A Hybrid Upstream Bandwidth Allocation Method for Multimedia Communications in EPONs

Jinsuk Baek[1], Min Gyung Kwak[2] and Paul S. Fisher[1]

Abstract – The Ethernet Passive Optical Network (EPON) has been considered to be one of the most promising solutions for the implementation of the Fiber To The Home (FTTH) technology designed to ameliorate the "last mile" bandwidth bottleneck. In the EPON network, an efficient and fair bandwidth allocation is a very important issue, since multiple optical network units (ONUs) share a common upstream channel for packet transmission. To increase bandwidth utilization, an EPON system must provide a way to adaptively allocate the upstream bandwidth among multiple ONUs in accordance to their bandwidth demands and requirements. We present a new hybrid method that satisfies these requirements. The advantage of our method comes from the consideration of application-specific bandwidth allocation and the minimization of the idle bandwidth. Our simulation results show that our proposed method outperforms existing dynamic bandwidth allocation methods in terms of bandwidth utilization.

Keywords: EPON, FTTH, Bandwidth allocation, Optical line terminal, Optical network terminal

1. Introduction

A Fiber To The Home (FTTH) network provides a significant QoS improvement for multimedia communications, since it requires a significantly lower transmission cost when compared to existing coaxial cable-based transmission systems [1]. In order to enable a FTTH network, an *Ethernet Passive Optical Network* (EPON) is used as an access network technology which connects subscribers to their corresponding service provider. The EPON is configured with a point-to-multipoint tree topology; each EPON entity is deployed in the tree. The essential entities typically consist of an *Optical Line Terminal* (OLT), an optical splitter, and multiple *Optical Network Terminals* (ONTs). As shown in Fig. 1, the OLT resides at the service provider's facility. Each subscriber has an ONT, which is connected to an optical splitter by a dedicated optical fiber. For upstream data packets, the optical splitter combines a number of optical signals sent by multiple ONTs into a single shared optical signal, which is then delivered to the OLT. In order to connect to the Internet via a telecommunication router, the OLT performs a conversion between the optical and electrical signals. For downstream data packets, all of the packets are encapsulated as an Ethernet frame and are carried from the OLT to the ONTs through the optical splitter, wherein the packets are replicated.

With the increase in bandwidth requirements, current xDSL or cable Internet services do not have sufficient capacity to support real time multimedia streaming applications, such as on-line games, distance education, video-on-demand, and VoIP [2]. Unfortunately, this issue is also difficult to resolve in EPON networks, since the available bandwidth is still limited over the multiple subscribers. More importantly, a constant upstream bandwidth allocation is required to support applications possessing burst traffic patterns. Therefore, an efficient allocation of the limited upstream bandwidth to the appropriate subscriber is a crucial issue in order to maximize the bandwidth utilization in an EPON-based FTTH network.

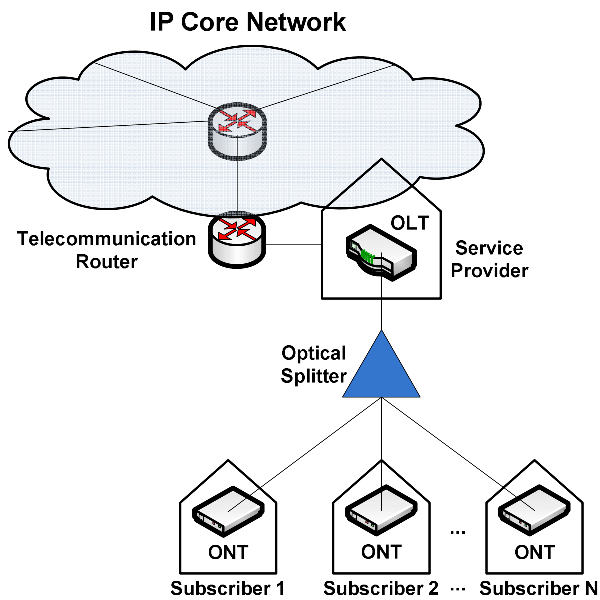

Fig. 1. The FTTH internet access network architecture

This allocation demand has led to numerous proposals, which can be broadly categorized into two methods –

