Application-Aware Software-Defined EPON Access Network*

Divya Chitimalla[1], Saigopal Thota[1], S. Sedef Savas[1], Pulak Chowdhury[1], Massimo Tornatore[1,2], Sang-Soo Lee[3], Han-Hyub Lee[3], Soomyung Park[3], HwanSeok Chung[3], and Biswanath Mukherjee[1]
[1]University of California Davis, USA; [2]Politecnico di Milano, Italy;
[3]Electronics and Telecommunications Research Institute, Korea
{dchitimalla, sthota, ssavas, pchowdhury, mtornatore, bmukherjee}@ucdavis.edu,
{soolee, hanhyub, smpahk, chung}@etri.re.kr

Abstract—As bandwidth requirements of users in passive optical networks (PONs) continue to increase rapidly, especially due to the growth of video-streaming traffic, the need for resolving bandwidth contention among competing users and applications becomes more compelling. To address this problem, we propose Application-Aware Software-Defined Ethernet Passive Optical Network (EPON) architecture. It utilizes application-level feedback from the client side (for video users) to the network, through a Software-Defined-Network (SDN) controller, to achieve better client/service-level differentiation in the downstream direction of an EPON access network. We adopt adaptive video-streaming which utilizes feedbacks regarding network congestion for improving Quality of Experience (QoE) of video-streaming clients. Numerical results for video-streaming applications demonstrate that the proposed architecture for downstream resource allocation considerably reduces video stalls, increases video buffer levels, and also reduces video-switching rate leading to better QoE for users as a result of better client and service-level differentiation.

Keywords—EPON; SDN; Application-awareness; Downstream traffic.

I. INTRODUCTION

Ethernet Passive Optical Network (EPON) is a dominant access network solution, thanks to its low capital (CAPEX) and operational (OPEX) expenditures. The traditional EPON architecture consists of an Optical Line Terminal (OLT) connected to the Optical Network Units (ONUs) through a feeder fiber, a splitter, and distribution fibers. The downstream traffic going from the OLT to the ONUs is broadcast, while the upstream traffic is multiplexed in the feeder fiber using a Medium-Access-Control (MAC) protocol, typically a dynamic one, such as the interleaved polling mechanism in IPACT [1]. It is known that optical fiber can support very high bit rates. Hence, each ONU can serve several clients, i.e., fiber-to-the-x (FTTx) clients, where x can be H (home), B (business), etc. Such clients can be connected to an ONU by fiber, copper (e.g., using the EPON over Copper (EPoC) protocol [2]), or wireless links [3].

In such scenarios where EPON is supporting many clients, traditional OLT-based downstream allocation will not be sufficient to achieve superior QoE of clients which may be using bandwidth-hungry applications such as video-streaming. Bandwidth contentions among the clients can lead to a resource crunch in the access network, leading to poor QoE of clients. Therefore, intelligent resource allocation and scheduling for downstream traffic is needed, one that can cater to the varying needs of different clients and their applications.

Bandwidth demand from video-streaming applications such as Skype, Netflix, etc. continue to grow at a rapid rate, utilizing considerable downstream bandwidth, which is becoming a bottleneck due to the omnipresence of high-bandwidth-consuming clients. The QoE of video-streaming is increasingly becoming a concern for network providers. To address these issues, the network needs to evolve from being application-unaware to application-aware to help improve resource allocation to the clients contending for bandwidth.

Application-aware networking keeps track of the application state, and optimizes its performance using this information [5]. The functionalities offered by Software-Defined Networking (SDN), a new networking paradigm based on a logically-centralized control plane that simplifies network management and supports network programmability, is

*This work was supported by ICT R&D program of MSIP//IITP. [14-000-05-001, Smart Networking Core Technology Development]

expected to enable application-awareness to become a reality. In SDN, the control plane is separated from network hardware, which enables dynamic control and the ability to allocate resources at any given moment to best effect [4]. In this study, we propose a novel software-defined architecture where the controller interfaces are connected with the client-side video-streaming applications and with the OLT, such that the controller can perform downstream bandwidth allocation for the ONUs based on the application state parameters of the clients. We also investigate a model where adaptive streaming of video clients takes information from the SDN controller to adapt the resolution of video-streaming. Our results show that the QoE of video clients is considerably improved, when an application-aware architecture is used, in terms of reduced video stall time, improved video quality, etc. compared to traditional EPON architecture.

The rest of this study is organized as follows. Section II gives an overview of the proposed application-aware SDN architecture along with an application-aware resource-allocation scheme and adaptive streaming methodologies for video-streaming applications. Section III gives the simulation model of our proposed scheme, brief descriptions of traditional application-unaware downstream schemes that we compare to our proposed scheme, adaptive streaming traffic model considered in simulations, evaluation setup, and the results. Section IV concludes the study.

II. APPLICATION-AWARE OLT DOWNSTREAM RESOURCE ALLOCATION

A. Overview of Architecture

The traditional EPON architecture is shown in Fig. 1(a). It consists of an OLT connected to multiple ONUs by optical fiber with a splitter. Typically, ONUs are at home or office locations. Some studies proposed coexisting optical and copper architecture using Ethernet over Copper (e.g., EPoC [2]). The application-aware SDN-enabled EPON architecture that we are proposing is shown in Fig. 1(b), where the traditional OLT-ONU framework is unaltered, but a feedback mechanism from applications at clients to the SDN controller is added. The links (colored lines) from the ONU to clients can be optical fiber, copper, or wireless links (for cellular transmissions). The links (dotted lines) between the clients, the SDN controller, and the OLT are logical links. The links from the clients to the SDN controller are for sending application feedback, and the links from controller to OLT are for controlling the OLT's downstream scheduling. Routing of control signals from the clients to the controller can be done by using upstream bandwidth.

EPON is also being investigated as a possible solution for mobile traffic front-hauling for Cloud Radio Access Network (C-RAN) [9] [10] architecture where a traditional base station is decoupled into a radio unit (RU, with antennas), and a Digital processing Unit (DU) with a goal of increasing coverage using small cells [7] [8] and reducing network cost. EPON used for front-hauling needs to support several mobile (RUs) and fixed (ONUs) clients. We refer to EPON architecture that supports a radio access network (RAN) in C-RAN scenario along with fixed (home or office) users as EPON wired-wireless converged architecture (see Fig. 2). However, this work focuses only on fixed users (highlighted oval in Fig. 2). Figure 2 also shows separated data plane, control plane, and service plane in the wired-wireless converged access network architecture. The service plane is responsible for achieving differentiated services for the clients. The control plane is responsible for sending and receiving the control packet information from the applications running on the client to the SDN controller. The data plane is where the actual physical transmission of the control and data packets occurs. As shown in the figure, an SDN controller is capable of having an interface with the clients (SDN management applications) through North-Bound API (NB-API). SDN management application will collect all the feedback from client applications, and feed them back to the SDN controller through NB-API. The controller can interface with the OLT by a South-Bound API (SB-API).

