Title

Fair sharing using dual service-level agreements to achieve open access in a passive optical network

Permalink

https://escholarship.org/uc/item/3mb4n542

Journal

IEEE Journal on Selected Areas in Communications, 24(8)

ISSN

0733-8716

Authors

Banerjee, A
Kramer, G
Mukherjee, B

Publication Date

2006-08-01

Peer reviewed

Fair Sharing Using Dual Service-Level Agreements to Achieve Open Access in a Passive Optical Network

Amitabha Banerjee, *Student Member, IEEE,* Glen Kramer, *Member, IEEE,*
and Biswanath Mukherjee, *Member, IEEE*

Abstract— **The Passive Optical Network (PON) is an attractive solution for high-bandwidth access networks. In the context of a broadband access network, the term *open access* implies the ability of multiple service providers to share the deployed access network infrastructure to make services available to the end users. Multiple services may thereby be delivered over a shared access channel. Open access requires fairness in terms of throughput, delay, jitter, and other network parameters in the access channel among the sharing entities, namely service providers and end users. Since the traffic in an access network is very bursty, an access network may be frequently subjected to high loads for certain durations of time. Meeting the above fairness requirements under such conditions is therefore very challenging.**

In this study, we first motivate the problem of meeting fairness requirements simultaneously to both service providers and users, which are located at opposite ends of an access channel. We then investigate the importance of two different sets of Service-Level Agreements (SLAs), which we call Dual SLAs. After formulating a mathematical model, we propose an efficient scheduling algorithm to meet Dual SLAs which is based on the well-known concept of max-min fairness.

We then demonstrate the effectiveness of our proposed algorithm through simulations using a discrete-event-simulator-based PON set-up, which compares the fairness of the Dual-SLA scheduling algorithm with that of other traditional fair queuing algorithms such as Deficit Round Robin (DRR).

Index Terms— **Passive optical network, open access, fair bandwidth sharing, dual service level agreement.**

I. INTRODUCTION

THE Passive Optical Network (PON) is a point-to-multipoint optical access network. An Optical Line Terminal (OLT) at the Central Office (CO) is connected to many Optical Network Units (ONUs) at remote ends using optical fiber and passive splitters [6], [7]. As an example of the PON, the Ethernet PON (EPON) has been standardized in the IEEE 802.3ah, which defines the Multi-Point Control Protocol (MPCP) for control and signaling messages in EPON. The standard prescribes a line rate of 1 Gbps to be shared between 16/32/64 ONUs which are located at a distance in the range of 10-20 km from the OLT. Various deployment models, Fiber-To-The-x (FTTx) (where x = B (Building), C (Curb), H (Home), N (Node), P (Premise), etc.), have been proposed in the literature for commercial deployment of EPON. FTTx specifies the level of penetration of ONUs in the access networks, e.g., whether the ONU is placed inside the home, or at the curb, or in a building. Thus, the ONU may be connected directly to an end-user as in FTTH, or through existing copper-based technologies such as Digital Subscriber Line (DSL). Ethernet technology is inexpensive and ubiquitous, and it interfaces cleanly with the technology in Local-Area Networks (LANs); hence, compared to other solutions such as Broadband PON (BPON) and Gigabit PON (GPON), EPON seems to be the most promising broadband access network technology today.

The IEEE 802.3ah standard of the EPON specifies only a control and signaling protocol mechanism; it does not mandate any specific approach towards assigning the channel bandwidth across different users. A vast number of Dynamic Bandwidth Allocation (DBA) algorithms have been proposed in the literature for allocating bandwidth in the upstream channel from the ONUs to the OLT. One of the first schemes for achieving statistical multiplexing in the upstream channel was Interleaved Polling with Adaptive Cycle Time (IPACT) [8], in which the OLT polls the ONUs individually, and issues transmission grants in round-robin fashion. Extensions to IPACT include supporting bandwidth guarantees for high-priority traffic [12], deterministic effective bandwidth (DEB) admission control [20], and many others. A good survey of DBAs may be found in [13].

In the context of an access network, a revenue model which allows an end user to freely choose its service providers (SPs), and for a SP to connect to the transport network and solicit customers for services it provides, is known as *open access* [2], [14]. Many traditional services such as voice and video are now available in the Internet Protocol (IP) domain, such as Voice over IP (VoIP) and IP-TV. In an *open access* PON model, different services provided by different SPs, and used by different users, share a common access channel. The network operator who deploys and maintains the PON network and facilitates *open access* is called an *access network operator (ANO). Open access* has emerged as a regulatory requirement for residential access networks in many regions of the world, such as several municipalities in the United States. Moreover, some community networks have employed such an economic model to serve their customers. *Open access*

Manuscript received May 9, 2005; revised December 22, 2005. This work has been supported by NSF Grant No. ANI-04-35525.

A. Banerjee, G. Kramer, and B. Mukherjee are with the University of California at Davis (e-mail: abanerjee@ucdavis.edu, glen.kramer@teknovus.com, mukherje@cs.ucdavis.edu.

Digital Object Identifier 10.1109/JSAC-OCN.2006.21805.

0733-8716/06$20.00

may be a favored model for municipality networks, particularly because municipalities own the right of way to deploy fiber and they may easily build and operate such networks. Municipalities may sign up local users, and various SPs may lease bandwidth from the municipality to provide services. In such a case, a user may avail of on or more services from one or more SPs. An example of a municipality deploying FTTH is the Palo Alto Municipality [11]. For a more elaborate discussion on economic aspects and deployment scenarios of *open access*, we refer the reader to [1].

A predominant characteristic in an access network is that a single channel (e.g., a 1-Gbps access channel in EPON) is shared across multiple users. An important difference between an access network and a LAN is that an access network serves independent, non-cooperating, and bandwidth-competing users. Also, the propagation delay between the OLT and ONU in a PON can be significantly larger than in a LAN. Hence, it is important to ensure some minimal degree of performance for each user, so that a bandwidth-hungry user does not adversely affect the performance of other users in the network. Similarly, if two competing SPs are providing the same category of services, e.g., IP-TV, then it is required that the bandwidth access to them be non-discriminatory. Most services may be available on-demand, and hence users may avail themselves of different services from different SPs at different intervals of time. Developing a set of fairness requirements for such a scenario, and meeting them, is an important problem. Moreover, traffic in an access network may be described as the multiplexing of several independent self-similar streams, which lead to bursty traffic. The network may thus be subjected to very heavy load for certain durations of time. Hence, the best-effort traffic model, which current access networks are based on, performs poorly under heavy load.

In this work, we propose and investigate the characteristics of a Dual-SLA scheduling algorithm to achieve fairness in open access for any shared access network. We then demonstrate its application in the context of an EPON access network. The outline of this paper is as follows. Section II discusses the fairness requirements for achieving *open access* in a PON. Section III formulates a mathematical model, and Section IV proposes the Dual-SLA scheduling algorithm to implement the fairness requirements. Section V studies various factors for applying the Dual-SLA algorithm in an EPON setting. Section VI demonstrates some illustrative results from simulations of the Dual-SLA scheduling algorithm. Section VII concludes our paper.

II. FAIRNESS IN AN OPEN ACCESS NETWORK

The most common form of open access in practice [1] is at the Media-Access Control (MAC) layer, or Layer 2. In this scenario, the Access Network Operator (ANO) deploys both the fiber and the link-layer electronics (OLT and ONUs for a PON) at both ends. SPs are offered a network connected to users, which they can use as a platform for delivering a bundle of services. A PON employing such a model of *open access* is shown in Fig. 1. In this scenario, a user could be a home, a multi-dwelling unit (MDU), an apartment, or an interface to a Local-Area Network (LAN). The ONUs may be located at the user home (FTTH), or be connected to the multiple users via other copper-based technologies such as DSL (FTTC, FTTB, FTTP, etc.). The OLT may be connected to a Metropolitan-Area Network (MAN), or directly to a long-haul backbone network. The issue of fairness in bandwidth allocation arises when the traffic demand exceeds the available capacity, i.e., during network congestion. For example, the instantaneous traffic arriving at the OLT destined for the ONUs may be much higher than the PON's channel capacity for certain periods of overload.

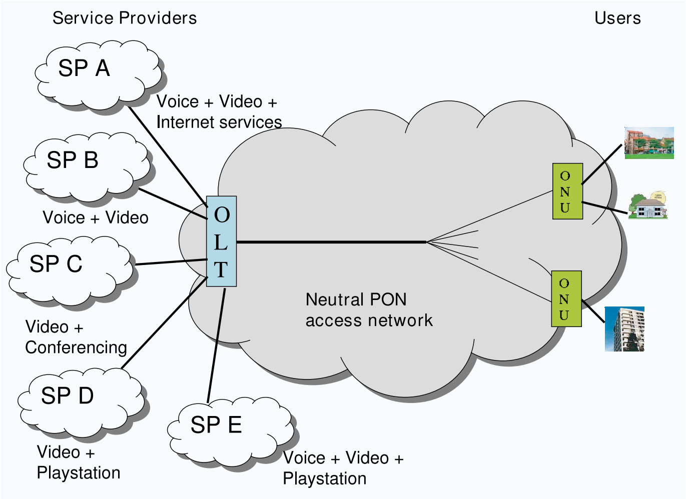


Fig. 1. A PON employing *open access*.