† Corresponding Author: Jinsuk Baek
[1] Department of Computer Science, Winston-Salem State University / Winston-Salem, NC 27110, USA {baekj, fisherp}@wssu.edu
2 Department of Computer Science, Illinois Institute of Technology / Chicago, IL 84112, USA mkwak@hawk.iit.edu
Received: July 15, 2012; Accepted:

namely static and dynamic [3]. Earlier EPON networks used the static method to allocate the upstream bandwidth. That is, every ONT has a predefined amount of bandwidth. This method is efficient enough as long as all of the services in the network require a constant bandwidth allocation or the capacity of the installed upstream bandwidth is too low to enable efficient allocation. However, this suboptimal allocation is not appropriate for a high speed, high utilization EPON network. The dynamic method overcomes the weaknesses of the static method by allowing the OLT to control the bandwidth allocation for the ONTs. This type of allocation method depends on an ONT request, upstream traffic measurement, or the consideration of the *Service Level Agreement* (SLA) of the subscriber. However, the allocation method generates frequent control messages from the ONTs, which causes further bandwidth consumption.

We present a new *Hybrid Upstream Bandwidth Allocation* (HUBA) method that outperforms the existing methods by considering the application-specific aspects of the ONTs. The HUBA method maximizes the advantages of the static method by allocating a constant upstream bandwidth to the ONT as long as the ONT remains tied to the same application. This reduces the traffic overhead caused by control messages that otherwise would be frequently exchanged between the ONT and ONUs. Instead, the control message exchange is only needed when the ONU switches to another application that requires a different amount of bandwidth from the previous application. This alleviates the processing overhead at the OLT by eliminating unnecessary idle messages periodically sent by the ONTs.

The rest of this paper is organized as follows: Section 2 describes the related work. In Section 3, we propose the new hybrid upstream bandwidth allocation method that takes into account the application-specific aspects. Section 4 discusses the performance evaluations of the proposed method. Finally, we conclude this paper in Section 5.

2. Related Work

In an EPON network bandwidth allocation can be implemented using one of the two general methods – static and dynamic. In the static methods, the bandwidth for each ONT is statically allocated by the OLT in a TDM-like manner. However, this statically allocated bandwidth assigned to each ONT cannot be shared with the other ONTs, even when the ONT changes to an idle status, all of which results in the degradation of the overall bandwidth utilization.

The dynamic method can be further subdivided into two different schemes – namely *status reporting* (SR) and *non-status reporting* (non-SR). In the SR scheme, the OLT requires the ONTs to send a status report during every time slot. The information sent by ONTs is used to determine the bandwidth required for each ONT in accordance to its demands.

One should note that the EPON standard only provides the tools to implement bandwidth allocation but leaves the actual allocation scheme open. This has led to many different implementation methods. One example uses an interleaved polling algorithm [4]. At a given time, the OLT knows 1) how many bytes are waiting in every ONT's sending buffer and 2) the round trip time to every ONT. The OLT keeps this information in its polling table and sends a grant message to the ONTs containing the ID of the destination ONT and the size of the granted window. The individual ONTs are then able to use the bandwidth up to the granted window size. At the end of the time slot, the ONTs generate a request message containing the remaining buffer size and send it to the OLT. When the OLT receives the request messages, it updates its polling table, generates grant messages based on the updated polling table, and then sends a grant message to the ONTs.

The SR scheme can be also implemented by using a priority scheduling method [5], which categorizes the services into three parts: high priority, medium priority, and low priority. The high priority service supports applications that require bounded end-to-end delays and jitter specifications. The medium priority service implements a traffic class for applications that are not delay sensitive but require a guaranteed bandwidth. The low priority service is provided to implement a best-effort traffic class. In order to handle these classified bandwidths, the method uses strict priority queuing and two types of control messages: gate messages and report messages. The OLT periodically generates a gate message containing the grant level, grant start time, and grant length field and multicasts it to the ONTs. The report messages generated by the ONTs provide the multiple queue length information from the ONTs. However, the performance of the SR schemes is limited because every ONT needs to generate and send a grant request message in every service cycle, causing additional bandwidth consumption. In addition, the calculation of the window size for every ONT during every service cycle results in time and calculation overhead, especially when the OLT has large number of ONTs. Moreover, the SR scheme does not provide and guarantee QoS in all network configurations while it is efficiently using available resources.