Traditional OLTs schedule downstream traffic for the ONUs by maintaining separate queues for each ONU [11] [12]. To provide service-level differentiation, Ref. [13] proposes to differentiate data services provided to a user through different classes of service (CoS), in such a way that each incoming downstream packet at the OLT is placed in the queue assigned to that particular CoS queue of the ONU. The number of queues at the OLT is limited to a maximum of seven for each ONU [13], but typical industry implementations support three queues per ONU as a higher number of queues leads to scalability issues. To further improve the service-level differentiation, a naïve approach would be to assign a queue for each service (application) at the OLT for each user. But this is infeasible considering the plethora of applications, and that future EPON might support many more clients than today (e.g., C-RAN mobile clients).

A network that is aware of applications (or services) that are utilizing the bandwidth is a promising solution to achieve better service differentiation. The functionalities offered by SDN [19], such as the ability to program the network and to interface with client applications, enable application-awareness to become a reality. Therefore, for achieving fine-grained client-and-service-level differentiation, we have an SDN controller interface with the applications to control the downstream resource allocation at the OLT. To illustrate the potential of this architecture, we consider video-streaming applications, and perform EPON downstream resource allocation based on the buffer levels of the videos that are streaming and playing at the client (e.g., YouTube, Netflix, etc.). Video-streaming applications are chosen for this study because they consume high downstream bandwidth and have flexibility in bandwidth requirements (due to adaptive video-streaming). Video buffer levels of streaming applications are reported to the controller through NB-API, and when the controller notices that certain clients served by a particular ONU are suffering from low video buffer levels (potential for experiencing video stalls and hence poor QoE), the controller triggers an action to restore a satisfactory QoE for the users.

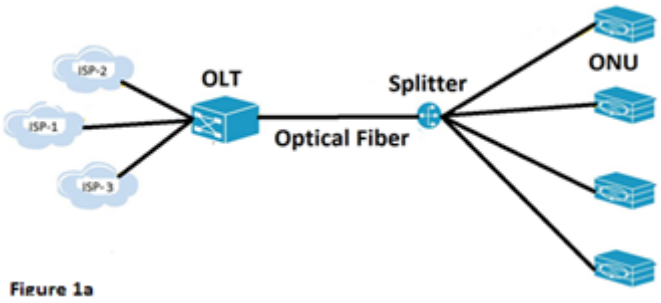

Figure 1: (a) Traditional EPON architecture.

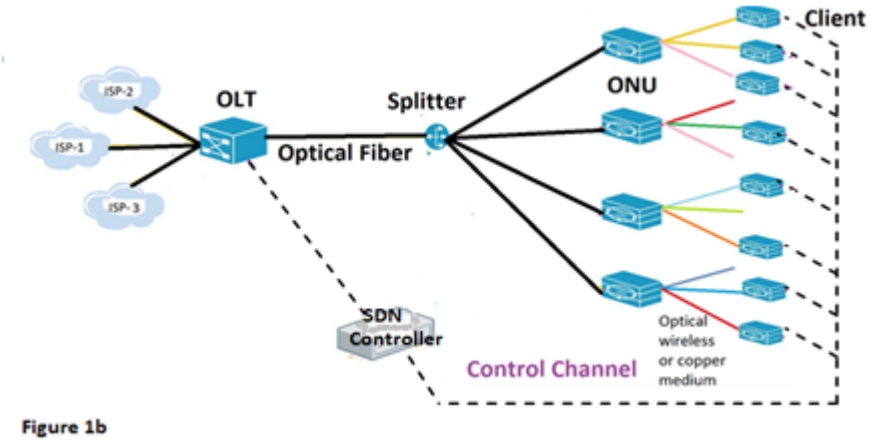

Figure 1: (b) Application-aware SDN-enabled EPON architecture.

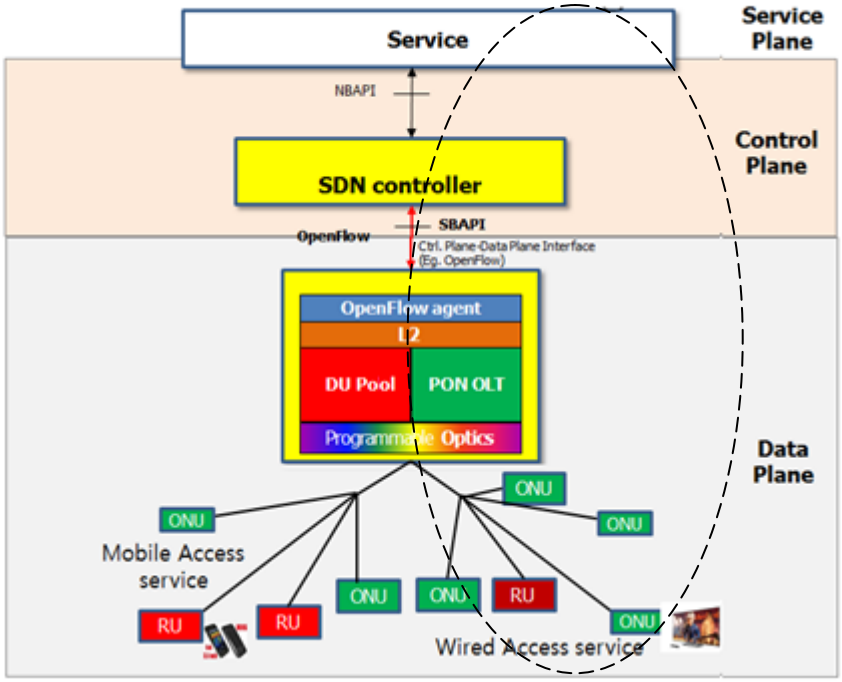

Figure 2: Wired-wireless converged access network.

It is assumed that the OLT uses a weighted round robin (WRR) scheduling policy [13] (see Section III-A, where weight of an ONU reflects amount of downstream bandwidth given to it) to schedule the next packet for downstream transmission. Since the controller has the overall picture of the applications running on the clients, it can calculate new bandwidth assignment for each ONU acknowledging this information. The algorithm used for calculating new bandwidth assignment (also referred to as "weights") is described in Section II-C. These weights are sent out in a control packet (MAC layer or SB-API). These weights are sent out in a control packet (MAC layer or SB-API) through the control interface between controller and OLT. The OLT extracts the new weights, and uses them for its weighted-round-robin scheduling of the downstream traffic.

An ONU, which has clients experiencing poor QoE, will be benefitted quickly after the OLT gets the control packet with new weights. Hence, suffering clients will have their video buffer levels increased because of additional downstream bandwidth allocated to them in the form of new weights, improving the QoE for users.

B. Client and Service-Level Differentiation

As access bandwidth is becoming a scarce resource, there is increasing contention for bandwidth at the client level and application (service) level. To provide better services to clients, the network provider can consider the different requirements and state of each client, and differentiate the services among them.

For instance, if a client has subscribed for the highest-quality video-streaming service to watch a World Cup match, he/she must be assured of the best possible QoE (high video quality with a minimum number of video stalls), and be given differentiated treatment from other clients who can tolerate lower QoE for video-streaming (e.g., users watching the match for free).

To illustrate this client-level differentiation, we have classified the clients into five categories based on the level of QoE which they are expecting from the network. A user of Category 5 is the highest-priority user, who accepts only very high video quality (say High Definition (HD)

and Standard Definition (SD)). Without loss of generality, let us consider that there are six video qualities which are available from the video-streaming server (say YouTube). The user types and acceptable resolutions are summarized in Table 1. Typical resolution types and the specifications in terms of pixels are summarized in Table 2.

