Utility Max-Min Fair Resource Allocation for Diversified Applications in EPON

Jingjing Zhang and Nirwan Ansari

Advanced Networking laboratory
New Jersey Institute of Technology, Newark NJ 07102, USA
{jz58,nirwan.ansari}@njit.edu

Abstract. In EPONs, differentiated services enable higher quality of service (QoS) for some queues over others. However, owing to the coarse granularity of DiffServ, DiffServ in EPONs can hardly facilitate any particular QoS profile. This paper investigates an application-oriented bandwidth allocation scheme to ensure fairness among queues with diversified QoS requirements. We first define application utilities to quantify users' quality of experience (QoE) as a function of network layer QoS metrics. We then formulate the fair resource allocation issue into a max-min utility problem, which is quasi-concave over queues' delayed traffic and dropped traffic. We further employ the bisection method to obtain the optimal solution of the quasi-concave maximization problem. The optimal value can be achieved by proper bandwidth allocation and queue management schemes in EPONs.

Keywords: QoE, EPON, utility, fairness, optimization.

1 Introduction

Differentiated services (DiffServ) is widely employed in access networks for quality of service (QoS) provisioning. Specifically, it 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. However, the coarse granularity of DiffServ can hardly meet any particular QoS requirement imposed by various applications. This is a critical issue for future access networks with 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 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 classes in DiffServ. Inappropriate QoS mapping leads to either QoS

X. Jun Hei and L. Cheung (Eds.): AccessNets 2009, LNICST 37, pp. 14–24, 2010.

over-provisioning or QoS under-provisioning. The diversified QoS requirements of applications pose great challenges on resource allocation in access networks.

This paper focuses on ensuring fairness for queues with diversified QoS requirements in Ethernet Passive Optical Networks (EPONs), which have gained popularity among the access network technologies for their low cost, 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 its 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. However, IEEE802.3ah does not specify any DBA algorithms for EPON. Fairness and QoS guarantee are usually regarded as objectives 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?***

To describe the diversified QoS requirements of applications, we adopt the concept of *application utility* to quantify users' quality of experience (QoE) as a function of received QoS of the specific application [1]. 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 degree at the user-level and high QoS performances at the network-level.

By virtue of application utility, we define fairness in terms of application utilities, and formulate the problem of ensuring fairness for requests as a utility max-min fairness optimization problem. From the optimization point of view, the single-objective utility max-min problem is a scalarization of the multi-objective max-min fairness optimization with respect to a set of QoS metrics, such as delay, loss ratio, and jitter. We also show that the utility max-min fairness optimization problem is quasi-linear over delayed traffic and dropped traffic of queues, in which the optimal solution can be obtained by employing the bisection method. To achieve the optimal value in the EPON system, proper bandwidth management and local queue management are required.

2 Related Works

DBA with fairness and QoS guarantee has received broad research attention during the past several years. As a seminal work in EPON DBA, IPACT interleaves polling messages with Ethernet frame transmission to maximize link utilization [2]. To provision QoS guarantees, the DiffServ framework was proposed to be incorporated into the DBA to address the intra-ONU scheduling issue [3, 4, 5, 6, 7]. Regarding fairness, 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] further proposed a two-stage queueing system, where

a proper local queue management scheme and a priority-based scheduling algorithm are employed. Kim *et al.* [8] adopted weighted fair queuing to give queues with different weights for their priorities. Besides intra-ONU scheduling, inter-ONU scheduling is needed to arbitrate bandwidth among ONUs for fairness. IPACT-LS prevents ONUs from monopolizing the bandwidth by setting a predetermined maximum of the granted resources [2]. Assi *et al.* [4] proposed to satisfy requests from light-load ONUs first, while penalizing heavily-loaded ONUs. Naser *et al.* [5] combined inter-ONU scheduling and intra-ONU scheduling together. Specifically, 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.

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

3 Application Utility

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.* [9] 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). Typically, 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 [10]. 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 quasi-linear over these QoS

metric. Some particular applications may be modeled by convex functions. Cao *et al.* [1] 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 single-objective max-min optimization with respect to application utility can be considered as a scalarization of the multi-objective max-min fairness optimization with respect to a set of criteria of delay, loss ratio, and jitter [11].

4 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 first discuss the queue management scheme, and then estimate QoS performances of ONUs and present the scheme to address the utility max-min fair resource allocation problem.

4.1 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.