Under the non-SR scheme [6-11], the OLT estimates the window sizes for ONTs based on the bandwidth usage during the previous service cycle. The ONT sends an idle message to the OLT when it does not have any data to transmit. The OLT counts the number of idle messages and uses it to calculate the bandwidth utilization and decreases the window size in the next service cycle. Conversely, when the ONT keeps on transmitting packets, the OLT increases the window size for the dedicated ONT. Whereas the non-SR scheme does not require report messages from the ONTs, which causes idle status time in every time slot, it eventually requires more bandwidth resources due to the overestimation of window sizes.

3. The Proposed Method

All of the existing schemes allocate bandwidth on the basis of different service types. It means they apply different bandwidth allocation methods that depend on the service type rather than on the specific application. Every user in an EPON network uses the same or different applications at a certain times and is likely to switch to another application having a different service profile from the previous application. In order to efficiently manage the limited bandwidth resources with this application variation, we propose a simple but efficient HUBA method that incorporates an application-adaptive perspective. To simplify our description, we assume that each ONT has a fixed number of applications running, since the proposed method can be applied in a straight forward manner to environments with a variable number of applications. Let T be a set of ONTs. There are n ONTs, which are connected to one OLT through an optical splitter.

$$T = \{\text{ONT}_1, \text{ONT}_2, \ldots, \text{ONT}_n\}. \quad (1)$$

For the basic operation of the proposed method we propose the following sequence of actions:

(1) Each ONT_i has its own time slot t_i and sends a bandwidth request message $\text{Req\_Msg}(k)_i$ to its OLT during every service cycle $s(k)_i$, which is defined as:

$$s(k)_i = t_i, \text{ for } k = 1$$
$$s(k+1)_i = s(k)_i + t_i, \quad \text{for all } k \geq 1 \quad (2)$$

For $1 \leq i \leq n$, and $k \geq 1$, the message $\text{Req\_Msg}(k)_i$ contains current application type App_T_i, packet size Size_P_i, and requested bandwidth size $\text{Req\_B}(k)$.

$$\text{Req\_Msg}(k)_i = \{\text{App\_T}(k)_i, \text{Size\_P}(k)_i, \text{Req\_B}(k)\} \quad (3)$$

(2) Once the OLT receives a message from ONT_i, it calculates the bandwidth based on two variables and sends grant message $\text{Grant\_Msg}(k)_i$ containing the size of allocated bandwidth Grant_B_i for ONT_i.

$$\text{Grant\_Msg}(k)_i = \{\text{Grant\_B}(k)_i \mid 1 \leq i \leq n, k \geq 1\} \quad (4)$$

(3) After receiving $\text{Grant\_Msg}(k)_i$, ONT_i starts transmitting its data packets via the allocated bandwidth resources at service cycle $s(k)_i$.

(4) Once current cycle $s(k)_i$ is finished, ONT_i checks to see if the target application has been changed.

- If there is no change, ONT_i does not generate any control message for the next cycle $s(k+1)_i$, maintaining the bandwidth size to be the same as the previous cycle's $s(k)i$.
- Otherwise, that is if ONT_i has switched to another application having a different service requirement, it generates a new message $\text{Req\_Msg}(k+1)_i$ for the next service cycle $s(k+1)_i$ and sends it to its OLT.

(5) Whenever the OLT receives a new Req_Msg message from ONT_i, it calculates the new bandwidth Grant_B(k+1) for the ONT by referencing the information in the received message. This bandwidth size is attached to the Grant_B(k+1) message and sent to the ONT. This process is repeated until all of the ONTs are disconnected from the OLT.