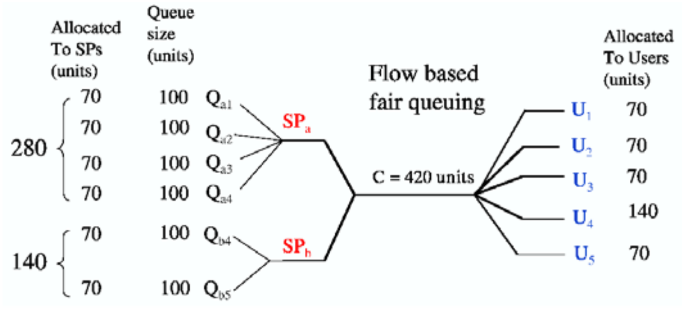
Our motivation in this work is to ensure fairness in terms of throughput in the shared access channel, among users and SPs simultaneously. Accomplishing both of the above objectives is challenging, because the users and SPs are connected to opposite ends of the access channel, which itself may be quite long (10 – 20 km) and hence is subjected to some significant propagation delay.

We refer to the connection from a SP to a user as a *flow*. For illustration of the fairness requirement, we consider an example with five users and two SPs sharing the access network. A total of 420 units of bandwidth is available in the access channel. Users U_1, U_2, U_3, and U_4 are accessing services simultaneously from one SP, SP_a. Two users U_4 and U_5 are accessing services from SP_b. Corresponding to each flow, a queue is maintained at the OLT. The queue sizes[1] for all the queues, Q_{a1}, Q_{a2}, Q_{a3}, Q_{a4}, Q_{b4}, and Q_{b5}, are 100 units each. Since the aggregate demand is greater than the available capacity, we cannot satisfy all the queues.

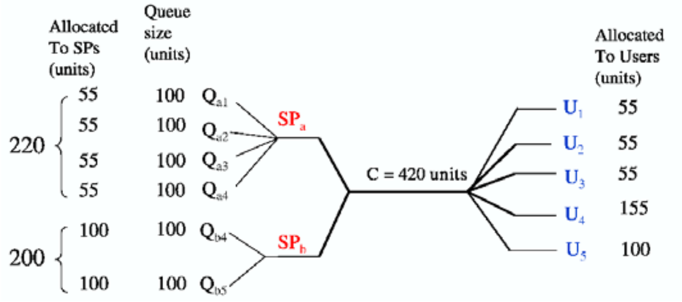
The first approach shown in Fig. 2(a) aims at fair sharing of the channel capacity, by considering all the flows independently. A number of algorithms have been proposed to achieve fair queuing across different independent network flows in the literature, e.g., Deficit Round Robin (DRR) [17], Core-Stateless Fair Queuing (CSFQ) [18], and many other packet implementations of Generalized Processor Sharing (GPS) [15].

---

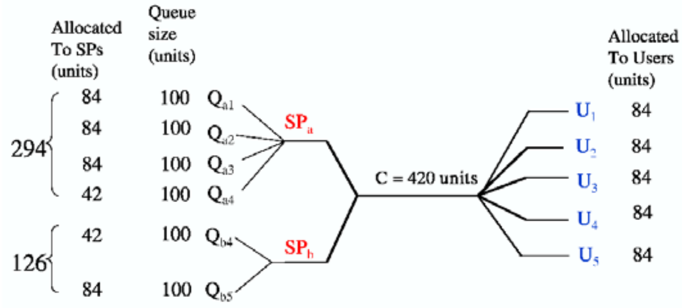
[1] The terms bandwidth, queue size, and time slot will be used interchangeably in this discussion. An PON channel is usually run in a time-division-multiplexing (TDM) fashion so, if its cycle time is 420 units, we say that the PON bandwidth is 420 units or it consists of 420 time slots. Backlogged traffic in queues at the OLT will need to use these time slots to travel to the ONUs. So, if a queue size is 100, it means that the OLT requires a bandwidth of 100 units, or 100 time slots, to transmit this information on the PON channel.



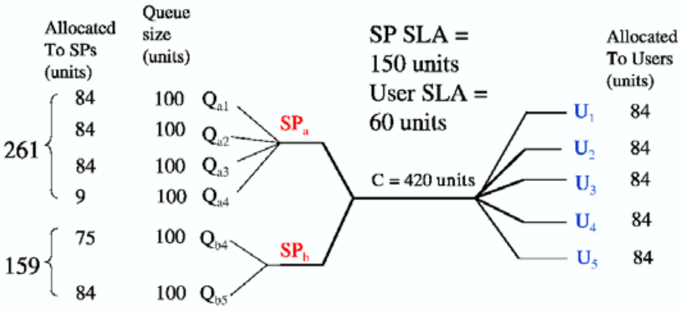
(a) Independent flows



(b) SP SLA



(c) User SLA



(d) Dual SLA

Fig. 2. Bandwidth allocation using different schemes.

Such fairness algorithms are useful for switches or routers which consider all the outgoing connections uniformly. The disadvantage of using such a flow-based fair-queuing model in an access network is that we wish to be fair to both SPs and users, and not just to flows. In Fig. 2(a), we observe that U_4 is allocated twice the bandwidth than other users, because it has two active flows. Similarly, SP_a is allocated twice the bandwidth than SP_b, because it has four active flows compared to two for SP_b. Such a scheme is not suitable for an access network because a user subscribing to a large number of services may deprive other users of bandwidth. Similarly, a SP having low volume of customers may be denied fair competition from a SP carrying a high volume of customers.

In Fig. 2(b), we show a solution in which the bandwidth is shared fairly among the SPs. This solution doesn't achieve good fairness across users, e.g., now, U_4 is allocated approximately three times more bandwidth than U_1, U_2 and U_3. In Fig. 2(c), we show a solution in which the bandwidth is shared fairly among the users. In this case, while users get uniform bandwidth, SPs do not.

From the above observations and discussions, we are motivated to investigate Dual Service-Level Agreements (SLAs), i.e., SLAs defined for both SPs and users in one system. A SLA provides a guarantee on the network performance of a channel, in terms of various metrics such as minimum throughput, delay, jitter, losses, etc. In the context of the discussion here, we limit ourselves to SLAs with minimum throughput specifications. SLAs in the context of SPs (SP SLAs) will enable a SP to provide Quality of Service (QoS)-aware services to its customers. For example, let us consider SP_a which provides a Video-on-Demand (VoD) service, which would require R_a bps average bandwidth per customer. Suppose SP_a negotiates a SLA of W_{MIN} bps guaranteed bandwidth. Then, SP_a would have a guarantee from the network operator to provide reasonable QoS to W_{MIN}/R_a users at any time, on average. If the number of users simultaneously requesting VoD from SP_a is greater than this amount, SP_a may block some requests to provide good service to the other users already in the system. Similarly, providing minimum guarantees to users (by having User SLAs) will allow the users reasonable services and would not starve any user out of bandwidth. Thus, the motivation behind Dual SLAs is to ensure minimum guarantees of bandwidth to both users and SPs under sudden bursts of high-load conditions, so that every entity receives some minimum degree of service.

A solution based on such a scheme with a chosen User SLA of 60 units per user and SP SLA of 150 units per SP is shown in Fig. 2(d), and the corresponding algorithm is described in detail in Section IV. We observe that all the users as well as the SPs are granted at least the bandwidth specified by the Dual SLA. Because we ensure fairness between users and SPs only, each flow (corresponding to a service) may not get uniform bandwidth allocation. For example, in Fig. 2(d), Q_{a4} receives an allocation of only 9 units for a demand of 100 units. This is primarily because U_4 requested services from two SPs, and during congestion, it receives good service from at least one of them. A further dimension to the example may be to introduce fairness between individual flows, once the SP SLA and the User SLA have been met, i.e., fairness across three dimensions, users, SPs, and queues at the same time. In this work, we focus our attention on the problem of achieving fairness between users and SPs using Dual SLAs, because this is a complex and important problem by itself. We shall investigate algorithms to meet the three-dimensional fairness problem, involving queues as well, in future work.

With respect to Dual SLAs, we must ensure that we do not oversubscribe the User SLAs or the SP SLAs, i.e., the sum of the User SLAs must be less than the channel capacity; similarly, the sum of the SP SLAs must be less than the channel capacity as well. However, the User SLAs and the SP SLAs are independent. Hence, it may not be always possible to allocate bandwidth in such a way so as to meet both sets of

SLAs. Therefore, we need two levels of SLA. The *primary SLA* is defined to be the one whose specified minimum guarantees must be given the highest priority to be met. After the *primary SLA* has been met, the next priority is to meet the *secondary SLA*. At high traffic loads and depending on the load distribution, when the secondary SLA for all the entities is not met, our definition of fairness requires that the deficits against the SLA for the corresponding entities be uniform. If a secondary SLA is not met in one time interval of bandwidth allocation, the corresponding secondary SLA is increased by the amount of deficit in the next time interval. By carrying over the deficits against the secondary SLA over subsequent time intervals, we ensure that the bandwidth defined by the secondary SLA shall be guaranteed if an aggregate amount of time were considered.

In the context of an *open access network*, the *access network operator (ANO)* must negotiate the SLA contracts with the SPs and users. The ANO is responsible for ensuring that the SLAs are not oversubscribed. SPs requiring higher throughput in the network may negotiate the corresponding SLA in return for paying an additional price for the same. The same is also true for users (It is expected that users will have a number of pre-determined service-level offerings to choose from). The ANO is responsible for ensuring that the SLAs are met. The Dual-SLA scheduling algorithm that we propose is one such algorithm which the ANO may employ in the system. An example of an ANO would be a municipality.