A client of Type 1 accepts any resolution from 1 to 6, and hence that client pays less for the service compared to a client of Type 2 or higher since the latters' minimum acceptable resolutions are higher. Hence, different revenues can be collected from clients of different types to achieve client-level differentiation. Note that, although this example deals specifically with video-streaming, the same concept can be extended to other applications, which do not have stringent bandwidth requirements unlike constant bitrate (CBR) (e.g., voice call) applications.

Client Type	Acceptable Resolutions
5	5,6
4	4,5,6
3	3,4,5,6
2	2,3,4,5,6
1	1,2,3,4,5,6

Table I: Client types and acceptable resolutions.

Resolution	Specification	Video Rate(Mbps)
6	2160p: 3840x2160	45
5	1440p: 2560x1440	10
4	1080p: 1920x1080	8
3	720p: 1280x720	5
2	480p: 854x480	2
1	360p: 640x360	1

Table 2: Resolution specifications and video rates.

The metrics we use in this study to evaluate QoE are as follows.

First, for a video user, stall is considered as the major cause for the reduction in QoE [14] [15]. Hence, video stall time is an important metric to compare the QoE of users for different resource-allocation schemes [15] [16] [17] [18]. Although EPON is a high-bandwidth access network, our study concentrates on the scenarios of resource crunch, where the access network bandwidth becomes a bottleneck (scarce resource) due to bursty traffic over some time durations, leading to significant interruptions in the video-streaming services.

Second, buffer level (or to-be-played video cached at the client application) is an important parameter that can reflect the QoE [15] of the video users. Although there is no direct correlation of buffer level with QoE, a good amount of buffer level to accommodate data for video playback is an essential parameter that reflects QoE of video users. To quantify the performance of clients belonging to same ONU, we consider the average of buffer levels at the clients, considering that larger buffer levels provide higher QoE.

Third, resolution of video is another parameter that determines QoE of clients. High resolution improves QoE of end users.

Fourth, video-switching rate is defined as the rate at which a video is switched from one resolution to another as a result of adaptive streaming. A high video-switching rate [20] is not desirable; however, there is no evidence that it has direct correlation with QoE of video clients.

Therefore, in our study, video stall time, average buffer level, resolution, and video-switching rate are considered to compare the effectiveness of our proposed mechanisms with respect to

traditional application-unaware schemes such as Round Robin, Static Weighted Round Robin (i.e., with static weights), etc. in Section III-D.

C. Application-Aware SDN-Enabled Resource Allocation (AASRA) Scheme

In this section, we present the details of our Application-Aware SDN-Enabled Resource Allocation (AASRA) scheme for downstream scheduling in EPON.

Before that, we introduce a baseline weighted-round-robin (WRR) [13] scheduling mechanism that we use as a part of the proposed scheme (we will also use this baseline scheme for comparison in Section III-A). Let us assume that an incoming downstream packet at the OLT is queued in the corresponding CoS queue for the destination ONU. In WRR, the OLT schedules the packet queues based on the weights assigned to the respective queues and the sizes of the queues, thereby taking into account: (a) the importance of a queue (represented by the weights), and (b) the amount of data awaiting at the OLT to be transmitted to the clients (represented by the queue length). The OLT multiplies the weights with the corresponding queue lengths and schedules the packet that is at the top of the queue for which the product is maximum. WRR is one possible method for weighted scheduling.

In AASRA scheme, we assume that the end-user's web browser can have a software extension (plug-in application) that can collect the statistics of video buffer level from a streaming application, and report this information to the SDN controller. Figure 5 shows the control flow for a video client implementing AASRA scheme using buffer-based adaptive streaming. Whenever the buffer level is below a pre-defined threshold (*min_threshold*), the application notifies the controller that it is suffering from low QoE (see Fig. 5). This information is sent using a control packet on the control channel shown in Fig. 1(b). When the control packet is processed at the SDN controller, a table which is maintained at the controller is updated with new information. Figure 6(a) shows the high-level control flow at the SDN controller implementing AASRA scheme with buffer-based adaptive streaming. The table at the controller has information on all the users which are suffering from low QoE (low video buffer levels in our case).

The SDN controller triggers the action of changing the weights at the OLT for downstream scheduling whenever it detects that a client is suffering from low buffer level (i.e., if the buffer level is less than the pre-defined *min_threshold,* see Fig. 6(a)). If the controller has decided to change the weights at the OLT, it calculates the new weights based on the information of users that are suffering from low QoE. Each ONU is assigned minimum weights (and hence minimum bandwidth) based on the number of clients it supports and the client types (summarized in Table 1). The minimum weights can be determined by the revenue model of the operator. The higher the minimum weight is for an ONU, the higher the bandwidth it would get in the downstream scheduling no matter how many users under the ONU are suffering. An ONU which has reported many cases of low QoE will get higher weights in the next schedule of the ONU queue and hence higher amount of bandwidth resource. The weights thus calculated by the controller are sent to the OLT.

Initial weights are assigned to each ONU by the controller, proportional to the number of clients supported by the ONU and client types, and are represented by *curr_weights* (note that the total sum of *curr_weights* is set to the total available downstream bandwidth). *min_weights* represents the minimum bandwidth an ONU gets, irrespective of QoE of the clients supported. As discussed before, *min_weights* of an ONU depends on number of clients the ONU supports and the priority of clients (client types). Excess bandwidth is the difference between total bandwidth and the total sum of minimum bandwidth that must be given to each ONU. This excess bandwidth is redistributed in proportion to the total number of clients which are suffering from low video buffer levels.

a. Number of users starving at ONU 'i' is "T_i"
b. Total starving users for all ONUs is "T"
c. Total number of ONUs is "N"

min_weights (i) → *function (# users, user_priority)*

Excess → $\sum_{i=1}^{N} curr\_weights(i) - \sum_{i=1}^{N} min\_weights(i)$

new_weights (i) → *min_weights (i) + (T_i/T)*Excess*

min_weights(i) represents the minimum weight assigned to the ONU 'i', *new_weights(i)* is the new weight assigned to ONU 'i' which has "S" users suffering from low QoE (low buffer level).

D. SDN-Based Adaptive Streaming (SBAS)

Adaptive streaming (or rate-adaptive video-streaming) is the method of sending video at a rate that matches the real-time end-to-end link capacity. The goal is to deliver uninterrupted video playback experience at a quality that is sustainable by the network conditions. This task requires generating video-streams at multiple bitrates. Currently, there are two ways of obtaining such streams. First option is to use legacy H.264/AVC and independently encode original video at multiple bitrates (see Fig. 3). This method is adopted by most of today's major video content providers such as YouTube, Netflix, etc. The second option is to use SVC extension of H.264 standard [21], which decreases the overall size of the video-streams by encoding only the difference signal for the higher layer(s). The player switches between streams encoded at different bitrates [6] depending on available bandwidth resources. Adaptive streaming is done by video content providers based on client feedback to reduce video stalls in cases of network resource scarcity. We use the first method (multiple bitrate encoding) in our numerical evaluation.

Existing adaptive-streaming algorithms use capacity (bandwidth) estimation methods to design adaptive streaming. However, Ref. [20] shows that capacity can vary widely over time in commercial services, and hence it cannot work as an accurate metric to base the rate adaptation. Also, Ref. [20] shows that an alternative approach of using video buffer levels directly rather than capacity estimation works better in terms of reducing video stall time. However, this work does not include application-awareness in the network as discussed in our study.