For the bandwidth calculation, we require the OLT to consider two variables, threshold CT and max_BW(APP_T), which is the maximum available bandwidth allowed for a specific application class App_T. Once the difference bandwidth size $B_{<k,\, k+1>}$ between two consecutive service cycles k and k+1 is calculated and the total available bandwidth TB is always less than or equal to $n\times$max_BW(APP_T), we consider four different cases used to calculate the bandwidth resource for a given ONT.

$$B_{<k,\, k+1>} = \text{Req\_B}(k+1) - \text{Grant\_B}(k) \quad (5)$$

Case 1: If $\text{CT} < B_{<k,\, k+1>} \leq \text{max\_BW(APP\_T)}$, then the OLT allocates the bandwidth for the ONT using:

$$\text{Grant\_B}(k+1) = \min\{\text{max\_BW(APP\_T)}, \text{Req\_B}(k+1)\}$$

Case 2: If $\text{max\_BW(APP\_T)} < B_{<k,\, k+1>}$, then the OLT allocates the bandwidth for the ONT using:

$$\text{Grant\_B}(k+1) = \text{max\_BW(APP\_T)}$$

Case 3: If $0 < B_{<k,\, k+1>} \leq \text{CT}$, then the OLT allocates the bandwidth for the ONT using:

$$\text{Grant\_B}(k+1) = \min\{\text{CT} + \text{Grant\_B}(k), \text{max\_BW(APP\_T)}\}$$

Case 4: If $B_{<k,\, k+1>} \leq 0$, then the OLT allocates the bandwidth for the ONT using:

$$\text{Grant\_B}(k+1) = \text{Req\_B}(k+1)$$

In addition to the defined operations, we further consider special cases in order to increase bandwidth utilization. The first case occurs when there are m ($<n$) active applications. In this case, there must be unused bandwidth resources because:

$$m\times\text{max\_BW(APP\_T)} < n\times\text{max\_BW(APP\_T)} \quad (6)$$

In order to utilize the unused bandwidth resources, the HUBA method increases the value of max_BW(APP_T) to upper_max_BW(APP_T) until the following condition is satisfied:

m×upper_max_BW(APP_T) ≈ n×max_BW(APP_T) (7)

Another special case is when one user is simultaneously running multiple applications. That is, there are m ($>n$) active applications. In this case, one user consumes a disproportionally large amount of bandwidth and thereby can lower the overall performance. To minimize this side-effect caused by unfair bandwidth consumption, the HUBA method decreases the value of max_BW(APP_T) to lower_max_BW(APP_T) until the following condition is satisfied:

m× lower_max_BW(APP_T) ≈ n×max_BW(APP_T) (8)

The proposed HUBA method overcomes the limitations found in the existing methods by combining their advantages. First, it maximizes the advantages of the static method. In a legacy static method, every ONT has a statically allocated bandwidth, even though there are different applications requiring different amounts of bandwidth. The proposed HUBA method allows the ONT to maintain the static bandwidth allocation as long as the ONT utilizes the same application. Once the user switches to another application, the proposed HUBA method adaptively switches to the dynamic allocation mode. As a result, it uses a static allocation within an application but a dynamic allocation for different applications. More importantly, this feature reduces the traffic in both directions as related to the exchange of control messages in every service cycle. Instead, this control message is exchanged only when the ONT switches to another application that requires a different amount of bandwidth than the previous application. Second, the HUBA method also maximizes the advantages of the dynamic method by greatly reducing the number of idle messages normally used to indicate that the OTL has no packet to transmit. These idle messages lower the performance of the EPON network because they constantly consume bandwidth resources until an ONT stops transmitting idle messages. Instead, the HUBA method sends only one idle packet to indicate no packet transmission at the beginning of the current service cycle. When the OLT receives this idle packet, it removes the allocated bandwidth for the ONT and uses the freed bandwidth resources for the other ONTs. This further increases bandwidth availability and so results in better bandwidth utilization.