In this work, we describe the Dual-SLA scheduling algorithm in the context of downstream operation of a PON, in which packets are broadcast from the OLT to the ONUs. Likewise, the algorithm may be applied to the upstream operation of a PON, and to downstream and upstream operations of other access networks.

III. A MATHEMATICAL MODEL FOR DUAL SLAS

We develop a mathematical model for fair queuing in a PON based on Dual SLAs. We state the problem as follows.

Given:

1) R: Channel rate of the PON.
2) M: Number of SPs in the system.
3) N: Number of users in the system.
4) W_i^{MIN}: Minimum bandwidth guaranteed to SP i by the corresponding SP SLA, $i \in 1, \ldots, M$. This SLA may be either the primary SLA or the secondary SLA.
   We require that:
   $$\sum_{i=1}^{i=M} W_i^{MIN} < R, \tag{1}$$
   so that the SLAs are not oversubscribed.
5) U_j^{MIN}: Minimum bandwidth guaranteed to user j by the corresponding User SLA, $j \in 1, \ldots, N$. This SLA is either primary or secondary, depending on what the SP SLA is. Similarly:
   $$\sum_{j=1}^{j=N} U_j^{MIN} < R \tag{2}$$
6) T: Maximum time cycle duration. The scheduler at the OLT may schedule bandwidth for a maximum duration of T in one iteration of scheduling. This time is limited by the maximum scheduling delay in a PON, as will be explained later.

Define:

1) C: Total capacity available in a time cycle of maximum duration T, i.e.,
   $$C = R * T \tag{3}$$
   The capacity C represents the total bytes available for scheduling in a time cycle of maximum time duration T. For simplicity in the mathematical formulation, we ignore various overheads in a system such as the inter-frame guard time, control-plane overhead, etc., and we assume that ideally 100% throughput may be achieved in the system. The above equation may be suitably adjusted to represent the actual available capacity.
2) w_i^{MIN}: Minimum guaranteed bandwidth by the SLA to SP i in a time cycle of maximum duration T, i.e.,
   $$w_i^{MIN} = W_i^{MIN} * T \tag{4}$$
3) u_j^{MIN}: Minimum guaranteed bandwidth by the SLA to user j in a time cycle of maximum duration T, i.e.,
   $$u_j^{MIN} = U_j^{MIN} * T \tag{5}$$
4) $q_{i,j,t}$: Queue size for the flow between SP i and user j at time t at which the scheduling algorithm in invoked. In case of downstream traffic, the queue size corresponds to queues located at the OLT. In case of upstream traffic, the queues are located at the remote entities, i.e., ONUs. Each queue corresponds to a service.
5) T_{adv}: Minimum time by which the scheduler must advance. For example, when there is no traffic in any of the flows, the scheduler waits for at least T_{adv} before invoking the scheduling algorithm again.

Determine:

1) Δ: Scheduling time cycle duration.
2) $g_{i,j,t,t+\Delta}$: Bandwidth allocated in time interval t to $t + \Delta$, by the scheduler invoked at time t, to the corresponding flow. We define this as the *allocated time-slot* for the corresponding queue. The scheduler is invoked at time t to calculate the allocated time-slot in the time cycle of duration Δ. Figure 3 describes how the scheduler is invoked at various instants of time.

Constraints:

1) *Maximum bandwidth allocation constraint*:
   $$g_{i,j,t,t+\Delta} \leq q_{i,j,t} \quad \forall i, j \tag{6}$$
   which states that the bandwidth allocated must be less than or equal to the queue size. In other words, the scheduler is work conserving.
2) *Capacity constraint*:
   $$\sum_{i=1}^{i=M} \sum_{j=1}^{j=N} g_{i,j,t,t+\Delta} \leq C \tag{7}$$

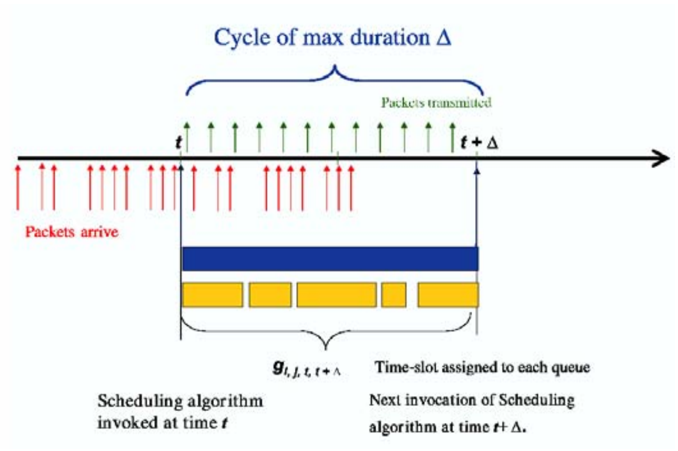


Fig. 3. Invocation of the scheduling algorithm for time-slot computation.

which states that the sum of all bandwidths allocated must be less than or equal to the capacity available for the maximum time cycle duration.

3) *Scheduling time cycle constraint*:

$$T_{adv} \leq \Delta \leq T \tag{8}$$

which states that the scheduling time cycle must be less than the maximum scheduling time cycle and greater than the minimum time by which the scheduler must advance.

Objectives:

The following objectives are in order of priority.

1) Meet the primary-SLA bandwidth requirement.
2) Try to meet the secondary-SLA bandwidth requirement. If it is not possible to meet all the secondary SLAs, then the deficit in meeting the secondary SLAs should be as uniform as possible.
3) Divide any surplus bandwidth after meeting SLAs fairly across the primary-SLA entities, and then correspondingly for each secondary-SLA entity.

IV. DUAL-SLA SCHEDULING ALGORITHM

We present the scheduling algorithm to implement our mathematical model formulated in Section III. For describing the algorithm, we choose an example in which the primary SLA is for the users, and the secondary SLA is for the SPs, although this may easily be reversed. The scheduling algorithm is invoked at time t, and it considers the given parameters as defined in the previous section. The algorithm is presented in the flowchart in Fig. 4. The following two cases arise.

Case I: Demand is less than capacity available in the maximum time cycle duration, i.e.,

$$if \quad \sum_{i=1}^{i=M} \sum_{j=1}^{j=N} q_{i,j,t} \leq C \tag{9}$$

then the demand may be met entirely using the available capacity, and the allocated time-slots may be determined by the following two equations:

$$g_{i,j,t,t+\Delta} = q_{i,j,t} \quad \forall \quad i \in 1,\ldots,M, \quad j \in 1,\ldots,N \tag{10}$$

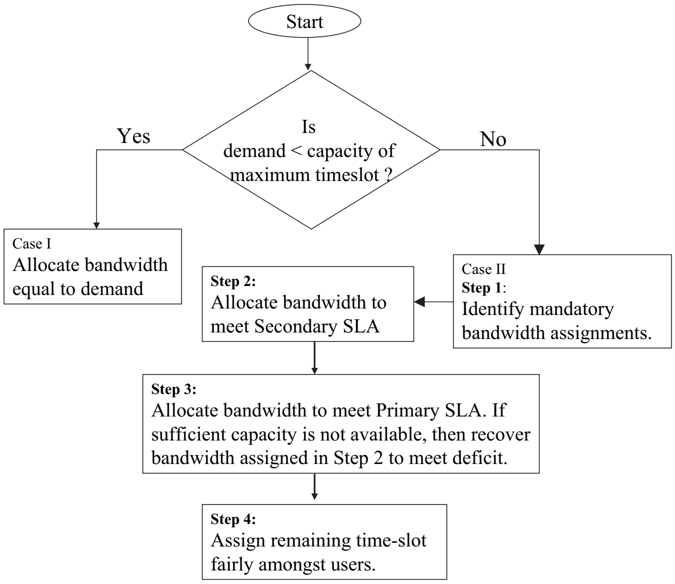


Fig. 4. Flowchart for Dual-SLA scheduling algorithm.

$$\Delta = Max \quad [T_{adv}, \sum_{i=1}^{i=M} \sum_{j=1}^{j=N} q_{i,j,t}/R] \tag{11}$$

The latter equation states that we would like to advance the scheduler by the larger of T_{adv} and the actual time that was taken to transmit the packets in the queues. The above equation assumes no additional overhead in the system, as mentioned before.

Case II: Demand is greater than available capacity. This is the interesting case where the role of fair scheduling comes in. The primary SLA is specified for either the users or the SPs, and depending on the choice of the primary SLA, the secondary SLA is specified for the opposite entity. In the following description of the algorithm, we assume that the primary SLA is for users, and the secondary SLA is for SPs.

Step 1: Identify Mandatory Bandwidth Allocations. The objective is to identify the following two situations:

1) Users whose cumulative demand is less than the primary SLA requirement, i.e., users j, for which the following equation is true:

$$\sum_{i=1}^{i=M} q_{i,j,t} < u_j^{MIN} \tag{12}$$

In such a case, the entire demand for that user must be met.

2) Users who do not satisfy the above, but who are currently subscribing to only a single SP. For such users, a grant equal to the primary SLA (u_j^{MIN}) must be assigned, because there is no alternate competing SP.