4.2 Estimating QoS Metric of Queues

OLT needs some information of queues at ONUs in order to estimate the QoS metric and calculate their utilities. Such information includes the amount of successfully transmitted traffic, the time stamp when the traffic is arrived, and the time stamp when the traffic is transmitted. However, OLT does not contain information with granularity as fine as the packet level. So, we estimate the average loss, delay, and jitter of packets in a queue. 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. The following address the issue of estimating packet loss ratio, delay, and jitter.

Table 1 list the notations used in this section.

Table 1. Notation

Symbol	Definition
$cycle$	The upper bound of the cycle duration
$q_{i,j}$	The reported traffic of queue j at ONU i
$q^b_{i,j}$	The backlogged traffic of queue j at ONU i in the last cycle
$\Delta_{i,j}$	The time duration allocated to queue j at ONU i
$\delta_{i,j}$	The dropped traffic of queue j at ONU i in the current DBA cycle
$t^1_{i,j}$	The last time stamp that the status of queue j at ONU i is reported
$t^2_{i,j}$	The time before the last time stamp that the status of queue j at ONU i is reported
$l_{i,j}$	The data loss ratio of queue j at ONU i
$d_{i,j}$	The average delay of successfully transmitted packets at queue j at ONU i
$v_{i,j}$	The jitter of successfully transmitted packets at queue j at ONU i
$d^b_{i,j}$	The average time that the backlogged traffic $q^b_{i,j}$ of queue j at ONU i spent in the buffer before time $t^2_{i,j}$
$d^{mb}_{i,j}$	The longest time that the backlogged traffic $q^b_{i,j}$ of queue j at ONU i spent in the buffer before time $t^2_{i,j}$
$\alpha_{i,j}$	The beginning time assigned to queue j at ONU i

Average Loss Ratio of the $q_{i,j}$ Reported Traffic. For queue j at ONU i, $\delta_{i,j}$ traffic among the total $q_{i,j}$ traffic is dropped. With the assumption that the delayed traffic can be transmitted finally, $q_{i,j} - \delta_{i,j}$ among $q_{i,j}$ is successfully transmitted. The average loss ratio $l_{i,j}$ is $(q_{i,j} - \delta_{i,j})/q_{i,j}$.

Average Delay of the $q_{i,j}$ Reported Traffic. We analyze four scenarios as follows.

- For the newly arrived $q_{i,j} - q^b_{i,j}$ traffic of queue j at ONU i, the average arrival time is $(t^1_{i,j} + t^2_{i,j})/2$. If they are successfully transmitted in the current cycle, the average departure time is $\alpha_{i,j} + \Delta_{i,j}/2$. The average delay of the newly arrived traffic is $\alpha_{i,j} + \Delta_{i,j}/2 - (t^1_{i,j} + t^2_{i,j})/2$.
- For the newly arrived traffic in the current cycle, if they are further delayed to the next cycle, the average delay $d_{i,j}$ is $d_{i,j} = \alpha_{i,j} + \Delta_{i,j}/2 - (t^1_{i,j} + t^2_{i,j})/2 + cycle$.
- For the backlogged traffic $q^b_{i,j}$ who already spent on average $d^b_{i,j}$ in the buffer before time $t^2_{i,j}$, the average delay is $d^b_{i,j} + \alpha_{i,j} + \Delta_{i,j}/2 - t^2_{i,j}$ under the condition that they are successfully transmitted in the current cycle.
- For the backlogged traffic $q^b_{i,j}$, if they are further delayed to the next cycle, the average delay will be $d^b_{i,j} + \alpha_{i,j} + \Delta_{i,j}/2 - t^2_{i,j} + cycle$.

Jitter of the $q_{i,j}$ Reported Traffic. We analyze four scenarios as follows.

- For the newly arrived traffic, if they are successfully transmitted in the current cycle, the maximum delay is $\alpha_{i,j} + \Delta_{i,j} - t^2_{i,j}$.

- For the newly arrival traffic, if some packets are delayed to the next cycle, the maximum delay of the $q_{i,j}$ reported traffic is $\alpha_{i,j} + \Delta_{i,j} - t^2_{i,j} + cycle$.
- For the backlogged traffic, if they are successfully transmitted in the current cycle, the maximum delay is $\alpha_{i,j} + \Delta_{i,j} - t^2_{i,j} + d^{mb}_{i,j}$.
- For the backlogged traffic, if some packets are further delayed to the next cycle, the maximum delay can be $\alpha_{i,j} + \Delta_{i,j} - t^2_{i,j} + d^{mb}_{i,j} + cycle$.

