{i,j}$ for each queue
8:     if $\sum_{i,j} tr^k_{i,j} < cycle$ bf then
9:         $b = x, x = (a + b)/2$
10:    else
11:        $a = x, x = (a + b)/2$
12:    end if
13: end while
```

In Algorithm 1, lines 4 and line 6 are calculated based on the estimation discussed in Section V-B. Lines 3 and 5 are calculated based on the specific application utility function. Let function $f^1(x_1)$ describe the application utility function with respect to loss ratio, function $f^2(x_2)$ describe the application utility function with respect to packet delay, and function $f^3(x_3)$ describe the application utility function with respect to jitter. $f^1_{i,j}(x_1) = f_{i,j}(x_1, 0, 0)$, $f^2_{i,j}(x_2) = f_{i,j}(0, x_2, 0)$, and $f^3_{i,j}(x_3) = f_{i,j}(0, 0, x_3)$, where $f_{i,j}(x_1, x_2, x_3)$ is the application utility function as defined in Section III. The maximum allowed loss ratio, the maximum delay, and the maximum jitter are obtained from the inverse function of $f^1_{i,j}(x_1)$, $f^2_{i,j}(x_2)$, and $f^3_{i,j}(x_3)$, respectively.

ALGORITHM IMPLEMENTATION

This section discusses the implementation of our proposed algorithm at ONUs and OLT.

Fig. 1 shows the block diagram of ONUs. The upcoming packets are first classified into different classes according to their corresponding applications. Then, a traffic policing process is performed to make sure that the traffic shape can be accommodated by the queues. Afterwards, packets are queued in buffers, and wait to be scheduled by the scheduler. After ONUs receive GRANT messages from OLT, they determine the amount of traffic to be transmitted and dropped, respectively. Then, ONUs drop traffic from the head of their queues, and send the traffic to be transmitted to their respective packetization blocks. After this, ONUs take the snapshot of their queues' lengths, prepare REPORT messages, and piggyback reports onto their data packets. As specified in IEEE 802.3ah, each REPORT can carry up to eight queue requests. Therefore, at most eight kinds of applications are supported. If there are more than eight kinds of applications, these applications need be classified into no more than eight classes.

At the OLT side, OLT first obtains REPORT messages from ONUs. With these collected reports and the application utility functions, OLT performs utility max-min fair DBA. Then, OLT notifies its DBA decision to ONUs via GATE messages. As

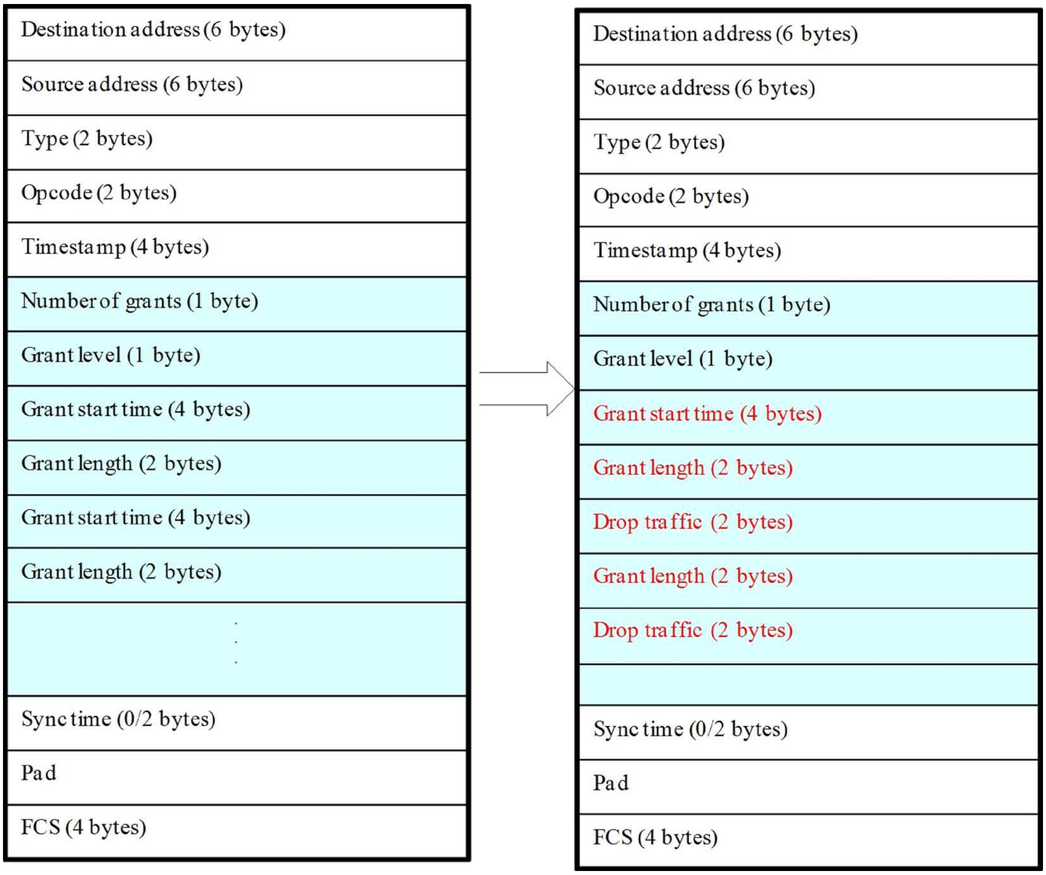


Fig. 2. Modification of the GATE message.