Steps 2, 3, and 4 use algorithm **Allocate Max-Min Fair Bandwidth** shown in Fig. 5. Given some available bandwidth, the objective is to ensure distribution of the bandwidth among a set of entities (SPs/ users) to achieve *max-min fairness*. A feasible allocation set $[x_1, x_2, x_3, \ldots, x_r]$ is said to be *max-min* fair [3] when it is impossible to increase the allocation to an entity r without losing feasibility or decreasing the allocation of any entity $r^{'}$ with allocation $x_{r^{'}} < x_r$. Simply put, a *max-min* fair allocation gives the most-poorly-treated entity (i.e., the entity which has received the least allocation) the largest share.

```
ALGORITHM Allocate Max-Min Fair Bandwidth
Given:
  1) $availableBW$: The available bandwidth which needs to be fairly distributed among several entities (SPs or users).
  2) $demand[]$: The demand array of the respective entities.
  3) $preAssigned[]$: Bandwidth already assigned to the entities.
  4) $size$: The number of entities.
To compute:
  1) $allocated[]$: Bandwidth allocated by the algorithm to the respective entities out of $availableBW$.
Algorithm:
  1) Sort preAssigned[] in increasing order; and arrange the other arrays in this sorted order.
  2) Let count = 1.
  3) For the first count items ($i$ is the loop variable):
     a) Define $currentAlloc[i] = preAssigned[i] + allocated[i]$
        Increment $allocated[i]$ by
        $MIN \begin{cases} demand[i] - currentAlloc[i] \\ demand[count+1] - currentAlloc[i], (when \quad count < size) \\ availableBW/count. \end{cases}$
     b) Reduce $availableBW$ by the corresponding amount.
  4) Increment count by 1 and repeat sub-step 3 above, until $availableBW = 0$ or $count = size + 1$.
  5) Return $allocated[]$ and $availableBW$.
Running time complexity = $O(size^2)$.
```

Fig. 5. Algorithm **Allocate Max-Min Fair Bandwidth**.

Step 2: Meet Secondary SLA. Let us consider the following:

$$availableBW = C - \sum_{i=1}^{i=M} \sum_{j=1}^{j=N} g_{i,j,t,t+\Delta} \tag{13}$$

$$demandSP[i] = MIN \begin{cases} \sum_{j=1}^{j=N} q_{i,j,t} \\ w_i^{MIN} \end{cases} \forall i \in 1, \ldots, M. \tag{14}$$

$$preAssignedSP[i] = \sum_{j=1}^{j=N} g_{i,j,t,t+\Delta} \quad \forall i \in 1, \ldots, M. \tag{15}$$

$$sizeSPs = M \tag{16}$$

We use algorithm **Allocate Max-Min Fair Bandwidth** to compute $allocatedSP[]$, i.e., the max-min fair distribution of $availableBW$ among SPs. The demand of the SPs is given by Eqn. (14). The demand for a SP is the minimum of the SP SLA and the queue size, thus ensuring that the secondary SLA is met in this step, if sufficient bandwidth were available. Let $allocatedSP[]$ be the bandwidth allocated by the algorithm to the SPs. Once $allocatedSP[]$ has been determined, we distribute $allocatedSP$ for each SP among its users using the same algorithm **Allocate Max-Min Fair Bandwidth**, thus computing $g_{i,j,t,t+\Delta}$ $\forall$ i, j. Note that, since the demand is constrained by the SP SLA, if all the SP SLAs were to be met, $availableBW$ would be surplus.

Step 3: Meet Primary SLA. We use the surplus $availableBW$ from Step 2 to meet the primary SLA. We compute the following:

$$demandUser[j] = MIN \begin{cases} \sum_{i=1}^{i=M} q_{i,j,t} \\ u_j^{MIN} \end{cases} \forall j \in 1 \ldots N. \tag{17}$$

$$preAssignedUser[j] = \sum_{i=1}^{i=M} g_{i,j,t,t+\Delta} \quad \forall j \in 1, \ldots, N. \tag{18}$$

$$sizeUsers = N \tag{19}$$

We use algorithm **Allocate Max-Min Fair Bandwidth** to distribute $availableBW$ among the users. The demand, as defined by Eqn. (17), considers the minimum of queue size and primary SLA for an user, the objective being to meet the primary SLA at least. After this step, if the returned $availableBW = 0$, it implies that the primary SLA requirement may not have been satisfied for some of the users owing to sufficient capacity not being available. Since SLAs are not oversubscribed, it implies that some users have been granted bandwidth in excess of the primary SLA in the computations in Step 2. Our objective is to satisfy the primary SLA requirement by *recovering* bandwidth from these users. While recovering and reassigning bandwidth, secondary SLAs may be violated, and hence the aim is to be fair to SPs in this process so that the deficits from the specified secondary SLA are uniform. Algorithm **Recover Bandwidth** shown in Fig. 6 is invoked for each user whose primary SLA is not met. In the next subsection, we discuss the fairness of Algorithm **Recover Bandwidth**.

```
ALGORITHM Recover Bandwidth ($User \quad j$)
Algorithm:
  1) Calculate: $deficit = u^{MIN} - \sum_{i=1}^{i=M} g_{i, \, , \, , +\Delta}$
  2) Sort SPs in descending order of cumulative bandwidth granted to them, i.e. $\sum_{=1}^{=N} g_{i, \, , \, , +\Delta}$. Similarly, sort users in descending order of cumulative bandwidth granted to them, i.e., $\sum_{i=1}^{i=M} g_{i, \, , \, , +\Delta}$. During the course of the algorithm, the above two sorted lists must be maintained efficiently[a] as bandwidth granted is shifted around.
     a) In the sorted order of SPs as defined by the above list, call ALGORITHM Recover Bandwidth From Same SP ($SP \quad i, \quad User \quad j$)
     b) If $deficit > 0$ after the previous step, then call ALGORITHM Recover Bandwidth From Diff SPs ($User \quad j$).

[a] Whenever $g_{i, \, , \, , +\Delta}$ is changed, the new positions of SP $i$ and User $j$ in their respective sorted lists must be computed. This may be done in $O(M)$ and $O(N)$ time-complexities, respectively.
```

Fig. 6. Algorithm **Recover Bandwidth**.

Fairness of ALGORITHM Recover Bandwidth

Lemma 1: Algorithm **Recover Bandwidth From Same SP** is fair.

Proof*: Algorithm* **Recover Bandwidth From Same SP** *attempts to reallocate bandwidth in multiple iterations. In each*

```
ALGORITHM Recover Bandwidth From Same SP ($SP\ \ i,\ \ User\ \ j$)
Algorithm:
The idea here is to shift the allocated bandwidth from one user having surplus to the other having a deficit withing the same SP. Since the allocated bandwidth to the SP remains constant in this step, no secondary SLA gets violated. Given a user $j$, SP $i$, and $deficit$, the algorithm tries to recover previously-granted bandwidth until $deficit$ using a maximum quanta of reallocation $r$, for SP $i$ to all users except $j$. Any primary SLA that has already been satisfied is not violated in this process. The algorithm is described below.
  1) Choose user $j' \neq j$, which has been assigned the maximum bandwidth.
  2) If $\sum_{i=1}^{i=N} g_{i,\,',\,,\,+\Delta} - r > u_{'}^{MIN}$, then
      a) Reduce $g_{i,\,',\,,\,+\Delta}$ by the smaller value between $r$ and $deficit$.
      b) Increase $g_{i,\,,\,,\,+\Delta}$ by the corresponding value from the above step.
      c) Reduce $deficit$ by the corresponding amount.
  3) Repeat the above sub-steps until $deficit = 0$ or no bandwidth can be recovered from any of the users for SP $j$.
```

Fig. 7. Algorithm **Recover Bandwidth From Same SP**.

```
ALGORITHM Recover Bandwidth From Diff SPs ($Deficit$, $User\ \ j$)
Algorithm:
The idea here is to shift bandwidth from one user having surplus to another user having deficit. The bandwidth may be shifted from one SP to another. Hence, in this process, some secondary SLAs may be violated.
  1) Let $recoveredBandwidth = 0$.
  2) Choose a SP $i$ which has been assigned the maximum bandwidth from the sorted list of SPs. From the users to which this SP has an active flow, choose the user $j'$ which has been assigned the maximum bandwidth from the sorted list of users.
  3) If $\sum_{i=1}^{i=N} g_{i,\,',\,,\,+\Delta} - r > u_{'}^{MIN}$, then
      a) Reduce $g_{i,\,',\,,\,+\Delta}$ by the smaller value among $r$ and $deficit$.
      b) Increase $recoveredBandwidth$ by the corresponding value from the above step.
      c) Reduce $deficit$ by the corresponding amount.
  4) Repeat the above sub-steps until $deficit = 0$.
  5) Distribute the $recoveredBandwidth$ across all SPs having an active connection to user $j$ using ALGORITHM Allocate Max-Min Fair Bandwidth.
```

Fig. 8. Algorithm **Recover Bandwidth From Diff SPs**.

iteration, it recovers a maximum amount of r bytes from the user which has been assigned the maximum bandwidth. This amount r is defined as the maximum quanta of reallocation. *If $r = 1$, then this ensures that bandwidth has always been recovered from the user having the maximum bandwidth. Moreover, since Algorithm* **Recover Bandwidth From Same SP** *reallocates the already-assigned bandwidth only within a particular SP, the total bandwidth assigned to the SP remains constant. Hence, the above algorithm is fair with respect to both users and SPs.*

Lemma 2: Algorithm **Recover Bandwidth From Diff SPs** is fair.