This optimization problem involves both sequencing and scheduling. We assume the ONU scheduling order remains the same as that in the last cycle, and focus on the scheduling problem in this paper. As shown before, $f_{i,j}$ is a quasi-linear function with respect to loss $l_{i,j}$, delay $d_{i,j}$, and jitter $v_{i,j}$. $l_{i,j}$, $d_{i,j}$, and $v_{i,j}$ are linear functions of granted bandwidth $\Delta_{i,j}$ and dropped traffic $\delta_{i,j}$. Therefore, the optimization problem is a quasi-concave maximization problem. We next present our scheme of obtaining an optimal solution to the problem.

4.3 Utility Max-Min Fair Bandwidth Allocation

With the estimation of QoS performances, OLT can perform bandwidth allocation for utility max-min fairness. We herein employ the bisection method to obtain the optimal solution of the quasiconcave utility max-min problem. The main idea is as follows: Let a be the lower bound of the utility, b be the upper bound of the utility, 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 $\delta_{i,j}$ and delayed traffic $\Delta_{i,j} - \delta_{i,j}$ which can guarantee x. If the sum of the minimum required bandwidth $\Delta_{i,j}$ 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 below.

Algorithm 1. Determine $\Delta_{i,j}$ and $\delta_{i,j}$

```
Let a = 0, b = 1, x = 1
while b − a < ε do
    calculate the maximum allowed loss ratio of each queue to ensure its corresponding utility to be above x
    calculate the maximum δ_{i,j} for each queue
    calculate the maximum delay and jitter of each queue to ensure its corresponding utility to be above x
    calculate the maximum Δ_{i,j} − δ_{i,j} for each queue
    calculate the minimum required Δ_{i,j} for each queue
    if Σ_{i,j} Δ_{i,j} < cycle then
        b = x, x = (a + b)/2
    else
        a = x, x = (a + b)/2
    end if
end while
```

In Algorithm 1, line 4 and line 6 are calculated based on the estimation discussed in Section 4.2. Line 3 and line 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. $f^1_{i,j}(x_1) = f_{i,j}(x_1, 0, 0)$, $f^2_{i,j}(x_2) = f_{i,j}(0, x_2, 0)$, $f^3_{i,j}(x_3) = f_{i,j}(0, 0, x_3)$, where $f_{i,j}(x_1, x_2, x_3)$ is the application utility function as defined in Section 3. 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.

5 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 $125\mu s$. 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 1K 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., $f_{i,j}(x_1, x_2, x_3) = f^1_{i,j}(x_1)$. For five queues in each ONU, $f^1_{i,j}(x_1)$ is defined as follows.

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

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

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

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

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

Fig. 1 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

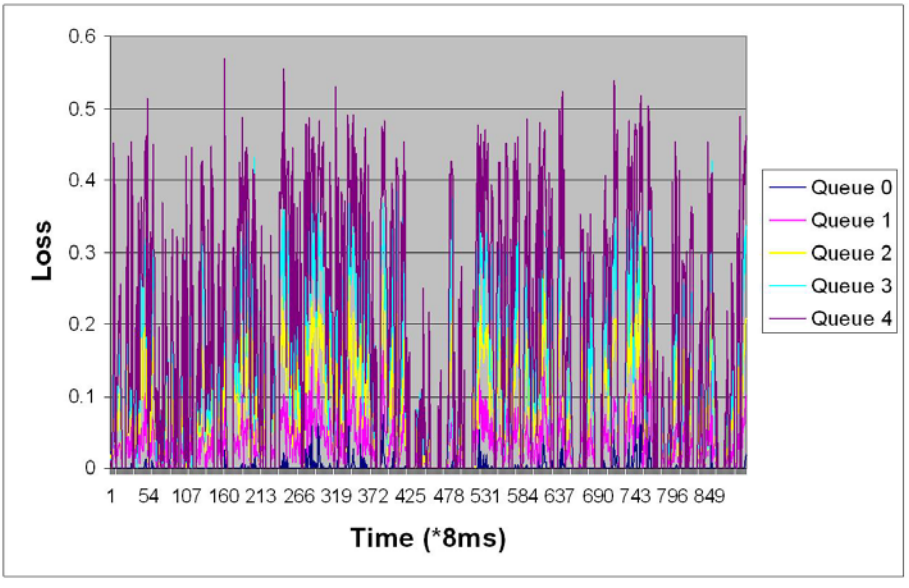

Fig. 1. Packet loss ratio vs. application utilities

3 should not exceed 0.01, 0.1, 0.2, and 0.3, respectively. From Fig. 1, 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.