Since commercial adaptive-streaming algorithms are proprietary, we designed our adaptive-streaming scheme for simulation modeling using two thresholds for the buffer level of video at the client: *buff_min* and *buff_max*. If the buffer level falls below *buff_min*, the traffic generator generates packets of lower resolution than previously what it was serving (see Fig. 5). If the buffer level goes above *buff_max*, the resolution of next packets goes higher than the previous ones.

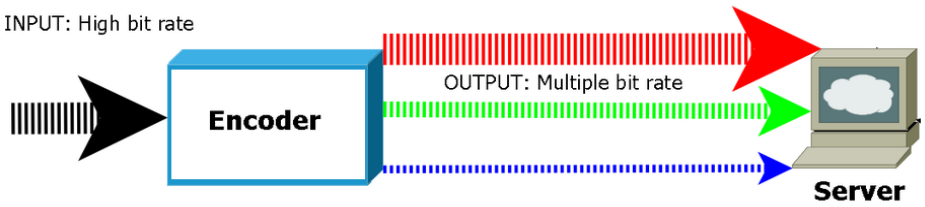

Figure 3: Adaptive bitrate encoder.

The adaptive streaming can be seen as a control loop at the application layer (e.g., http) that is trying to change the video quality to ensure smooth playback by combating unpredictable variations in available bandwidth. The client application feeds back its status using an http

message to the streaming server. This control loop exists in a traditional application-unaware architecture also, and is shown in Fig. 4 (adaptive-streaming control loop). However, for an application-unaware network, there is no control loop on the network side to optimize the video experience. Figure 4 shows the application-aware network control loop for the proposed application-aware EPON, where application's status on the client side is fed back to the controller and OLT to optimize video experience. We find that the additional network control loop is required to achieve superior QoE for video clients. It tries to reduce video-quality degradation as much as possible by providing additional bandwidth to clients suffering from poor QoE when the adaptive streaming loop can no longer achieve this by reducing video bitrate to combat video stalls. It is important to note that the two controls (adaptive streaming and application-aware resource allocation) work independently.

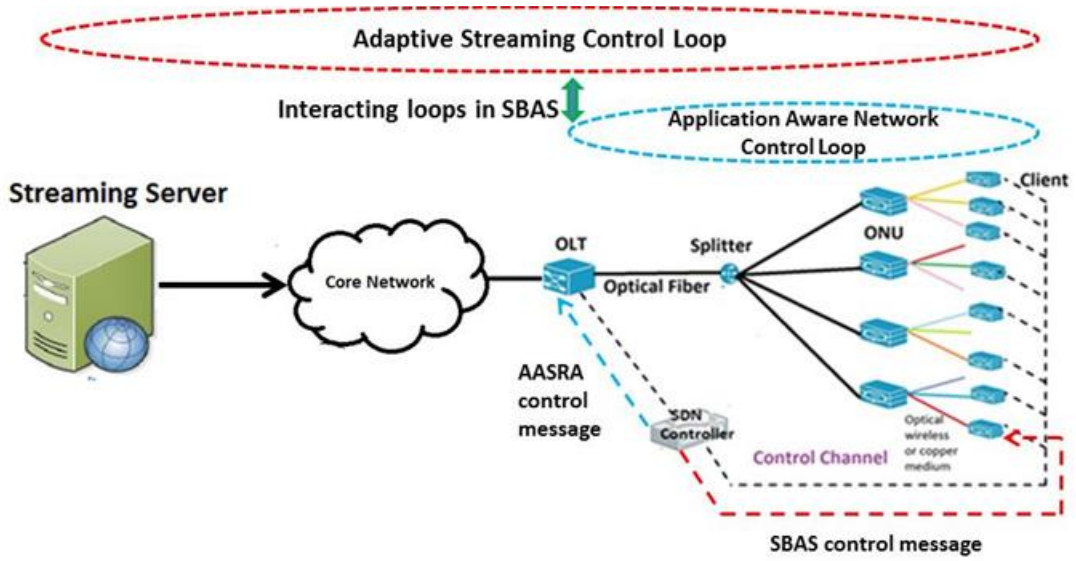

Figure 4: Control loops in application-aware network.

The algorithm discussed till this point is not affecting the end-to-end adaptive streaming but it is using the application information to optimize video experience by reducing video stall time. Figure 4 shows the AASRA control message, from SDN controller to OLT, which decides bandwidth allocation to ONUs based on clients' applications. In AASRA, the SDN controller does not affect adaptive streaming. Since the adaptive-streaming algorithm between a server and a client tries to optimize that particular client's performance, its myopic view (compared to SDN controller's global view of all clients) might end up in an approach that is not globally optimal. Hence, we propose the SDN-based adaptive streaming (SBAS) approach where the SDN controller suggests to the client to get to the lowest possible resolution when it senses that network is congested (using a threshold on number of clients suffering from video stalls). In the second approach, we use the AASRA scheme along with SBAS to achieve the best QoE for video clients. Figure 6(b) shows control flow at the SDN controller implementing AASRA with SBAS scheme. When the SDN controller senses that the number of clients suffering from poor QoE is large (greater than a pre-defined threshold, i.e., *sbas_threshold*), it can no longer efficiently redistribute bandwidths using the AASRA scheme to restore QoE of video clients. This is due to the fact that some clients in the system might be getting video resolutions higher than their lowest acceptable and they will not be affected by AASRA scheme immediately. Hence, in SBAS, the controller triggers a control message to all such clients to get to their lowest acceptable resolution (SBAS control message in Fig. 4) whenever it cannot accommodate bandwidth for all clients with poor QoE (see Fig. 6(b)). Hence, we note that SBAS along with AASRA makes the two control loops, i.e., adaptive streaming control loop and application-aware network control loop, interactive. Figure 4 shows the

control loops interacting (adaptive streaming and application-aware network) for the proposed SDN-based adaptive streaming scheme where the two interact unlike traditional adaptive streaming.