Proof*: Algorithm* **Recover Bandwidth From Diff SPs** *recovers a maximum amount of r bytes in one iteration from the (SP, user) pair which has been assigned the maximum bandwidth. If $r = 1$, then this ensures that bandwidth is always recovered from the (SP, user) pair with the maximum bandwidth. The recovered bandwidth is then distributed among all active SPs of user j using Algorithm* **Allocate Max-Min fair Bandwidth**. *Hence, the algorithm is fair.*

Thus, if $r = 1$, the algorithm is fair to both SPs and users using the above two lemmas. However, this may lead to a high time complexity for the algorithm since only one byte may be reallocated in one iteration. Therefore, we propose a trade-off between fairness and time complexity, by increasing the value of r.

The three most-frequent packet sizes observed from measurement of downstream Ethernet traffic in CATV networks, as reported in the literature [16], are 64, 594, and 1518 (including the 18-byte Ethernet packet header). Ethernet packets are non-divisible, and an integral number of Ethernet packets must be fit into the allocated bandwidth. Hence, for a Dual-SLA implementation in an EPON, we propose that the value of r be set to any of the above sizes, or multiples of them, so that bandwidth is shifted in units of Ethernet packet sizes. In Section VI, we investigate in greater detail the fairness and time-complexity trade-off.

Step 4: Distribute Surplus Bandwidth. If surplus bandwidth is still available after Step 3, it implies that the primary and secondary SLAs have been met. Any surplus bandwidth is distributed among users using ALGORITHM **Allocate Max-Min Fair Bandwidth**. For each user, the bandwidth is divided among SPs uniformly.

V. APPLICATION TO AN ETHERNET PON

In this section, we discuss the various aspects of the application of the proposed Dual-SLA scheduling algorithm to an EPON. Section V-A provides a solution for handling non-fragmentable Head-of-Line (HOL) packets in an EPON. Section V-B discusses fairness with respect to the secondary SLA. Section V-C discusses various approaches to handle different priorities of traffic. Section V-D discusses the run-time complexity of the Dual-SLA scheduling algorithm, and its impact on the EPON performance. Sections V-E and V-F discuss the application of the scheduling algorithm to downstream and upstream operation of the EPON, respectively.

A. Unused Time-Slot due to Head-of-Line (HOL) Blocking

The Dual-SLA scheduling algorithm described above considers an idealistic setting of fluid flows similar to that in the Generalized Processor Sharing (GPS) algorithm. However, Ethernet packets are non-fragmentable. Therefore, it is possible that the head-of-line (HOL) packet in a queue cannot be transmitted in the remaining time-slot for a queue. This leads to an unutilized portion of the allocated time-slot. Although the unutilized amount for a queue is less than the Ethernet Maximum Transfer Unit (MTU) of 1518 bytes, the channel utilization may be significantly lower if unused time-lots were left for multiple queues. We borrow the idea of deficit counters from the Deficit Round Robin (DRR) algorithm to track the usage of these unused remainders.

After the first iteration of sending packets which fit in the allocated time-slot to each queue, the cumulative unused time-slot for the OLT (in downstream direction) and each ONU (in upstream direction) is computed. In the second iteration of sending packets, the HOL packets from the queues chosen in descending sorted order of their unused granted time-slot remainders are transmitted in the unused time-slot. After all possible HOL packets have been transmitted, a new deficit value for each queue q is computed using the following equation:

$$deficit_q = actualTransmittedBytes_q - allocatedBytes_q \tag{20}$$

If the value of $deficit_q$ is positive, it means that queue q transmitted more bytes than the fair allocation. If the value of $deficit_q$ is negative, it implies that the queue could not use its fair allocation.

The value of $deficit$ is carried over to the next round of scheduling computation. For each queue, $allocatedBytes$ is modified by the amount of $deficit$. Thus, for queue q, the bytes allocated by the Dual-SLA scheduling algorithm in the next time cycle of scheduling is revised as follows:

$$allocatedBytes_q = MAX \begin{cases} allocatedBytes_q - deficit_q \\ 0 \end{cases} \tag{21}$$

Using the above approach, the bandwidth utilization of an EPON system is significantly improved.

B. Fairness with Respect to the Secondary SLA

For some traffic demands, the Dual-SLA scheduling algorithm may not meet the secondary SLA for all SPs. The minimum guaranteed bandwidth w_i^{MIN} corresponding to the secondary SLA for the next time-slot of computation may be altered to reflect the deficit in meeting the secondary SLA of the previous computation. It is also important to ensure that, after modification of the SLAs, the SLAs are not oversubscribed; therefore, the sum of guaranteed bandwidths must not exceed the available channel capacity. We define a *max threshold factor* of Γ by which the secondary SLAs may be adjusted so as to avoid the effects of short-term unfairness due to deficits corresponding to the secondary SLA. ALGORITHM **Adjust Secondary SLA** to alter the secondary SLA for an entity is presented in Fig. 9.

```
ALGORITHM Adjust Secondary SLA
Given:
  1) $deficitSecondarySLA[]$: The deficits with respect to
     the secondary SLA for the previous cycle of scheduling.
  2) $\Gamma$: The maximum factor by which the secondary SLAs
     may be increased in the next cycle of scheduling.
  3) $secondarySLA[]$: The guaranteed bytes in a time cycle
     of duration $T$ by the secondary SLA.
Compute:
     $adjustedBytes[]$: The number of bytes by which the
     secondary SLA may be increased, for the next cycle of
     scheduling.
Algorithm:
  1) Calculate:
       $availableBytes = \sum_i secondarySLA_i * \Gamma$     (22)
  2) Use Algorithm Allocate Max-Min Fair Bandwidth to
     compute $adjustedBytes[]$, providing the inputs as:
       a) $availableBW = availableBytes$
       b) $demand[] = deficitSecondarySLA[]$
       c) $preAssigned[] = secondarySLA[]$
```

Fig. 9. Algorithm **Adjust Secondary SLA**.

C. Priority-Based Traffic

Since the Dual-SLA scheduling algorithm assumes fluid flows, it does not accommodate priorities in packet-based traffic. Most services can be broadly classified into two categories, Constant Bit Rate (CBR) and Variable Bit Rate (VBR). CBR traffic is delay and jitter sensitive, and has a constant traffic demand, e.g., voice traffic. VBR traffic has varying traffic demand. Different priorities may be assigned to VBR traffic.

Since CBR traffic is predictable, the corresponding grants may be assigned at the beginning of each cycle. Each user may be granted an amount until the User SLA for CBR traffic is met. Once the CBR grants are made, the Dual-SLA algorithm may be invoked for the remaining bytes in the cycle, taking into account the grants already made.

Different queues may be maintained for different priority classes of VBR traffic. In a given time-slot, packets from a lower-priority queue may be transmitted only after packets from a higher-priority queue have been transmitted. A more detailed discussion in the context of upstream traffic on an EPON may be found in [9].

D. Execution time of the Dual-SLA Scheduling Algorithm

The Dual-SLA scheduling algorithm considers the queue sizes at time t as an input. Let the time required to execute the scheduling algorithm be T_s. If we were to consider the transmission of packets in the downstream direction, the first packet would be transmitted only at time $t + T_s$. Hence, T_s imposes an additional overhead on the packet latency in the access network. Therefore, an additional requirement of the scheduling algorithm is to have an efficient run-time complexity in a hardware implementation.

The worst case time-complexity analysis of various steps of the Dual-SLA Scheduling Algorithm (assuming $N > M$) is summarized in Table I. For ALGORITHM Recover Bandwidth, we assume that the number of iterations for recovery of bandwidth per step is a small constant, which is true when higher values of the *maximum quanta of reallocation* (r) are chosen. The analysis is for a fully-connected scenario in which each user is connected to every SP. The worst case time complexity for the Dual-SLA Scheduling Algorithm is thus $O(MN^2)$. This high time complexity is mainly due to ALGORITHM Allocate Max-Min Fair Bandwidth having a running time complexity of $O(size^2)$. The number of ONUs in a typical PON deployment is 16/32. If we consider one user connected to each ONU, then $O(MN^2)$ may be considered a reasonable worst-case time complexity. However, if multiple users were connected to an ONU, the scalability of the Dual-SLA Scheduling Algorithm may be an important concern.

TABLE I

WORST-CASE TIME COMPLEXITY OF THE DUAL-SLA SCHEDULING ALGORITHM.

Algorithm Step	Worst-Case Time Complexity
Case I: Demand less than Capacity	$O(MN)$
Case II, Step 1: Identify Mandatory Bandwidth Allocations	$O(MN)$
Case II, Step 2: Meet Secondary SLA	$O(MN^2)$
Case II, Step 3: Meet Primary SLA	$O(N^2 + NM^2)$
Case II, Step 3: ALGORITHM Recover Bandwidth	$O(MN)$
Case II, Step 4: Distribute Surplus Bandwidth	$O(N^2 + NM^2)$

TABLE II

PARAMETERS FOR SIMULATION OF THE EPON SYSTEM.