specified in IEEE 802.3ah, each GATE message can carry up to four grants. Each grant contains one “start time” field of four bytes and one “grant length” of two bytes. However, with this format, OLT cannot convey both the delayed traffic and dropped traffic information to ONUs.

In order to implement our proposed algorithm, we have made some changes on the format of the GATE message. In the time allocation process, we will always assign queues in one ONU with continuous time durations in a cycle. In this case, for a particular ONU, the time duration allocated to one queue follows that of another queue. Consequently, with the “start time” of the first queue and the time duration allocated to all queues, the “start time” of all the other queues can be deduced. In another words, only the first queue needs the “start time” field, and the information at the “start time” fields of all other queues is redundant and can be voided. Utilizing the voided “start time” fields, we can convey both the delayed traffic and dropped traffic information to ONUs.

Fig. 2 shows the modified GATE message. Instead of using six bytes to describe one grant, the modified GATE message only uses four bytes, among which two bytes for the “grant length” field, and another two bytes for the “dropped traffic” field. The original GATE message uses 24 bytes to describe four grants, each grant is described by six bytes. The modified GATE message maintains the same length of 24 bytes with the original GATE message. Among these 24 bytes, the first four bytes are used as the “start time” field of the first queue. The remaining 20 bytes can carry up to five grants since each grant is described by four bytes.

VI. SIMULATION RESULTS AND ANALYSIS

In this section, we investigate the performance of our proposed utility max-min fair algorithm presented above. The simulation model is developed on the OPNET platform. The number of ONUs is set as 16. The round trip time between ONUs and OLT is set as 125 μs. The channel data rate is set as 1.25 Gb/s. The maximum cycle length is set as 2 ms. Since self-similarity is exhibited by many applications, we input the queues with self-similar traffic. The pareto parameter is set as 0.8. The packet length is uniformly distributed between 64 bytes to 1500 bytes. An ONU in a cycle is labeled as light-load when the total request of its queues is less than 1 K bytes.

In the simulation, we want to show that our scheme can guarantee fairness among queues, each of which may exhibit any application utility. We assume each ONU has five queues corresponding to five kinds of applications. Our objective is to show that QoS profiles received by the five queues conform to the corresponding profiles derived from their application utilities. We claim that fairness is achieved if application utilities obtained by queues are similar with each other.

First, we consider the application utility as a function of packet loss ratio, i.e., $f_{i,j}(x_1, x_2, x_3) = f_{i,j}^1(x_1)$. For five queues in each ONU, $f_{i,j}^1(x_1)$ is defined as follows:

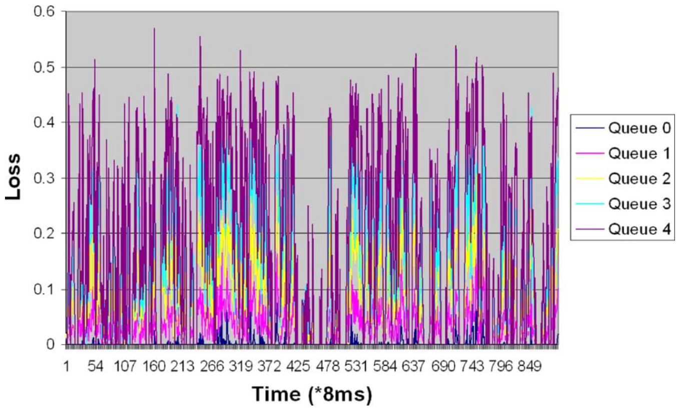


Fig. 3. Packet loss ratio versus application utilities.