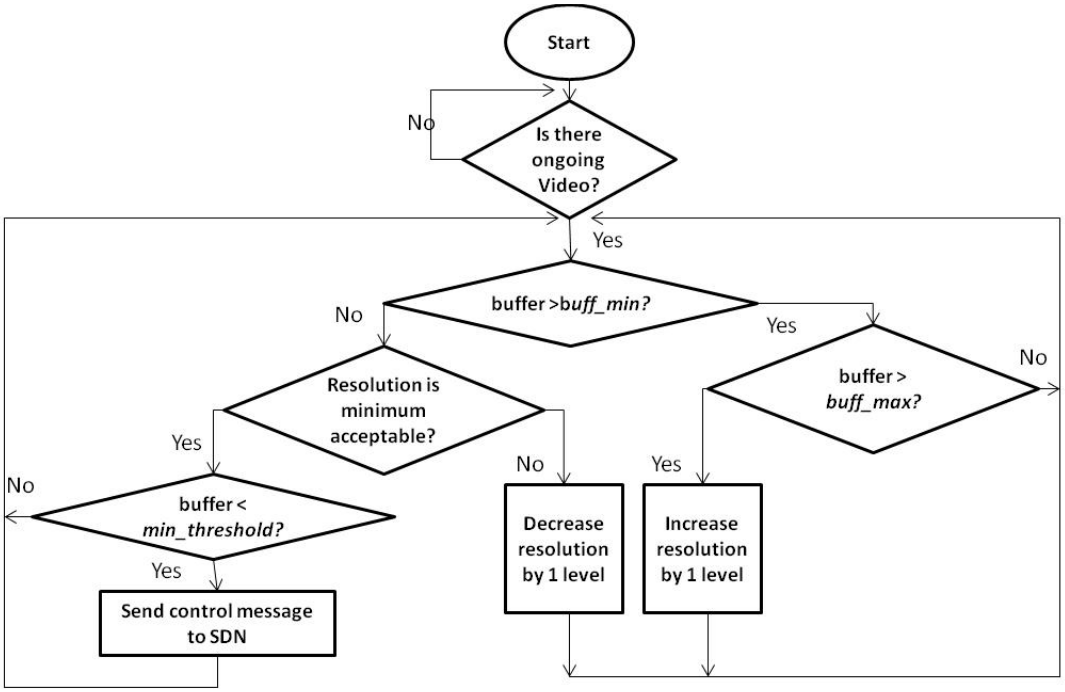

Figure 5: AASRA flow chart at the client.

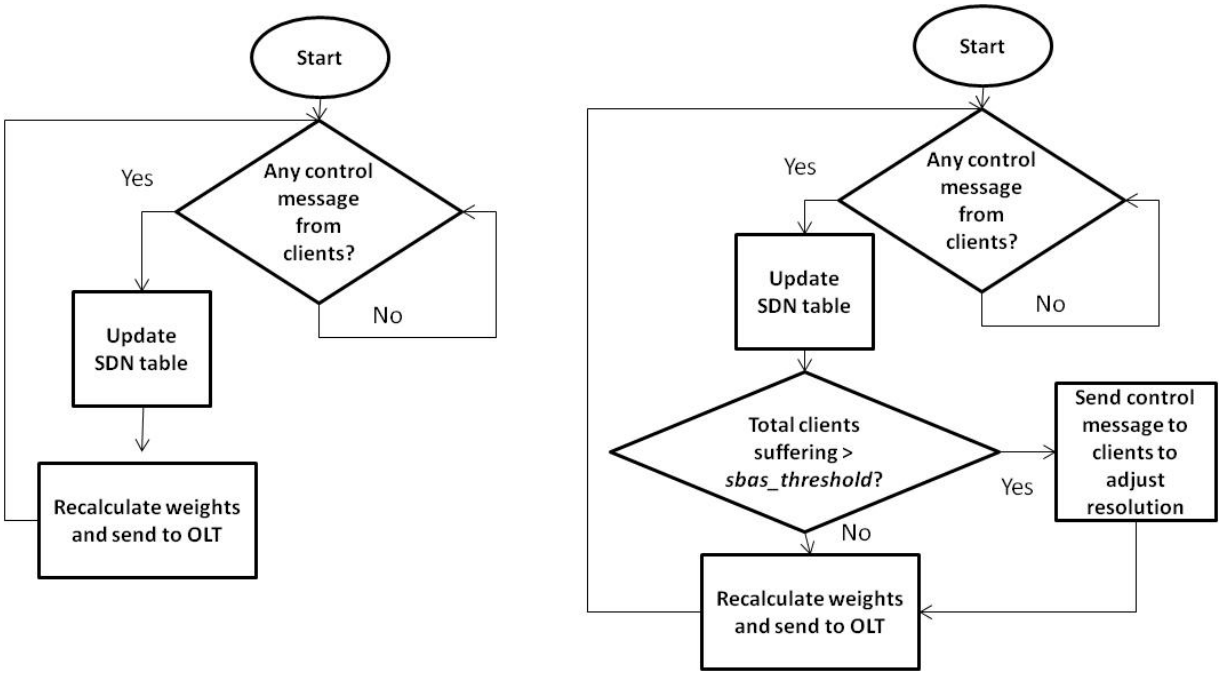

Figure 6(a): AASRA flow chart at the SDN controller; (b) AASRA with SBAS flow chart at the controller

III. Illustrative Numerical Examples

We performed packet-level discrete-event-simulation experiments on the access network architecture to evaluate the proposed Application-Aware SDN-Enabled Resource Allocation (AASRA) and AASRA with SBAS schemes. The high-level overview of our simulator for AASRA scheme is summarized as a flow diagram in Fig. 7.

Packet Source is the module in the simulation model (shown in Fig. 7) that is responsible for generating video-streaming traffic (using adaptive streaming and SBAS as discussed in Section II-D). The generated packets go into the respective downstream queues at the OLT, which uses an event scheduler to send out the packets at their scheduled time of arrival. The packets destined for each ONU are en-queued in the respective CoS queue. DS OLT Scheduler is the module which performs the proposed downstream allocation for the incoming downstream packets at the OLT by triggering the packet-depart event at the event scheduler.

If a packet arriving at the client is a video packet, then the module Video Buffer Calculator calculates the buffer level by taking the information of the packets received. If it is detected that the video buffer level is below *min_threshold*, this module triggers an event, called *Notify_SDN*. This message is processed by the event scheduler (see Fig. 7) and passed on to the SDN controller module which utilizes this information to re-assign weights at the OLT.

The re-assigned weights are received at the DS OLT scheduler when the event named *New_alloc_Rx* (Fig. 7) is scheduled at the event scheduler.

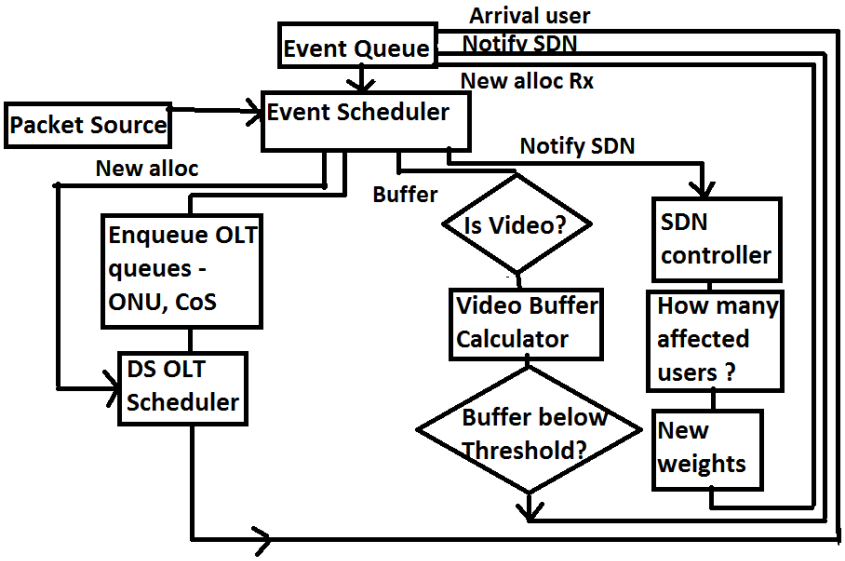

Figure 7: Flow diagram for simulation of AASRA scheme.