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

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

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

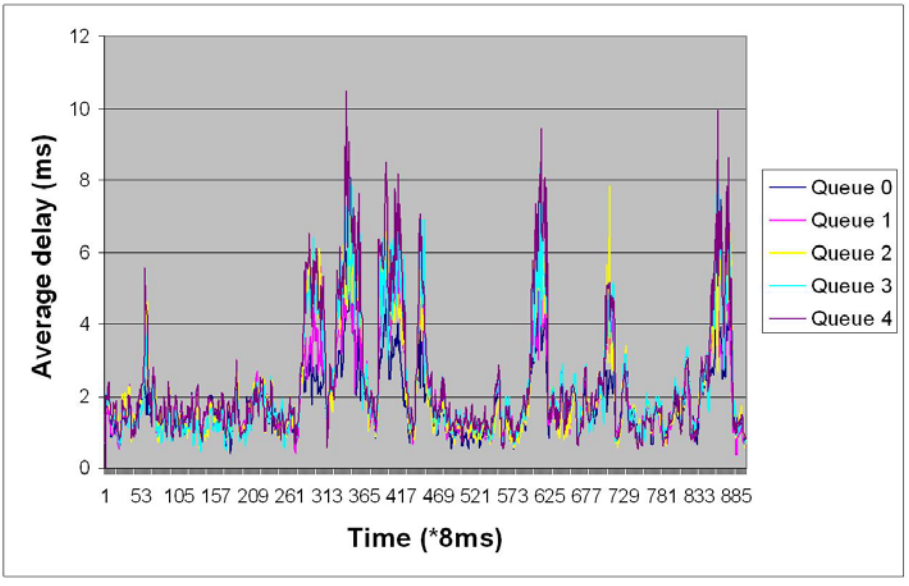

Fig. 2. Packet delay vs. application utilities

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

$$f_{i,4}^2(x_2) = \begin{cases} 1 & x_2 \leq 7ms \\ e^{(x_2-7)/7} & x_2 > 7ms \end{cases}, \forall i$$

Fig. 2 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 to around 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 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.

6 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 quasi-linear 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 quasi-concave 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. Cao, Z., Zegura, E.: Utility max-min: an Application-oriented Bandwidth Allocation Scheme. IEEE INFOCOM 2, 793–801 (1999)
2. Kramer, G., Mukherjee, B., Pesavento, G.: IPACT a Dynamic Protocol for an Ethernet PON (EPON). IEEE Communications Magazine 40(2), 74–80 (2002)
3. Kramer, G., Mukherjee, B., Dixit, S., Ye, Y., Hirth, R.: Supporting Differentiated Classes of Service in Ethernet passive optical networks. OSA Journal of Optical Networking 1(8), 280–298 (2002)
4. Assi, C., Ye, Y., Dixit, S., Ali, M.: Dynamic Bandwidth Allocation for Quality-of-Service over Ethernet PONs. IEEE Journal on Selected Areas in Communications 21(9), 1467–1477 (2003)
5. Naser, H., Mouftah, H.: A joint-ONU Interval-based Dynamic Scheduling Algorithm for Ethernet Passive Optical Networks. IEEE/ACM Transactions on Networking 14(4), 889–899 (2006)
6. Luo, Y., Ansari, N.: Bandwidth Allocation for Multiservice Access on EPONs. IEEE Communications Magazine 43(2), S16–S21 (2005)
7. Jiang, S., Xie, J.: A Frame Division Method for Prioritized DBA in EPON. IEEE Journal on Selected Areas in Communications 24(4), 83–94 (2006)
8. Kim, C., Yoo, T., Kim, B.: A Hierarchical Weighted Round Robin EPON DBA Scheme and Its Comparison with Cyclic Water-filling Algorithm. In: IEEE International Conference on Communications, pp. 2156–2161 (2007)

9. Tasaka, S., Ishibashi, Y.: Mutually Compensatory Property of Multimedia QoS. In: IEEE International Conference on Communications, vol. 2, pp. 1105–1111 (2002)
10. Takahashi, A., Yoshino, H., Kitawaki, N.: Perceptual QoS Assessment Technologies for VoIP. IEEE Communications Magazine 42(7), 28–34 (2004)
11. Nace, D., Pioro, M.: Max-min Fairness and its Applications to Routing and Load-balancing in Communication Networks: a Tutorial. IEEE Communications Surveys & Tutorials 10(4), 5–17 (2008)