Variable	Description	Value
K	Number of ONUs in the EPON	16
M	Number of SPs	6
N	Number of users (One user connected to each ONU)	16
R	Line rate of EPON	1 Gbps
L	Line rate of traffic generated	100 Mbps
Q	Maximum buffer size in each queue	1 MB
T	Maximum scheduling time cycle	500 μs
T_{ad}	Minimum advance time of scheduler	200 μs
Γ	Maximum threshold factor for adjustment of the secondary SLA	0.2

Approximations of ALGORITHM Max-Min Fair Bandwidth, such as those reported in [4], may be considered in such a scenario to improve the scalability of the algorithm.

E. Downstream Operation

In downstream direction, packets are broadcast by the OLT to all ONUs. Once the time-slots granted to each queue ($g_{i,j,t,t+\Delta}$) are computed by the Dual-SLA scheduling algorithm invoked at time t, packets may be transmitted for each queue in a Deficit Round Robin (DRR) sequence, until the time-slot allocated to the queue is filled. The idea of using DRR is to achieve fairness in delay across each flow in the system.

A packet arriving during time interval t to $t + \Delta$ at the OLT may be delayed by a maximum duration of Δ before the scheduling algorithm runs and schedules it. The value of Δ is limited by the maximum scheduling time cycle T. The ITU G.114 [10] standard recommends that the maximum tolerable delay for voice traffic in an access network be 1.5 ms We believe that a good choice of T would be 500 μs so that a high-priority voice packet is not subjected to a scheduling delay greater than 500 μs.

F. Upstream Operation

In upstream direction, the queues are located at the remote ONUs. The queue sizes may be collected by the OLT using REPORT messages defined in the Multi-Point Control Protocol (MPCP) [5]. Once the time-slots are computed by the Dual-SLA scheduling algorithm, the grants for each ONU are broadcast by the OLT using GATE messages [5]. Corresponding to the grant, each ONU transmits the packets for each queue in Deficit Round Robin (DRR) sequence. The idea is to achieve fairness in each flow for each ONU.

TABLE III

TRAFFIC FROM SPS TO USERS.

Time	SPs	User Set	Average Traffic Rate	Aggregate Load
0s–120s	1	1–9 (Set I) + 10–12 (Set II) + 13–14 (Set III) + 15–16 (Set IV)	40 Mbps	0.64
20s–120s	2, 3, and 4	10–12 (Set II)	50 Mbps	1.09
40s–120s	5	13–14 (Set III)	75 Mbps	1.24
60s–120s	6	15–16 (Set IV)	75 Mbps	1.39

VI. ILLUSTRATIVE NUMERICAL EXAMPLES

We simulate an open access system on an EPON using a Java-based discrete-event simulator. The various parameters of the EPON are shown in Table II. We consider one user connected to each ONU (based on the FTTH model). The OLT may be connected to SPs via a MAN or long-haul backbone network, whose capacity is much greater than the EPON line rate.

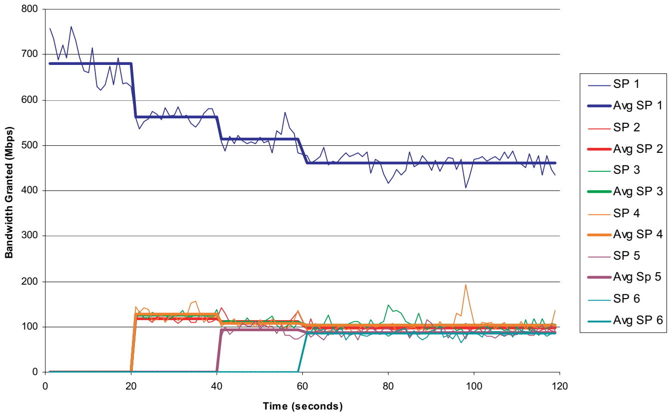
Results are for shown for simulations of downstream traffic. We do not demonstrate the results for simulation of upstream traffic here because of space constraints, and also because they are along the same lines as that of the downstream traffic. The overhead of the control messages of the MPCP protocol is assumed to be insignificant.

The traffic pattern considered is depicted in Table III. All traffic is generated to be self-similar with a Hurst parameter of 0.8 [19] and an average traffic rate as specified in Table III. The fifth column (Aggregate Load) depicts the cumulative average normalized load (with respect to the EPON line rate of 1 Gbps). The three main sizes of Ethernet packets observed in CATV networks are 64, 592, and 1518 bytes (including the 18-byte Ethernet packet header). Together, the above three account for 65% of Ethernet packet sizes. Hence, for simulating Ethernet traffic, we use the above three packet sizes according to the probability distribution shown in Table IV. These probability values correspond to actual measured probability values scaled to account for the remaining packet sizes.

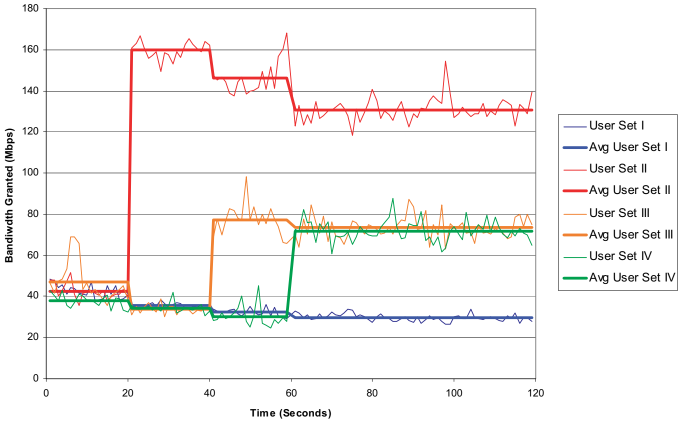
Our intention is to analyze the performance of the algorithm under heavy load, particularly because Internet and video traffic is extremely bursty, and access networks may be frequently subjected to very high loads for short durations of time. The traffic is shown in Table III. All traffic is assumed to be VBR with equal priority. We store arriving packets in a tail-drop queue before they are serviced by the scheduler. The cumulative traffic demand for each SP and for each user for the time interval (60 - 120 seconds), when the network is under

TABLE IV
PACKET-SIZE DISTRIBUTION FOR ETHERNET TRAFFIC.

Packet size	Probability
64 bytes	0.54
594 bytes	0.27
1518 bytes	0.19

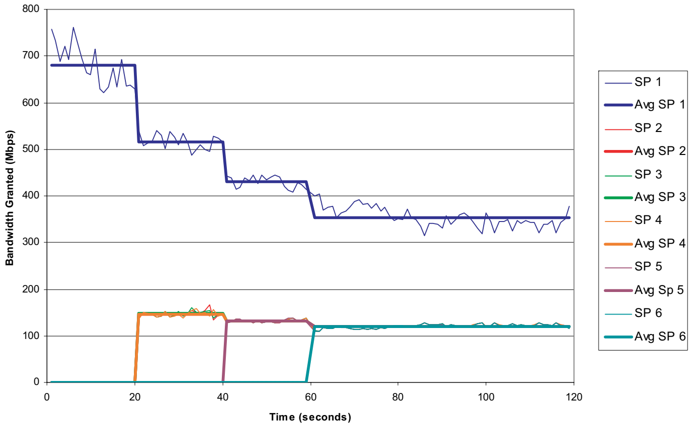


(a) SPs

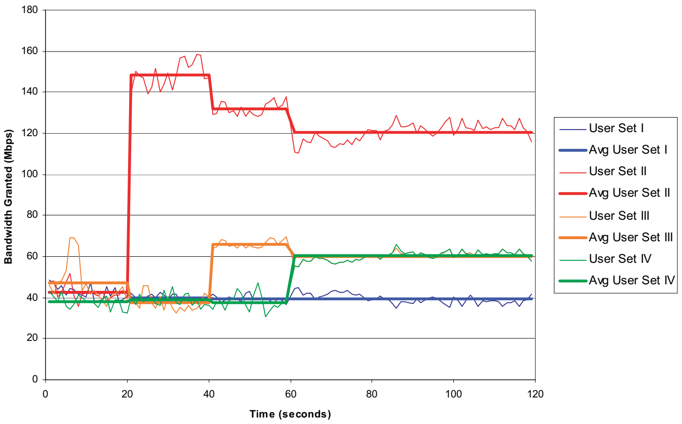


(b) Users

Fig. 10. Throughputs when using the DRR scheduler.



(a) SPs



(b) Users

Fig. 11. Throughputs when using Dual-SLA based scheduler.

TABLE V
CUMULATIVE AVERAGE TRAFFIC FROM SPS TO USERS AT 60-120 S.

SPs	Cumulative average traffic rate	User set	Cumulative average traffic rate
1	640 Mbps	Set I	40 Mbps each
2, 3, 4, 5, and 6	150 Mbps each	Set II	150 Mbps each
		Set III and IV	75 Mbps each

maximum load, is shown in Table V. In the time interval of 60 - 120 seconds, each SP in the set SPs $2 \ldots 6$ has a cumulative demand of an average rate of 150 Mbps, to provide services to their respective users. Hence, we expect that a fair algorithm would assign almost equal average bandwidth to the above SPs. Similarly, User Set I, with a demand of an average rate 40 Mbps, has much less average bandwidth demand compared to the other users. The fairness objective is that User Set I must receive some minimal performance, even under heavy network load.