A. Comparison of Schemes

The proposed schemes AASRA and AASRA with SDN-based adaptive streaming (AASRA with SBAS) are compared with two application-unaware schemes, which follow the traditional EPON downstream scheduling without an SDN controller interfacing with the client-side applications. They are:

1. Round Robin (RR): This is a well-known scheme, where each ONU queue is scheduled one after the other in a circular manner. There is no differentiation among the different ONUs.
2. Static Weighted Round Robin (WRR): This is an improvement over the basic Round Robin scheme because, in WRR, different ONUs are assigned different weights based on the number of clients they support and the client types they belong to. This is same as the minimum weights calculation of our proposed scheme. These weights are static and do not adapt to the feedback from the client-side applications.

B. Traffic Modeling

Buffer-based adaptive streaming algorithm (using multi-rate video encodings) as discussed in Section II-D is used to generate video traffic for AASRA scheme. SBAS is used to generate video traffic for *AASRA with SBAS* scheme. The resolutions are kept within the bounds of

acceptable resolutions as shown in Table 1. Videos are requested in chunks of size 4 seconds [20], and adaptation of videos happens in chunk granularity (resolution can change on a chunk-by-chunk basis). Chunk size is determined as 4 seconds in Ref. [20] using control-theory mechanisms.

An increasing load on the downstream bandwidth which results in poor performance at the client's video-streaming application is first combated by adaptive streaming, that will try to reduce the resolution of the next packets destined to that user. But when the resolutions cannot be further reduced, the buffer level will get reduced. When the buffer level falls below a pre-defined *min_threshold*, it is reported to the SDN controller (see Fig. 5). Then, the SDN controller reallocates new weights, using our AASRA scheme. The new weights are now in favor of the suffering users, which get benefitted from the next resource allocation by the OLT.

C. ONU and User Setup

For evaluating the proposed schemes, we considered an OLT-ONU setup with five ONUs, each connected to five clients. ONU 1 has all clients of type 1, ONU 2 has all clients of type 2, ONU 3 has all clients of type 3, ONU 4 has all clients of type 4, and ONU 5 has all clients of type 5. So, ONU 5 has all very-high-priority clients whose video resolution requirement is highest. Similarly, ONUs 1, 2, 3, and 4 have their share of requirements based on the acceptable resolutions summarized in Table 1.

This is the scenario considered to analyze the effect of having client-level differentiation that can significantly affect the bandwidth requirements at the ONUs. However, our approach works well for scenarios where different types of clients are supported by a single ONU. From Table 2, we can see that a video of resolution 4 needs approximately 4 times more bandwidth than that of resolution 1. Hence, we can expect that ONU 4, with 5 clients each of type 4, would require 20 times more bandwidth. But it is unrealistic to expect that these clients would in fact generate 20 times more revenue. So, according to different revenue models, the minimum weights in the algorithm can be decided by the network operator. In our simulations, the values used for *buff_min*, *buff_max*, *min_threshold*, and *sbas_threshold* are 8, 16, 2 seconds, and 0.4 (40 percent of total video clients), respectively.

D. Illustrative Results

In this section, the total video stall time per ONU, average duration of stall, average resolution, video-switching rate, and buffer level are plotted for each of the schemes mentioned in Section III-A. In Fig. 8, we plot total stall time for all clients as a percentage of total video time at each ONU. We can see that our proposed AASRA scheme outperforms application-unaware schemes (WRR and RR) in all ONUs (ONUs with different client types). SBAS combined with AASRA can further improve the performance on stall time in all the ONUs. As shown in Fig. 8, our scheme maintains a nearly constant amount of stall time at each ONU ensuring

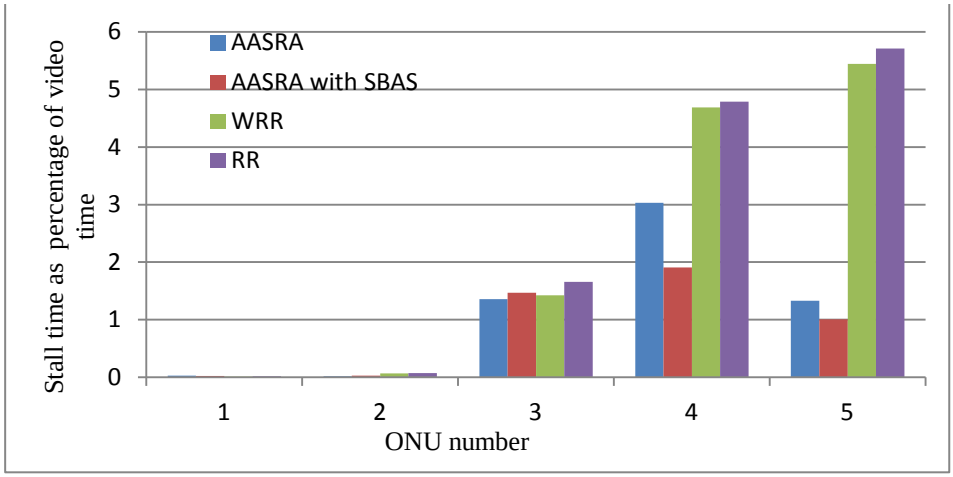

Figure 8: Total stall time (in percentage of total video time) at each ONU for the four schemes.

good QoE for client types 3, 4, and 5 despite the huge variation in the video quality (and hence bandwidth demand) that they exhibit.

This is due to the adaptive weights that change in accordance to the buffer levels. It can be noted that the performance of our schemes is slightly better for clients of type 5 than 3, and 4, due to higher minimum bandwidth given to clients at ONU 5 to reduce their stall times. Weighted Round Robin performs better than Round Robin in terms of average stall time.

In Fig. 9, we plot average video buffer of clients at each ONU. AASRA with SBAS also outperforms application-unaware schemes in terms of average buffer level at the clients. Video buffer size for application-unaware schemes (WRR and RR) is similar to each other on an average and lower compared to both AASRA and AASRA with SBAS. It can be seen as a general trend that the ONU 5 clients have lower buffer levels in chunks. Though it may seem counterintuitive given the fact that they are actually high priority users, it must be taken into consideration that, in terms of buffer size in bytes, one chunk of type 5 is four times the size of a type-1 chunk.

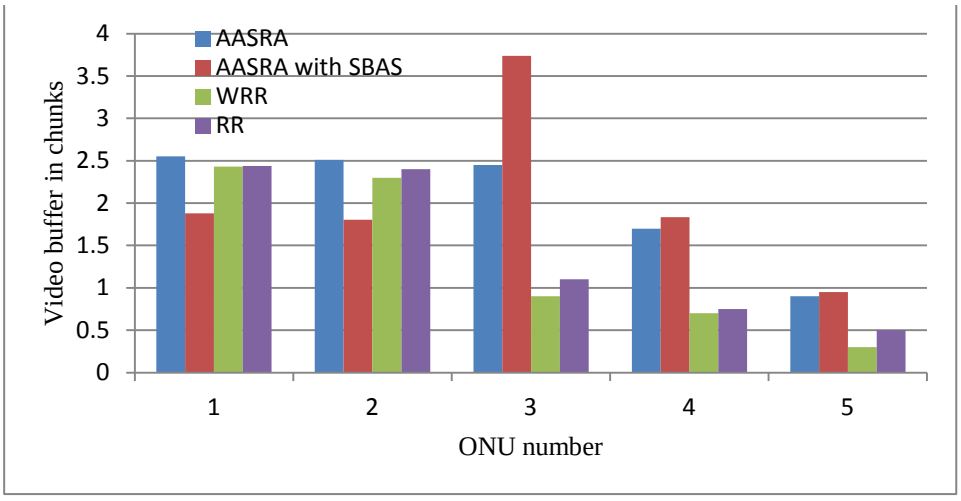

Figure 9: Average video buffer level at each ONU for the four schemes.