4. Performance Evaluation

In order to clarify the performance improvement of the proposed method, we configured the test topology depicted in Fig. 1. We utilized eight ONTs, which were connected to an OLT through an optical splitter supporting an upstream bandwidth capacity of 1Gbps. All of the simulations were conducted in an ns-2 environment under this point-to-multipoint topology.

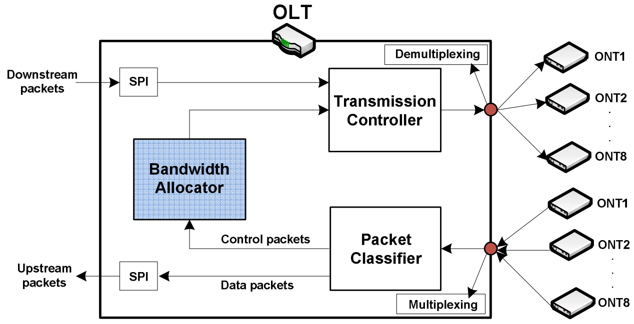

Fig. 2. The OLT system structure

Our framework used the following parameters:

First, as shown in Fig. 2, the OLT consists of three functional units – a packet classifier, a bandwidth allocator, and a transmission controller. The packet classifier categorizes the incoming packets into data packets and control packets. The data packets are transferred to an upper layer protocol stack through a *Serial Peripheral Interface* (SPI); the control packets are delivered to the bandwidth allocator, where a new bandwidth allocation for the ONT is calculated. The granted bandwidth is sent to the ONT through a transmission controller together with the other necessary control information, such as the ONT timer and guard timer, which are designed to avoid transmission collisions among the ONTs.

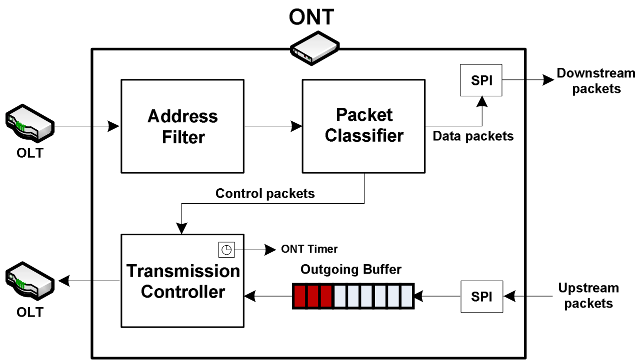

Fig. 3. The ONT system structure

Second, the ONT generates data packets to be sent to the OLT. We set this interval to 64μs. The ONT has four functional units, as shown in Fig. 3. The address filter decides whether or not the received packets are destined for the ONT. After the packets are filtered, they are classified into data packets and control packets in the packet classifier. The data packets are transferred to an upper layer protocol stack. The control packets are referred to the transmission controller to generate a Req_Msg for the next service cycle. This message is sent to the OLT as soon as the ONT timer has expired. Before the timer expires, all of the generated packets are kept in an outgoing buffer.

In the simulation, we implemented two bandwidth allocation algorithms: *Dynamic Bandwidth Allocation* (DBA) with SR and the proposed HUBA algorithm, both of which were simulated at the bandwidth allocator unit on the OLT side. Since our main focus was to observe the operation of the proposed method and compare it to other methods, we did not consider the impact of the different outgoing buffer sizes. Therefore, we assumed that the OLT had a sufficiently large buffer to accommodate all of the generated packets. Similarly, we assumed that each ONT employed only three types of applications requiring low, medium, and high traffic services. We initially set the threshold CT to 15,000 bytes, which we then changed to observe the impact of this value on the overall performance. The simulation parameters are summarized in Table 1.

Table 1. The Simulation Parameters

Parameters	Values
Number of OLTs	1
Number of ONTs	8
Upstream bandwidth capacity	1Gbps
Packet generating interval	64 μs
Round trip time (RTT)	0.1 ms
Guard timer	2 μs
Threshold CT	15,000 bytes

Our simulation was performed for 10 seconds using eight ONTs having up to three different applications requiring different bandwidth amounts. We did not consider static allocation because it is only useful in a very special environment. The non-SR DBA method was also excluded in our simulation because the expected benefit of the proposed method is linear to the number of generated idle messages. For the DBA-SR method, we constructed 25,000 bytes for the max_BW for each service cycle.