In Fig. 10[2], we show the results when the Deficit Round Robin (DRR) algorithm is chosen as a scheduler. The DRR algorithm, as explained in Section II, ensures fairness across each independent network flow. The throughput results include Ethernet and IP packet headers. As depicted in Table III, traffic changes at the times of 20s, 40s, and 60s. The thick average line in Fig. 10 indicates the average throughput over the time intervals: 0–20s, 20–40s, 40–60s, and 60–120s. The average value thus jumps discretely at the the times of 20, 40, and 60 seconds. We observe that, at high load $(60s < time < 100s)$, the average throughput of SP 1, which has 16 subscribers, is quite high (around 470 Mbps) at the cost of the other SPs. SPs 5 and 6, which have fewer number of active flows, receive lower average throughput than SPs 2, 3, and 4. User Set II, which has four active flows, receives high bandwidth at the cost of User Set I (with one active flow), which receives approximately 30 Mbps of average throughput.

Our objective in using Dual SLAs is to ensure some minimum degree of service to both SPs and users. Hence,

---

[2] The graphs appear much better in color than in gray-scale print. The graphs in color are also available at the website: http://networks.cs.ucdavis.edu/∼amitabha/OpenAccessJournal/

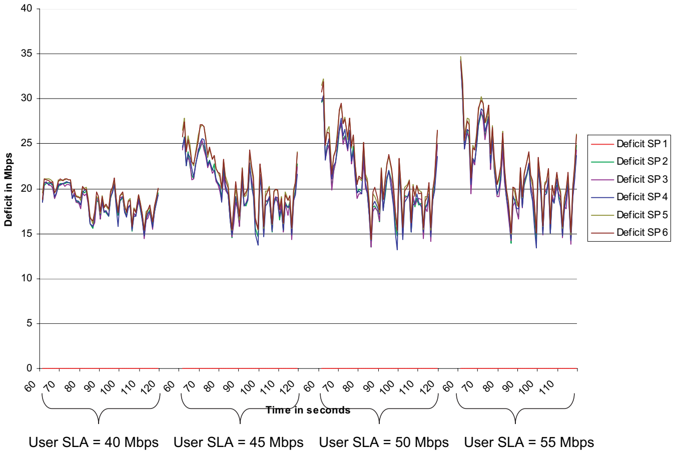


Fig. 12. Analysis of deficits averaged over every second with respect to secondary SLA.

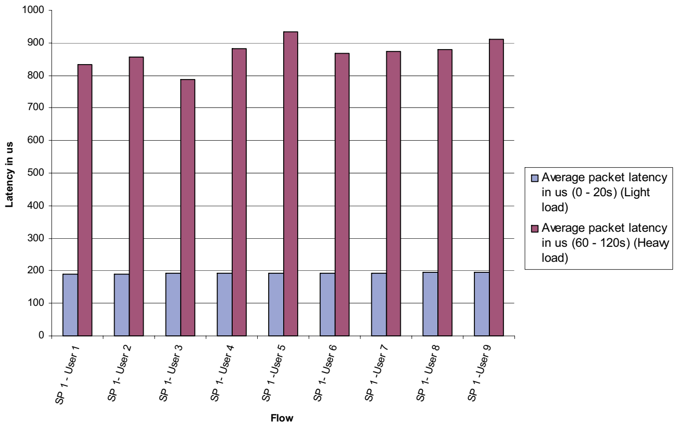


Fig. 13. Average latencies for 9 flows at light and heavy load.

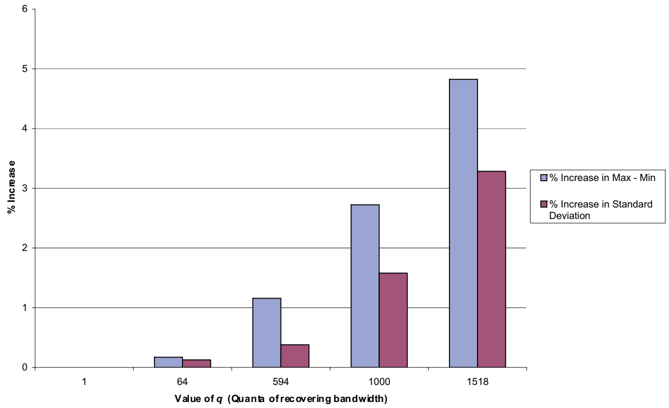


Fig. 14. Analysis of performance vs. fairness of the Dual SLA Scheduling Algorithm

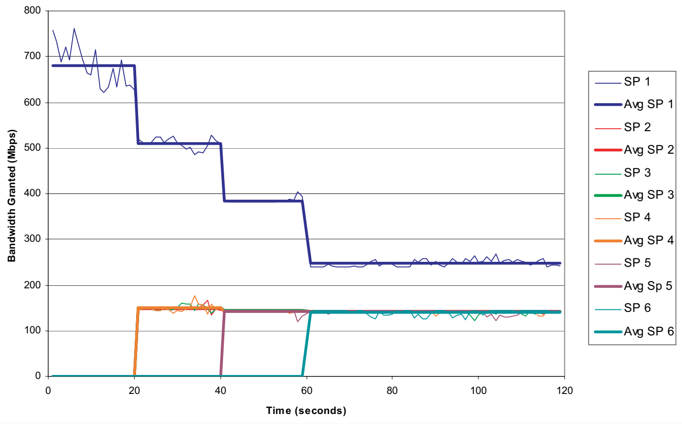
we choose a SP SLA of 150 Mbps so that SPs $2 \ldots 6$ are allocated close to their required average traffic demand. Similarly, we choose a User SLA of 50 Mbps, so that set I of users, which demand average bandwidth of 40 Mbps, get the required bandwidth. Therefore, even under heavy network load, we would expect User Set I to get a reasonable degree of service. The Dual-SLA scheduling algorithm implemented is the version described in Section IV with modifications for implementing Head-of-Line (HOL) packets described in Section V-A and for fairness with respect to the secondary SLA described in Section V-B.

Figures 11(a) and 11(b) show the throughput when the Dual-SLA scheduler is employed. The thick average line indicated the average throughput as before. We observe that the average throughput for SPs $2 \ldots 6$ at high load is much more fair than that in DRR. The average throughput is usually close to 150 Mbps, which is the SP SLA. Similarly, User Set I has an average throughput close to its average traffic demand of 40 Mbps. We observe that a Dual-SLA based scheduler meets the requirements of guaranteeing a minimum degree of service to both SPs and users, and it is fair to both entities.

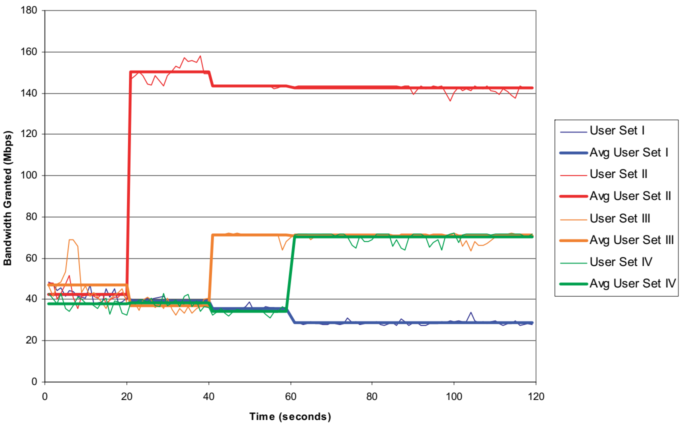
One of the objectives of fairness is to be fair in terms of deficits with respect to the secondary SLA when traffic load is high. In Fig. 12 , we plot the average deficits with respect to the secondary SLA measured across 1-second intervals over the time duration 60 - 120 seconds, as the User SLA is varied from 40 Mbps to 60 Mbps. Note that the deficits are high, because the network is under heavy load during this time interval. Our observation is that the deficits for SPs $2 \ldots 6$ are very close to one another, which is illustrative of the fairness of the algorithm in terms of deficits against the secondary SLA at heavy load. As the User SLA is increased, the deficits are higher, because less bandwidth is available for meeting the secondary SLA after the primary SLA has been met. We also observe that the deficit for SP 1 is uniformly close to 0, at all times, and at all values for User SLA. This is because SP 1 has 9 dedicated users, and satisfying the primary SLA requirements for all these users implies enough bandwidth allocated toward SP 1 at all times.

Figure 13 compares the average packet latency in the access network for 9 flows (corresponding to the 9 users in User Set I which avail service from SP 1). The average packet latency is shown for two cases – one for the time interval (0 – 20 seconds), when the network is under light load (normalized load value of 0.64), and the other for the time interval (60 – 120 seconds), when the network is under heavy load (normalized load value of 1.39). At light load, the average packet latency is expectedly low (in the range of 200 μs) and uniform. At heavy load, the latency is close to uniform across the flows (the observed range is between 790 - 920 μs). Moreover, despite the fact that the network is under heavy load, all 9 flows get good service in terms of packet latency (latency is less than 1.5-ms specification of the ITU G.114 standard). This is because the Dual-SLA scheduler guarantees the User-SLA-specified bandwidth to all users. Even in a congested network, users with traffic demand less than the User SLA achieve good service. Thus, we have demonstrated that the proposed scheduling algorithm is fair in terms of packet latency.