$$f^1_{i,0}(x_1)=\begin{cases}1 & x_1\le 0.01\\ \frac{(1-x_1)}{0.99} & x_1\in[0.01,1]\end{cases},\ \forall i$$

$$f^1_{i,1}(x_1)=\begin{cases}1 & x_1\le 0.1\\ \frac{(1-x_1)}{0.9} & x_1\in[0.1,1]\end{cases},\ \forall i$$

$$f^1_{i,2}(x_1)=\begin{cases}1 & x_1\le 0.2\\ \frac{(1-x_1)}{0.8} & x_1\in[0.2,1]\end{cases},\ \forall i$$

$$f^1_{i,3}(x_1)=\begin{cases}1 & x_1\le 0.3\\ \frac{(1-x_1)}{0.7} & x_1\in[0.3,1]\end{cases},\ \forall i$$

$$f^1_{i,4}(x_1)=\begin{cases}1 & x_1\le 0.4\\ \frac{(1-x_1)}{0.6} & x_1\in[0.4,1]\end{cases},\ \forall i.$$

Fig. 3 shows the sampled packet loss ratio of queues with the above five different application utilities. The sampling is taken every 8 ms. From the application function $f^1_{i,0}(x_1)$, $f^1_{i,1}(x_1)$, $f^1_{i,2}(x_1)$, $f^1_{i,3}(x_1)$, and $f^1_{i,4}(x_1)$, we know that utilities of the five queues equal to the highest value of 1 when the packet loss ratios of queue 0, 1, 2, 3 and 4 are below 0.01, 0.1, 0.2, 0.3, and 0.4, respectively. Therefore, for fairness, if the packet loss ratio of queue 4 is lower than 0.4, packet loss ratio of queue 0, 1, 2, and 3 should not exceed 0.01, 0.1, 0.2, and 0.3, respectively. From Fig. 3, we can see that almost all points comply with this rule. On the other hand, when the network is heavily loaded and the maximum utility cannot be guaranteed for queues, the packet loss ratio of queue 0, 1, 2, 3, and 4 will be increased to be higher than 0.01, 0.1, 0.2, 0.3, and 0.4, respectively. For fairness, this increase should enable the five queues achieve the same utility. For example, based on the application utilities, when the packet loss ratio of queue 2 equals to 0.24, queue 0, queue 1, queue 3, and queue 4 should experience packet loss ratio of 0.065, 0.15, 0.34 and 0.43, respectively, for the same utility. Simulation results show that when the packet loss ratio of queue 2 is increased to around 0.24, packet loss ratio of queue 0, queue 1, queue 2, and queue 3 are around 0.078, 0.166, 0.36, and 0.45, respectively. The minor discrepancy between the theoretical values and the simulation values is probably attributed to the disagreement between the number of dropped bits and the size of the packet to be dropped. Therefore, in terms of the packet loss ratio, our algorithm can guarantee fairness among the five queues.

Here, we consider application utility as a function of packet delay, i.e., $f_{i,j}(x_1,x_2,x_3)=f^2_{i,j}(x_2)$. $f^2_{i,j}(x_2)$ for the five queues are defined as follows:

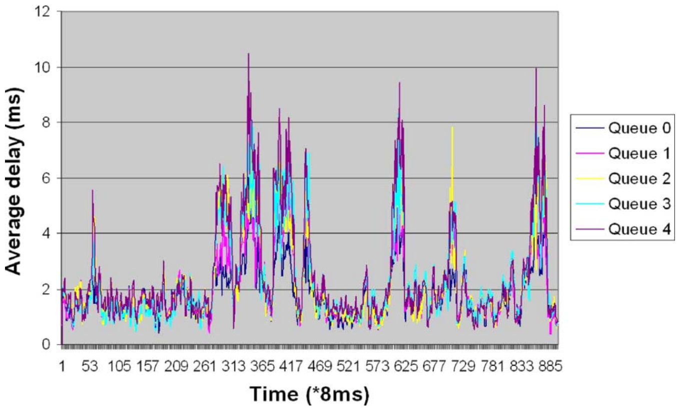


Fig. 4. Packet delay versus application utilities.