Figure 10 shows the average duration of stall for the application-aware and application-unaware schemes. The proposed AASRA and AASRA with SBAS reduce the average duration of stall considerably compared to application-unaware schemes. Since RR performs worse than WRR in providing better QoE for video clients having varying requirements, we do not consider it for further evaluation of the proposed application-aware schemes.

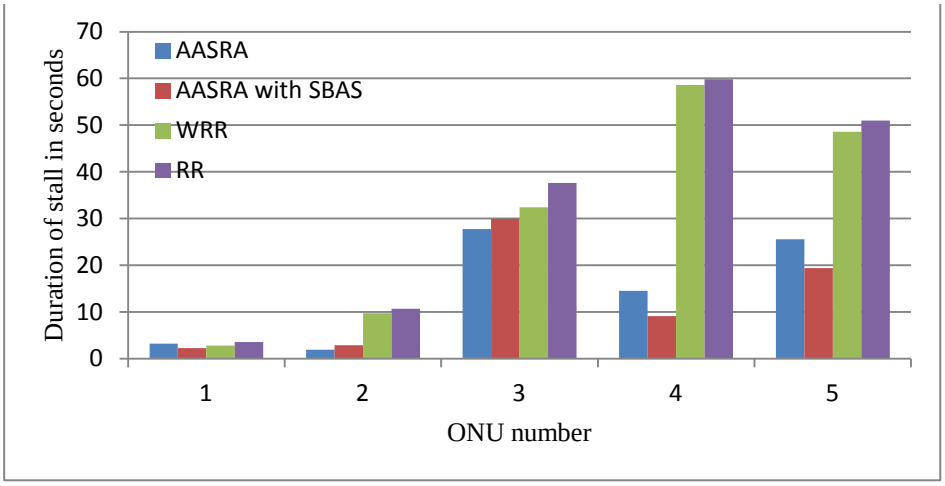

Figure 10: Average duration of stall in a 1000-second video.

All the schemes perform more or less the same in terms of average resolution supported at each ONU, as shown in Fig. 11. This is the case since a high-network-load scenario is studied, and every client gets near to the lowest resolution that they can tolerate. On an average, application-unaware WRR provides slightly better resolution to clients; however, this is because the video stalls are not taken care of by the application-unaware scheme. AASRA provides slightly worse in terms of resolution but having SDN-based adaptive streaming helps in improving the average resolution considerably. AASRA with SBAS provides best of both worlds by giving least stall time and good performance in resolution. AASRA with SBAS performs slightly better than the AASRA scheme in terms of reducing video stall time. However, it can be seen that video quality in terms of resolution improves a lot by using AASRA with SBAS approach. Hence, we can see that interaction of control loops (Fig. 4) is beneficial for the system.

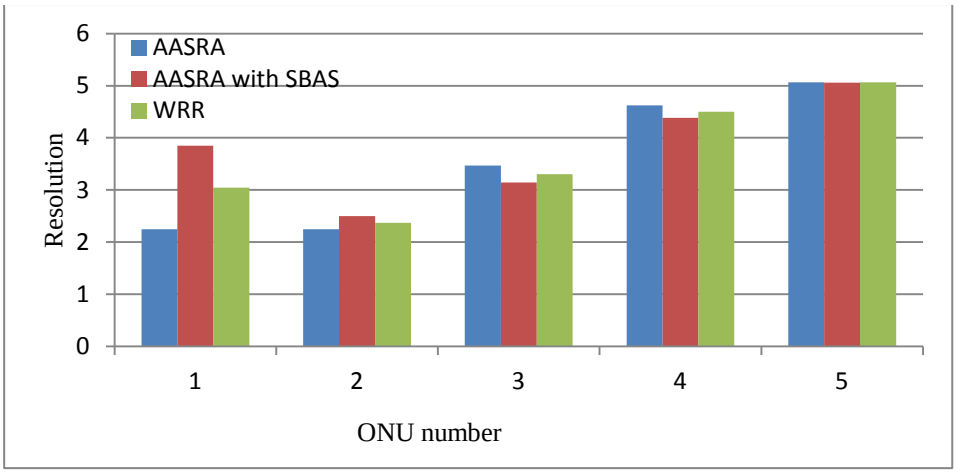

Figure 11: Average resolutions for WRR, AASRA, and AASRA with SBAS.

We define video-switching rate as the rate at which video at the client changes its resolution (due to adaptive streaming). According to [20], video switching is not a very positive experience for the video viewer, although it is not a very significant metric to determine QoE. Figure 12 shows that video-switching rate of application-aware schemes is lesser than application-unaware WRR. This can be explained by the fact that, in an application-unaware network, adaptive streaming can only reduce video stalls by changing the video rates and it does not have control over bandwidth allocated (unlike AASRA), thus leading to higher video-switching rates. Again, SDN-adaptive streaming helps reduce video switching even further compared to AASRA scheme.

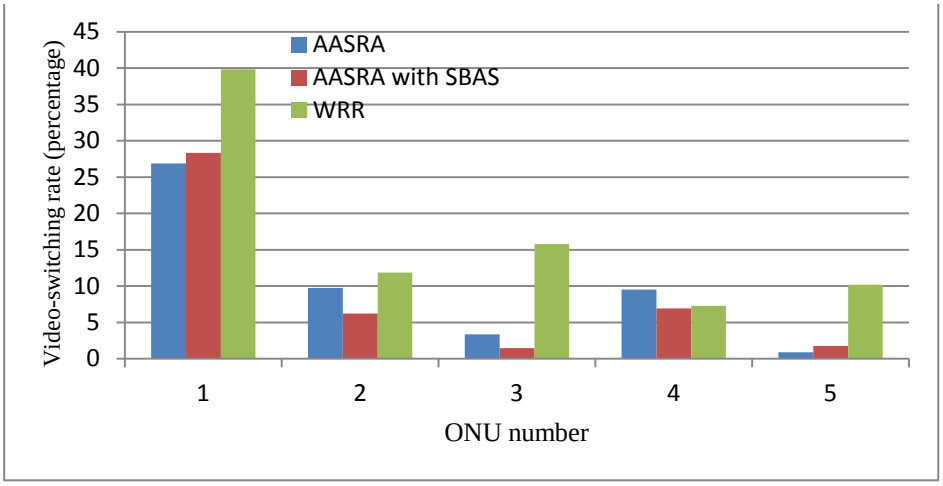

Figure 12: Video switching rate for WRR, AASRA, and AASRA with SBAS.

Figure 13 shows the video stall time for AASRA, AASRA with SBAS, and WRR for decreasing background traffic (high-priority traffic that preempts video). It is expected that video stall will increase as high-priority background traffic increases. AASRA with SBAS performs the best compared to the other schemes. AASRA and AASRA with SBAS perform similar when background traffic is small (and hence there is no bandwidth crunch).

