Proceedings of the APAN – Research Workshop 2019
ISBN 978-4-9905448-9-8

A Modified Deficit Weighted Round Robin traffic Scheduling Algorithm for GPON Networks

Md. Hayder Ali and Mohammad Hanif Ali

***Abstract*—In this paper, we propose the modified deficit weighted round robin (MDWRR) traffic scheduling algorithm for Gigabit Passive Optical Network (GPON), which guarantees the real-time priority traffic. The proposed scheduling algorithm is a variation of the deficit weighted round robin (DWRR) algorithm and it assures the highest priority traffic transmission with minimization of delay. WRR algorithm to be aware of bandwidth and improves the fairness. But for certain traffic types, fairness is not the desired behavior. To achieve predictable service for sensitive, real-time traffic, a priority level for scheduling needs to be introduced. By enabling strict priority, or by offering several priority levels and using DWRR scheduling between queues with the same priority levels, service assurance with regards to delay and loss protection can be achieved for demanding traffic types, such as voice and real-time broadcasting. By offering several priority levels and using DWRR scheduling between queues with the same priority levels, service assurance with regards to delay and loss protection can be achieved for demanding traffic types, such as voice and real-time broadcasting.**

***Index Terms*—MDWRR, GPON, DBA.**

I. Introduction

ACCessing Internet is going to be a fundamental right like other basic human rights. No matter how often a user uses Internet, waiting for a web page to load is really an annoyance. To overcome this situation, Internet speed has increased significantly in the past decade to keep pace with the demand of users, new services and bandwidth-hungry applications. These demands include as multimedia content-based e-commerce, video on demand, high definition TV, IPTV, online gaming, social media, etc. The communication protocols for multimedia traffic have received a great deal of attention in the past few years. Since multimedia traffic must support various types of traffic simultaneously, it is crucial to process data according to its characteristics. Thus, protocol designers have to grasp the characteristics of traffic and select a processing method suitable for the performance requirements. For instance, real-time audio traffic in a voice service requires rapid transmission, but the loss of a small amount of audio information is tolerable. On the other hand, the transfer of a text file should guarantee 100% reliable transfer; real-time delivery is not of primary importance in this case. Real-time video service, such as video on demand (VOD), requires not only rapid transfer but also high reliability. When a piece of video information is lost, its quality of service (QoS) is degraded. Therefore, multimedia communication protocols should be designed to provide the performance requirements of a wide range of multimedia services [1].

During the operation of the network, the Optical Line Terminal (OLT) assigns grants to a given Optical Network Unit (ONU) based on the aggregate effective bandwidth of the traffic of the QoS queues at the ONU. Roughly speaking, a given ONU is assigned grants proportional to the ratio of the aggregate effective bandwidth of the traffic of the ONU to the total aggregate effective bandwidth of the traffic of all ONUs supported by the OLT. In turn, a given ONU uses the grants that it receives to serve its QoS queues in proportion to the ratio of the effective bandwidth of the traffic of a queue to the aggregate effective bandwidth of the traffic of the QoS queues supported by the ONU. A given ONU uses the grants not utilized by QoS queues to transmit from best effort queues [2].

Development of scheduling algorithms used within switch architecture is very important. It is especially important when multimedia traffic is observed, because this traffic can support various types of traffic simultaneously. Therefore, it is necessary to develop and implement scheduling algorithms capable of providing different services for different types of traffic. For example, real-time audio traffic requires low delay

Manuscript Received: 30-04-2019".
Md. Hayder Ali (Phone: +8801712993175, E-mail: hayder.iict@gmail.com) and Mohammad Hanif Ali (Phone: +8801712596955, E-mail: hanif_ju03@juniv.edu) are with Department of Computer Science and Engineering, Jahangirnagar University, Savar, Dhaka, Bangladesh.

but allows small loss of data. To achieve low latency and fast transfer the priority traffic a modified deficit weighted round robin traffic scheduling algorithm is proposed in this paper. However, this algorithm has major implementation drawback because it requires knowledge of incoming traffic in order to calculate weights of traffic flows.

The rest of this paper is organized as follows. Queueing system for Dynamic Bandwidth Allocation (DBA) implementation in GPON networks is stated in section 2. A brief about deficit weighted round robin is presented at Section 3. The modified deficit weighted round robin (MDWRR) is introduced in Section 4. In Section 5, pseudo code of MDWRR is briefly described. In Section 6, Benefits of MDWRR is stated while limitation of it at section 7. Finally, Section 8 draws a conclusion to this paper.

II. QUEUEING SYSTEM IN GPON DBA

All ONUs report their upstream data queue occupancy, to be used by the OLT calculation process. Each ONU may have several Transmission containers (T-CONTs), each with its own traffic class. By combining the queue occupancy information and the provisioned Service Level Agreement (SLA) of each T-CONT, the OLT can optimize the upstream bandwidth allocation. ONUs do not provide explicit queue occupancy information. Instead, the OLT estimates the ONU queue status, typically based on the actual transmission in the previous cycle. With the shift to more symmetrical networks, it is critical to correctly design the ONU upstream queue depth. It is important to find the balance between sufficient queuing that prevents packet loss, but does not add latency that impacts TCP performance. The Bandwidth Allocation algorithm has a major impact on minimizing latency and on the required buffer size. To ensure smooth service with priority traffic and low delay, we had used Deficit Weighted Round Robin (DWRR) algorithm, we were not getting our desire result for low latency and high priority traffic first. We have modified it and found a better result compare to DWRR.