For the HUBA method, we constructed 17,000 bytes, 25,000 bytes, and 34,000 byte max_BWs for the low, medium, and high traffic applications, respectively. In addition, at the 3rd second some of the ONTs' changed their application type, requiring different packet sizes. ONT1 and ONT3 changed to applications requiring increased packet sizes, whereas ONT2, ONT6, and ONT8 changed to applications requiring decreased packet sizes. ONT4, ONT5 and ONT7 did not change their applications. The packet sizes vary from 1,000 bytes to 7,000 bytes, depending upon the application.

Figs. 4 and 5 show the simulation results for the two different methods. As can be seen in Fig. 5, the proposed HUBA method allows ONT1, running an application requiring more bandwidth, to adaptively adjust its upstream bandwidth from 79 Mbps to 127 Mbps. In addition, it maximizes the bandwidth allocation for ONT2, which requires the highest bandwidth allocation, allowing 160-164 Mbps. With the DBA-SR method, the same ONT has to remain with the medium bandwidth amount, allowing only a maximum of 138 Mbps. Note that ONT7 has a lower

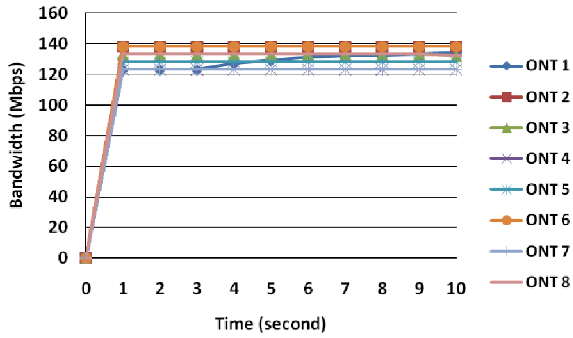

Fig. 4. The DBA-SR simulation results

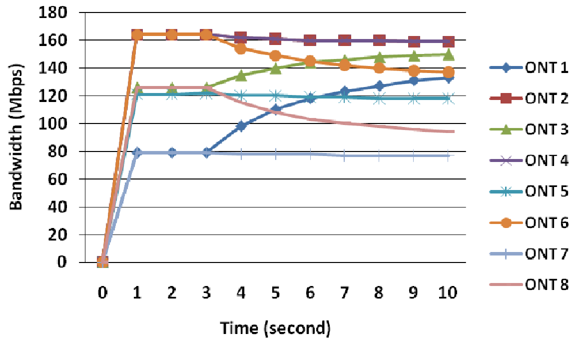

Fig. 5. The HUBA simulation results

granted bandwidth with the HUBA compared to the DBA-SR. This is because 1) the ONT7 stays with the same application as that of the previous service cycle, and 2) the HUBA lets other ONTs maximize the bandwidth allocation for their burst transmission.

Figs. 6 and 7 show the bandwidth allocation at the 6th second for a legacy HUBA and the HUBA with adjustments when the value *m* and *n* are not equal. Fig. 6 shows the simulation results when ONT3 stops transmitting at the 4th second, which results in smaller *m* than *n* values. In our simulation, we allowed each ONT to increase their max_BW up to upper_max_BW, which is 10% larger than max_BW. This adjustment shows a small but meaningful improvement in the average bandwidth utilization over all of the ONTs. Fig. 7 shows the simulation results when ONT1 simultaneously runs three applications, one application from the beginning and two additional applications at 4 seconds. In this case, the value of *m* becomes larger than *n*. The adjustment algorithm requires each ONT to decrease their max_BW to lower_max_BW, which is 20% smaller than max_BW. Compared to the legacy HUBA, the adjustment shows a 5.18 % improvement in the bandwidth utilization on average. Finally, we must mention that we performed simulations for different threshold CT values ranging from 2,000 bytes to 30,000 bytes. The results of these changes did not show any significant impact to the overall performance of all of the ONTs. Therefore, this value should be set empirically depending on the current network conditions.