$$f^2_{i,0}(x_2)=\begin{cases}1 & x_2\le 3\text{ ms}\\ e^{(x_2-3)/3} & x_2>3\text{ ms}\end{cases},\ \forall i$$

$$f^2_{i,1}(x_2)=\begin{cases}1 & x_2\le 4\text{ ms}\\ e^{(x_2-4)/4} & x_2>4\text{ ms}\end{cases},\ \forall i$$

$$f^2_{i,2}(x_2)=\begin{cases}1 & x_2\le 5\text{ ms}\\ e^{(x_2-5)/5} & x_2>5\text{ ms}\end{cases},\ \forall i$$

$$f^2_{i,3}(x_2)=\begin{cases}1 & x_2\le 6\text{ ms}\\ e^{(x_2-6)/6} & x_2>6\text{ ms}\end{cases},\ \forall i$$

$$f^2_{i,4}(x_2)=\begin{cases}1 & x_2\le 7\text{ ms}\\ e^{(x_2-7)/7} & x_2>7\text{ ms}\end{cases},\ \forall i.$$

Fig. 4 shows the sampled average delay of packets arrived during each sampling period. Due to the bursty characteristic of the arriving traffic, the delay of traffic for all the five kinds of queues fluctuates. Under the light load scenario, requests from all queues can be satisfied, and delay of all queues are about 3/2 times of the DBA cycle. Under the heavy load scenario, delay of all queues increases but with different degrees, as determined by their own application utilities. Let u be the converged utility in Algorithm 1 under heavy load scenario, i.e., $u=a$or b with $a\approx b$. Then, delays of queue 0, queue 1, queue 2, queue 3, and queue 4 are $3(1-\ln u)$, $4(1-\ln u)$, $5(1-\ln u)$, $6(1-\ln u)$, and $7(1-\ln u)$, respectively. Simulation results show that the delay of queue 0 is the lowest, whereas the delay of queue 4 is the highest. The proportions between the delays of any two queues conform roughly to the theoretical values. So, the simulated delay performances of the five queues generally agree with the delay profiles derived from their respective application utilities, but with some slight discrepancy. The main reason of the discrepancy lies in the inaccurate estimation of the delay. We make optimistic assumption that delayed traffic can be successfully transmitted in the next cycle. However, the delayed traffic may not get a chance to be transmitted in the next cycle, but may be further delayed. In this case, the queue with delayed traffic has smaller utility over others though Algorithm 1 guarantees the same utility for queues.

From the above, we can see that the QoS profiles obtained from the simulations conform to those derived from application utilities. When the network is heavily loaded, the queues can achieve nearly equal utilities. Hence, fairness is guaranteed for the queues. Our scheme is potentially able to accommodate any number of queue classes by properly designing their respective application utilities.

VII. CONCLUSION

This paper has tackled the issue of ensuring fairness among applications with diversified QoS requirements in EPONs. We first employ application utility to describe the relationship between users' QoE and network-level QoS of each application. Application utility is a quasilinear function over packet loss ratio, delay, and jitter. By virtue of application utility, we formulate the problem of ensuring fairness among applications with diversified QoS requirements into a utility max-min fairness problem. The maximization problem possesses quasiconcave property with respect to the delayed traffic and dropped traffic. We hence adopt the bisection method to obtain the optimal solution of the maximized minimum utility. The optimal value can be achieved via proper bandwidth management and queue management. As compared to schemes using DiffServ, our proposed scheme possesses finer granularity and is able to ensure fairness among diversified applications with proper design of application utilities and estimation of QoS metrics.

REFERENCES

[1] J. Zhang and N. Ansari, "Utility max-min fair resource allocation for diversified applications in EPON," in *AccessNets*, Hong Kong, China, 2009.

[2] G. Kramer and G. Pesavento, "Ethernet passive optical network (EPON): Building a next-generation optical access network," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 66–73, Feb. 2002.

[3] G. Kramer, B. Mukherjee, S. Dixit, Y. Ye, and R. Hirth, "Supporting differentiated classes of service in Ethernet passive optical networks," *J. Opt. Netw.*, vol. 1, no. 8, pp. 280–298, 2002.