While the DRR algorithm would allocate bandwidth equally to all flows, the Dual-SLA scheduling algorithm tries to meet service requirements of some flows even under heavy load, depending on the SLA specifications. As an example, all 9 flows corresponding to User Set I receive good service even



(a) SPs



(b) Users

Fig. 15. Throughput when SP SLA is the primary SLA and the User SLA is the secondary SLA.

under heavy load. An interesting analysis would be to find out which flows receive poor service under heavy load. In the above example, the flows from SP 1 to User Set II, III, and IV receive poor service under heavy load. This is because the SLA requirements of SP 1 are already met by the allocations to User Set I, and the traffic demand from these users (in Sets II, III and IV) to other SPs (2 . . . 6) is very heavy.

The next set of results illustrates the trade-off between fairness and running time-complexity of ALGORITHM **Recover Bandwidth**, which we discussed in Section IV. To evaluate the fairness with respect to SPs, the range $(Maximum - Minimum)$ and the standard deviation of deficits in meeting the secondary SLA are measured at intervals of 1 second. The average of the above values in the time interval (60 - 120s) is then measured. Figure 14 shows the percentage increase in the above measures, for various values of the *maximum quanta of reallocation* (r) with respect to the measured values for $r = 1$. We consider four different values of r : 64, 594, and 1518 bytes, corresponding to dominant packet sizes of Ethernet traffic, and another value of $r = 1000 \quad bytes$ (1000 being a value between 594 and 1518).

The results presented till now were based on the User SLA as the primary SLA, and the SP SLA as the secondary SLA. Figures 15(a) and 15(b) demonstrate results with the SP SLA as the primary SLA and the User SLA as the secondary SLA for throughput with respect to SPs and users. Comparing with Figures 11(a) and (b), we observe that, at light load (0 - 20s), the throughputs remain the same. However, at heavy load (60–120s), all SPs receive at least the SLA-specified bandwidth of 150 Mbps. But User Set I receives less bandwidth because the secondary SLA cannot be met owing to heavy load. Thus, reversing the priority of SLAs achieves the desired result.

VII. CONCLUSION

In this work, we examined the fairness requirements for open access in a broadband access network such as a PON. We presented and investigated the characteristics of a novel approach for meeting the fairness requirements, by employing Dual Service-Level Agreements. Through extensive simulations of an EPON system, the proposed algorithm was shown to deliver much better fairness in terms of bandwidth allocation and packet latencies in a congested network than traditional fair-queuing algorithms for independent flows which have been previously reported in the literature.

REFERENCES

[1] A. Banerjee and M. Sirbu, "Towards technologically and competitively neutral fiber to the home (FTTH) infrastructure," available at http://100x100network.org/papers/banerjee-tprc2003.pdf

[2] A. Banerjee, G. Kramer, and B. Mukherjee, "Achieving open access in Ethernet PON (EPON)," in *Proc. IEEE/OSA Optical Fiber Communications Conference (OFC) 2005*.

[3] D. P. Bertkesas and R. Gallager, *Data Networks,* Englewood Cliffs, NJ: Prentice-Hall, 1992.

[4] S. Bhatnagar and B. Nath, "ϵ-fairness: a trade-off between overhead and max-min fairness," in *Proc. IEEE International Conference on Communications (ICC) 2003*.

[5] Clause 64,65, IEEE 802.3ah standard, approved 24 June 2004.

[6] G. Kramer, *Ethernet Passive Optical Networks*, McGraw-Hill, 2005.

[7] G. Kramer and G. Pesavento, "Ethernet Passive Optical Network (EPON): building a next generation optical network," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 69-74, Feb. 2002.

[8] 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.

[9] G. Kramer, B. Mukherjee, S. Dixit, Y. Ye, and R. Hirth, "Supporting differentiated classes of service in Ethernet passive optical networks," *OSA J. Optical Networking*, vol. 1, no. 8/9, Aug./Sept. 2002.

[10] International Telecommunication Union Recommendation G.114, available at http://www.itu.int/itudoc/itu-t/aap/sg12aap/history/g.114/g114.html

[11] Palo Alto FTTH deployment project at http://www.pa-fiber.net/

[12] M. Ma, Y. Zhu, and T. H. Cheng, "A bandwidth guaranteed polling MAC protocol for Ethernet passive optical networks," in *Proc. IEEE INFOCOM 2003*.

[13] M. McGarry, M. Meier, and M. Reisslein, "Ethernet PONs: a survey of dynamic bandwidth allocation (DBA) algorithms," *IEEE Optical Communications*, vol. 42, no. 8, pp. S8-S15, Aug. 2004.

[14] NSF Workshop Report, "Residential broadband revisited: research challenges in residential networks, broadband access, and applications," Oct. 2003, available at http://cairo.cs.uiuc.edu/nsfbroadband/

[15] A. K. Parekh and R. G. Gallagher, "A generalized processor sharing approach to flow control in integrated services networks: the single node case," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 344-357, June 1993.

[16] D. Sala and A. Gummalla, "PON functional requirements: services and performance," presented at the IEEE 802.3ah meeting in Portland, Oregon, July 2001, available at http://www.ieee802.org/3/efm/public/jul01/presentations/sala_1_0701.pdf

[17] M. Shreedhar and G. Varghese, "Efficient fair queuing using round robin," in *Proc. ACM SIGCOMM '95*.

[18] I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queueing: a scalable architecture to approximate fair bandwidth allocations in high-speed networks," *IEEE/ACM Trans. Networking*, vol. 11, no. 1, pp. 33-46, Feb. 2003.

[19] M. S. Taqqu, W. Willinger, and R. Sherman, "Proof of a fundamental result in self-similar traffic modeling," *ACM/SIGCOMM Computer Commun. Review*, vol. 27, pp. 5-23, Apr. 1997.

[20] L. Zhang *et al.*, "Dual DEB-GPS schedular for delay-constraint applications in Ethernet passive optical networks," *IEICE Trans. Commun.*, vol. E86-B, no. 5, May 2003.

Amitabha Banerjee (S'02) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Delhi, India, in 2000, and the M.S. degree in computer science from the University of California, Davis, in March 2004. He is currently working toward the Ph.D. degree in the Computer Science Department, University of California, where he is a Research Assistant in the Networks Research Laboratory.

From June 2000 to July 2002, he was with Tavant Technologies, Santa Clara, CA. He was with Los Alamos National Labs between August–October 2004, and July–September 2005. Amitabha's research interests include broadband optical access networks, and end-to-end data transfer over Lambda Grid networks.

Glen Kramer (S'00–M'03) is the Chief Scientist for Teknovus, Inc. He received his Ph.D. in Computer Science in UC Davis, where he remains a research associate in the Networks Research Lab.

Glen is a member of the IEEE Standards Association and past editor of the EPON Protocol Clause in the "Ethernet in the First Mile" standard. Author of *Ethernet Passive Optical Networks* (McGraw Hill 2005), he has done extensive research in areas of traffic management, Quality of Service and fairness in access networks. Glen is the founder of the EPON Forum and teaches EPON tutorials and workshops at conferences around the world.

Biswanath Mukherjee (S'82–M'87) received the B.Tech. (Hons) degree from Indian Institute of Technology, Kharagpur (India) in 1980 and the Ph.D. degree from University of Washington, Seattle, in June 1987. At Washington, he held a GTE Teaching Fellowship and a General Electric Foundation Fellowship.

In July 1987, he joined the University of California, Davis, where he has been Professor of Computer Science since July 1995, and served as Chairman of Computer Science during September 1997 to June 2000. He is a winner of the 2004 Distinguished Graduate Mentoring Award at UC Davis. Two PhD Dissertations (by Dr. Laxman Sahasrabuddhe and Dr. Keyao Zhu), which were supervised by Professor Mukherjee, were winners of the 2000 and 2004 UC Davis College of Engineering Distinguished Dissertation Awards. To date, he has graduated nearly 25 PhD students, with almost the same number of MS students. Currently, he supervises the research of nearly 20 scholars, mainly PhD students and including visiting research scientists in his laboratory.

Mukherjee is co-winner of paper awards presented at the 1991 and the 1994 National Computer Security Conferences. He serves or has served on the editorial boards of the *IEEE/ACM Transactions on Networking*, *IEEE Network*, *ACM/Baltzer Wireless Information Networks* (WINET), *Journal of High-Speed Networks*, *Photonic Network Communications*, *Optical Network Magazine*, and *Optical Switching and Networking*. He also served as Editor-at-Large for optical networking and communications for the IEEE Communications Society. He served as the Technical Program Chair of the IEEE INFOCOM '96 conference. He also serves as Chairman of the IEEE Communication Society's Optical Networking Technical Committee (ONTC).

Mukherjee is author of the textbook *Optical WDM Networks* to be published by Springer in 2005. Earlier, he authored the textbook *Optical Communication Networks* published by McGraw-Hill in 1997, a book which received the Association of American Publishers, Inc.'s 1997 Honorable Mention in Computer Science. He is a Member of the Board of Directors of IPLocks, Inc., a Silicon Valley startup company. He has consulted for and served on the Technical Advisory Board (TAB) of a number of startup companies in optical networking. His current TAB appointments include: Teknovus, Intelligent Fiber Optic Systems, and LookAhead Decisions Inc. (LDI). Mukherjee's research interests include lightwave networks, network security, and wireless networks.