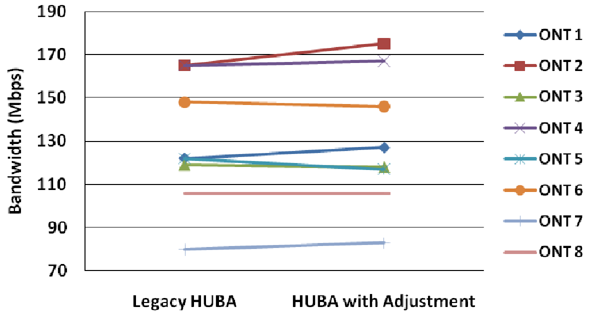

Fig. 6. The HUBA simulation results when *m* is less than *n*

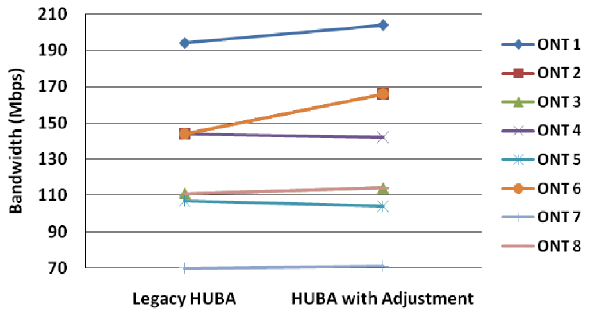

Fig. 7. The HUBA simulation results when *m* is bigger than *n*

5. Conclusion

We presented a new method that improves the bandwidth allocation for multiple applications being executed within the home setting. It is evident that applications involving streaming media require a considerable amount of bandwidth at a constant delivery rate. Unfortunately, these applications are not scheduled, so static scheduling is not appropriate. Therefore the dynamic allocation of bandwidth on a real time basis is required. The drawback to traditional dynamic allocation methods is the necessary overhead of the control packets that are required to manage the bandwidth allocation. Our technique is designed to minimize these control packets, and to dynamically allocate bandwidth to those applications requiring such service. We have shown by our simulation results that we can indeed improve the bandwidth allocation in the dynamic environment, and do it in a more efficient manner. This allows the high demand users to maximize their opportunities to receive uninterrupted streams while all other users have access to the necessary bandwidth for their usage. We believe that this algorithm is a step in the right direction to improve bandwidth allocation to the home.(Ed Note: An interesting and very well written paper. I have edited it in order to make it more readable and to increase understanding. Please read it carefully in order to ensure that I have not inadvertently altered any meaning. My name is Mark S, and I hope you will use Nurisco for all your editing needs. Thank you!)

References

[1] G. Kramer, and G. Pesavento, “Ethernet Passive Optical Network (EPON): Building a Next-Generation Optical Access Network,” *IEEE Communication Magazine*, 40(2): 66–73, Feb. 2002. Article (CrossRef Link)

[2] J. Zheng, and H. T. Mouftah, “A Survey of Dynamic Bandwidth Allocation Algorithms for Ethernet Passive Optical Networks,” *Optical Switching and Networking*, 6: 151–162, Mar. 2009. Article (CrossRef Link)

[3] F. O. Haran, and A. Sheffer, “The Importance of Dynamic Bandwidth Allocation in GPON Networks,” White paper, PMC, last accessed in July 2012. Article (CrossRef Link)

[4] G. Kramer, B. Mukherjee, and G. Pesavento, “IPACT: A Dynamic Protocol for an Ethernet PON (EPON),” *IEEE Communication Magazine*, 40(2): 74–80, Feb. 2002. Article (CrossRef Link)

[5] S. Choi, and J. Huh, “Dynamic Bandwidth Allocation Algorithm for Multimedia Services over Ethernet PONs,” *ETRI Journal*, 24(6): 465–468, Dec. 2002. Article (CrossRef Link)