[4] C. Assi, Y. Ye, S. Dixit, and M. Ali, "Dynamic bandwidth allocation for quality-of-service over Ethernet PONs," *IEEE J. Sel. Areas Commun.*, vol. 21, pp. 1467–1477, Nov. 2003.

[5] H. Naser and H. Mouftah, "A joint-ONU interval-based dynamic scheduling algorithm for Ethernet passive optical networks," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 889–899, Aug. 2006.

[6] Y. Luo and N. Ansari, "Bandwidth allocation for multiservice access on EPONs," *IEEE Commun. Mag.*, vol. 43, no. 2, pp. S16–S21, Feb. 2005.

[7] S. Jiang and J. Xie, "A frame division method for prioritized DBA in EPON," *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 83–94, 2006.

[8] Z. Cao and E. Zegura, "Utility max-min: An application-oriented bandwidth allocation scheme," in *IEEE INFOCOM*, Mar. 1999, vol. 2, pp. 793–801.

[9] S. Thakolsri, S. Khan, E. Steinbach, and W. Kellerer, "QoE-driven cross-layer optimization for high speed downlink packet access," *J. Commun.*, vol. 4, no. 9, pp. 669–680, 2009.

[10] Y. Luo and N. Ansari, "Limited sharing with traffic prediction for dynamic bandwidth allocation and QoS provisioning over Ethernet passive optical networks," *J. Opt. Netw.*, vol. 4, no. 9, pp. 561–572, 2005.

[11] C. Kim, T.-W. Yoo, and B.-T. Kim, "A hierarchical weighted round robin EPON DBA scheme and its comparison with cyclic water-filling algorithm," in *IEEE Int. Conf. Communications*, Jun. 2007, pp. 2156–2161.

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

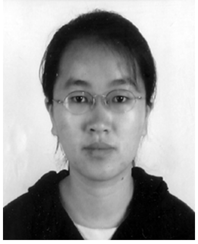
[13] S. Tasaka and Y. Ishibashi, "Mutually compensatory property of multimedia QoS," in *IEEE Int. Conf. Communications*, 2002, vol. 2, pp. 1105–1111.

[14] A. Takahashi, H. Yoshino, and N. Kitawaki, "Perceptual QoS assessment technologies for VoIP," *IEEE Commun. Mag.*, vol. 42, no. 7, pp. 28–34, Jul. 2004.

[15] D. Nace and M. Pioro, "Max-min fairness and its applications to routing and load-balancing in communication networks: A tutorial," *IEEE Commun. Surv. Tutorials*, vol. 10, no. 4, pp. 5–17, 2008.

[16] D. Verma, H. Zhang, and D. Ferrari, "Delay jitter control for real-time communication in a packet switching network," in *IEEE TRICOMM*, Apr. 1991, pp. 35–43.

[17] S. Boyd and L. Vandenberghe*, Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.



Jingjing Zhang (S'09) received the B.E. degree from Xi'an Institute of Posts and Telecommunications, China, in 2003, and the M.E. degree from Shanghai Jiao Tong University, China, in 2006, both in electrical engineering. She is pursuing the Ph.D. degree in electrical engineering at the New Jersey Institute of Technology (NJIT), Newark.

Her research interests are broadly in optimization and broadband computer networking, with current focuses mainly on capacity analysis, planning, design, and resource allocation in next generation passive optical networks, handover and resource allocation in integrated wireless and optical access networks, and QoE provisioning in next generation networks.

Nirwan Ansari (S'78–M'83–SM'94–F'09) is Professor of Electrical and Computer Engineering at New Jersey Institute of Technology (NJIT), Newark. His current research focuses on various aspects of broadband networks and multimedia communications.

Dr. Ansari was/is serving on the editorial/advisory boards of eight journals, including as Senior Technical Editor of *IEEE Communications Magazine* (2006–2009). Some of his recent recognitions include IEEE Leadership Award (2007, from Central Jersey/Princeton Section), the NJIT Excellence in Teaching in Outstanding Professional Development Award (2008), IEEE MGA Leadership Award (2008), the NCE Excellence in Teaching Award (2009), a couple of best paper awards. the Thomas Alva Edison Patent award (2010) and designation as an IEEE Communications Society Distinguished Lecturer (two terms, 2006–2009).