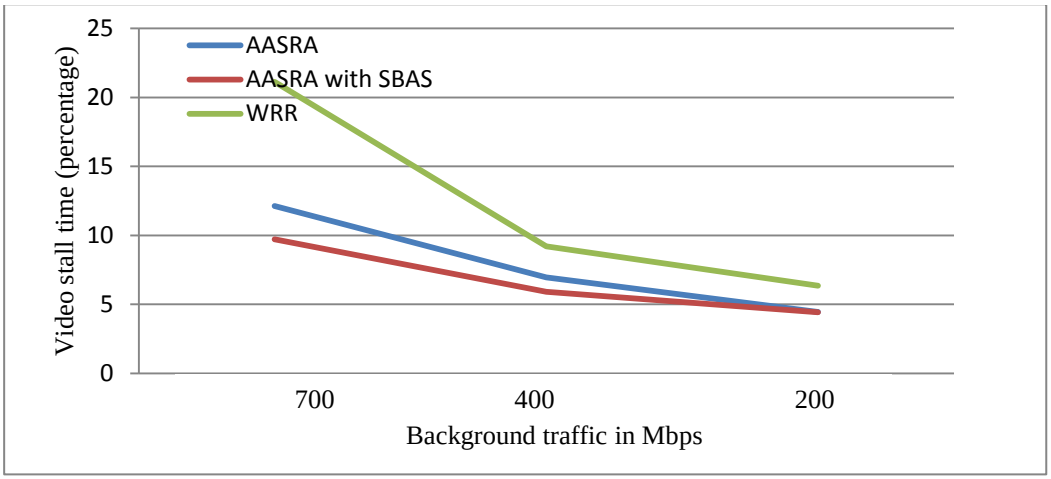

Figure 13: Stall time in percentage for increasing background traffic.

IV. CONCLUSION

The proposed application-aware SDN-enabled EPON architecture provides better client-and-service-level differentiation with the help of an SDN controller and application-state information. This study demonstrates downstream scheduling based on application state of video-streaming application, and the proposed application-aware SDN-enabled resource allocation (AASRA) scheme outperforms static application-unaware schemes in terms of video stall time and buffer level, and hence provides a better video-streaming experience. SDN-based adaptive streaming further improves the performance of the system. This can be extended to other applications such as video conferencing, e.g., Skype, etc. where upstream scheduling can be done using the controller. A hardware-based implementation of our system can be developed to further investigate the feasibility of our model.

REFERENCES

[1] G. Kramer, B. Mukherjee, and G. Pesavento, “IPACT a dynamic protocol for an Ethernet PON (EPON),” *IEEE Communications Magazine*, vol. 40, no. 2, pp. 74-80, Feb. 2002.

[2] P. Bhaumik, S. Thota, K. Zhangli, J. Chen, H. Elbakoury, L. Fang, and B. Mukherjee, “EPON Protocol over Coax (EPoC): Round-trip time aware dynamic bandwidth allocation,” *Proc., 17th International Conference on Optical Network Design and Modeling (ONDM)*, pp. 287-292, Brest, France, April 2013.

[3] P. Chowdhury, B. Mukherjee, S. Sarkar, G. Kramer, and S. Dixit, “Hybrid wireless-optical broadband access network (WOBAN): prototype development and research challenges,” *IEEE Network*, vol. 23, no. 3, pp. 41-48, May-June 2009.

[4] A. Lara, A. Kolasani, B. Ramamurthy, “Network Innovation using OpenFlow: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 493-512, First Quarter 2014.

[5] T. Zinner, M. Jarschel, A. Blenk, and F. Wamser, “Dynamic application-aware resource management using Software-Defined Networking: Implementation prospects and challenges,” *Proc., IEEE Network Operations and Management Symposium (NOMS)*, Krakow, Poland, May 2014.

[6] S. Thomas, “Dynamic adaptive streaming over HTTP -- standards and design principles,” *Proc., Second Annual ACM Conference on Multimedia Systems (MMSys)*, 2011.

[7] B. Badic, T. O'Farrrell, P. Loskot, and J. He, “Energy Efficient Radio Access Architectures for Green Radio: Large versus Small Cell Size Deployment,” *Proc., IEEE 70th Vehicular Technology Conference Fall (VTC 2009-Fall)*, Sept. 2009.

[8] V. Chandrasekhar, J. Andrews and A. Gatherer, “Femtocell networks: a survey,” *IEEE Communications Magazine*, vol. 46, no. 9, pp. 59-67, Sept. 2008.

[9] S. Karthikeyan, Y. Mustafa, S. Shailendra, R. Sampath, and V. Srikanth, “FluidNet: a flexible cloud-based radio access network for small cells,” *Proc., 19th ACM Annual International Conference on Mobile Computing & Networking (MobiCom)*, 2013.

[10] China Mobile, “C-RAN: the road towards green RAN,” White Paper, ver 2 (2011).

[11] A. Dhaini, C. Assi, A. Shami, and N. Ghani, “Adaptive fairness through intra-ONU scheduling for Ethernet passive optical networks,” *Proc. International Conference on Communications (ICC'06)*, pp. 2687-2692, June 2006.

[12] J. Chen, B. Chen, and S. He, “A novel algorithm for intra-ONU bandwidth allocation in Ethernet passive optical networks,” *IEEE Communications Letters*, vol. 9, no. 9, pp. 850-852, 2005.

[13] G. Kramer, B. Mukherjee, S. Dixit, Y. Ye, and R. Hirth, “Supporting differentiated classes of service in Ethernet passive optical networks,” *OSA Journal of Optical Networking*, vol. 1, no. 8/9, pp. 280-298, August 2002.

[14] F. Wamser, D. Hock, M. Seufert, B. Staehle, R. Pries, and P. Tran-Gia, “Using buffered playtime for QoE-oriented resource management of YouTube video streaming,” *Transactions on Emerging Telecommunication Technologies*, vol. 24, no. 3, pp. 288-302, April 2013.

[15] O. Oyman and S. Singh, “Quality of experience for HTTP adaptive streaming services,” *IEEE Communications Magazine*, vol. 50, no. 4, pp. 20-27, April 2012.

[16] A. Curtis, K. Wonho, and P. Yalagandula, “Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection,” *Proc., INFOCOM*, April 2011.

[17] F. Wamser, D. Staehle, J. Prokopec, A. Maeder and P. Tran-Gia, “Utilizing buffered YouTube playtime for QoE-oriented scheduling in OFDMA networks,” *Proc., 24th International Teletraffic congress*, September 2012.

[18] R. Schatz, M. Fiedler and L. Skorin-Kapov, “QoE-Based Network and Application Management,” *Springer International Publishing in Quality of Experience*, pp. 411-426, 2014.

[19] M. Channegowda, R. Nejabati, and D. Simeonidou, “Software-Defined Optical Networks Technology and Infrastructure: Enabling Software-Defined Optical Network Operations [Invited],” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 10, pp. A274-A282, Oct. 2013.

[20] T. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, “A buffer-based approach to rate adaptation: evidence from a large video streaming service,” *Proc., ACM SIGCOMM*, 2014.

[21] C.Gürler, B.Görkemli, G.Saygili, and A.Tekalp, "Flexible Transport of 3-D Video Over Networks," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 694-707, April 2011.