[6] S. Yin, Y. Luo, N. Ansari, and T. Wang, “Stability of Predictor-Based Dynamic Bandwidth Allocation over EPONs,” *IEEE Communication Letters*, 11(6): 549–551, Jun. 2007. Article (CrossRef Link)

[7] H. C. Leligou, C. Linardakis, K. Kanonakis, J. D. Angelopoulos, and T. Orphanoudakis, “Efficient Medium Arbitration of FSAN-Compliant GPONs,” *International Journal of Communication Systems*, 19(5): 603–617, Jun. 2006. Article (CrossRef Link)

[8] M. Han, H. Yoo, B. Yoon, B. Kim, and J Koh, “Efficient Dynamic Bandwidth Allocation for FSAN-Compliant GPON,” *Journal of Optical Networks*, 7(8): 783–795, Aug. 2008. Article (CrossRef Link)

[9] B. Kantarci, and H. T. Mouftah, “Delay-Constrained Admission and Bandwidth Allocation for Long-Reach EPON,” *Journal of Networks*, 7(5): 812–820, May 2012. Article (CrossRef Link)

[10] X. Shao, Y-K. Yeo, Z. Xu, X. Cheng, and L. Zhou, “Shared-Path Protection in OFDM-Based Optical Networks with Elastic Bandwidth Allocation,” Optical Fiber Communication Conference, pp. OTh4B, Mar. 2012. Article (CrossRef Link)

[11] K. Christodoulopoulos, I. Tomkos, and E. A. Varvarigos, “Elastic Bandwidth Allocation in Flexible OFDM-Based Optical Networks,” Journal of Lightwave Technology, 29(9):1354–1366, May 2011. Article (CrossRef Link)

Jinsuk Baek is an Associate Professor of Computer Science at the Winston-Salem State University (WSSU), Winston-Salem, NC. He is the director of the Network Protocols Group at WSSU. He received his B.S. and M.S. degrees in Computer Science and Engineering from Hankuk University of Foreign Studies (HUFS), Korea, in 1996 and 1998, respectively, and his Ph.D. in Computer Science from the University of Houston (UH) in 2004. Dr. Baek was a post doctorate research associate at the Distributed Multimedia Research Group at UH. He acted as a consulting expert on behalf of Apple Computer, Inc in connection with the Rong and Gabello Law Firm, which serves as legal counsel to Apple computer. He has served as an editor for KSII Transactions on Internet and Information Systems. He also served or is currently serving as a reviewer and Technical Program Committee for many important journals, conferences, symposiums, and workshops in the Computer Communications Networks area. His research interests include wireless sensor networks, scalable reliable communication protocols, mobile computing, network security protocols, proxy caching systems, and formal verification of communication protocols. He is a member of the IEEE.

Min Gyung Kwak received his B.S. degree in Computer Science from Winston-Salem State University (WSSU), Winston-Salem, NC in 2009, and his M.S. degree in Computer Science from the Illinois Institute of Technology (IIT), Chicago, IL in 2012. He worked for Wake Forest Baptist Medical Center as an Intern. He was a member of the Network Protocol Research Group at WSSU. He is currently working at Samsung Techwin. His research interests include network communication protocols, and network security protocols.

Paul S. Fisher is R. J. Reynolds Distinguished Professor of Computer Science at the Winston-Salem State University (WSSU), Winston-Salem, NC. He is the director of the High Performance Computing Group at WSSU. He received his B.A. and M.A. degrees in Mathematics from the University of Utah and his Ph.D. in Computer Science from Arizona State University. He has written and managed more than 100 proposal efforts for corporations and the DoD involving teams of 1 to 15 people. He has worked as a consultant to the U.S Army, U.S Navy, U.S Air Force and several companies over the years. In the 1990's he commercialized an SBIR funded effort and built Lightning Strike, a wavelet compression codec, then sold the company to return to academia. His current research interests include wired/wireless communication protocols, image processing, and pattern recognition.