III. DWRR ALGORITHM

In the classic DWRR algorithm, the scheduler visits each non-empty queue and determines the number of bytes in the packet at the head of the queue. The variable Deficit_Counter is incremented by the value quantum. If the size of the packet at the head of the queue is greater than the variable Deficit_Counter, then the scheduler moves on to service the next queue. If the size of the packet at the head of the queue is less than or equal to the variable Deficit_Couner, then the variable Deficit_Counter is reduced by the number of bytes in the packet and the packet is transmitted on the output port. The scheduler continues to dequeue packets and decrement the variable Deficit_Counter by the size of the transmitted packet until either the size of the packet at the head of the queue is greater than the variable Deficit_Counter or the queue is empty. If the queue is empty, the value of Deficit_Counter is set to zero. When this occurs, the scheduler moves on to service the next non-empty queue [9]. (Figure 1).

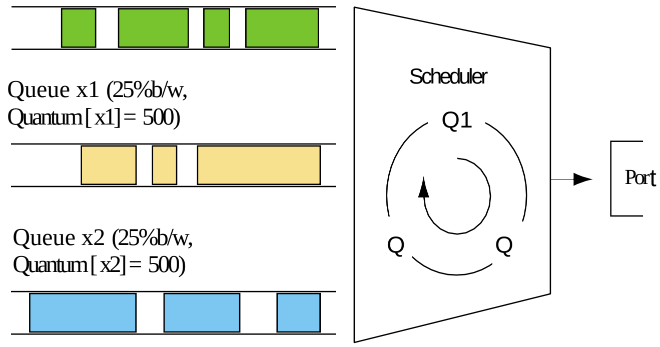

Fig. 1. Deficit Weighted Round Robin Queuing
Source: Juniper Networks, Inc. (By Chuck Semeria)

IV. MDWRR ALGORITHM

As DWRR is not enough for guarantee the priority traffic. We have modified it so that it could calculate the amount of traffic and could assign a priority. The different priority levels mean that there are multiple levels of scheduling. Queue two (Q2) and queue 3 (Q3) have a high priority level. If there are packets in these two queues and if they have enough credit state, they are serviced before the two low-priority queues.

Let us evaluate the first scheduling turn. Because MDWRR separates queues with the same priority level, Q2 is scheduled to receive service in the first cycle. The quantum for Q2 is 200. Thus, its credit state is 200 and it can clear one 200- byte packet from the queue. Q2′ s deficit counter value is now 0, and the deficit counter must be greater than zero to receive service in the same scheduling cycle. The next queue visited is the next priority-high queue, Q3. This queue schedules one packet, but then enters into a negative credit state because of its quantum. Q3′ s credit state is 100. When removing 1000 bytes from the queue, the value of the deficit counter becomes -900 (Figure 2).

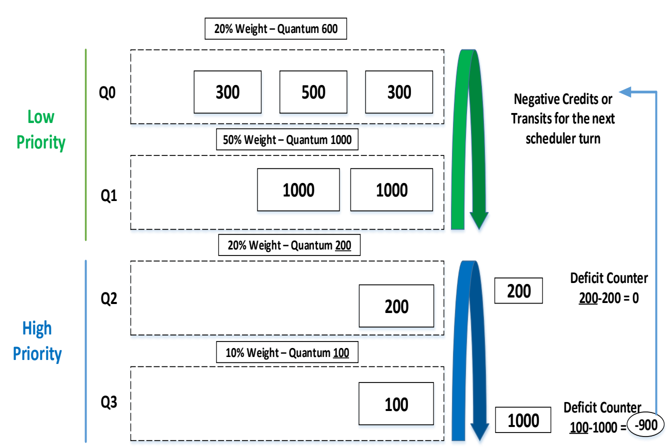

Fig. 2. MDWRR scheduling, turn 1 Q2 and Q3

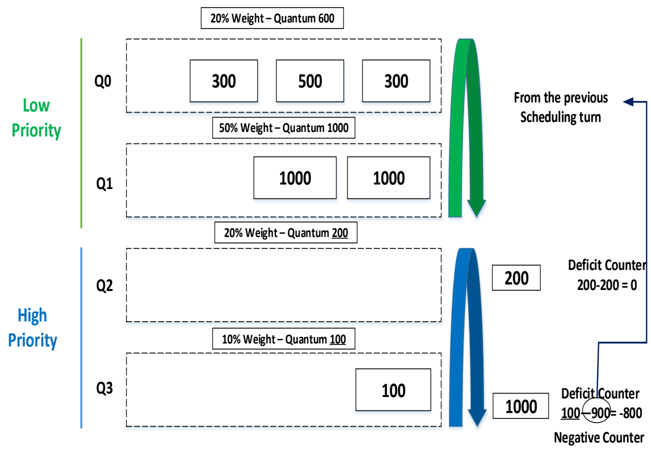

Fig. 3. MDWRR scheduling, turn 2 Q2 and Q3

At the next scheduling turn, Q2 is visited again because its priority level is high and because it contains packets. The quantum is 200 bytes, which is also the number of bytes waiting to be cleared from the queue, as illustrated in Figure 3.

Q3 is in negative credit state. This means it gets no scheduling cycles because the deficit counter is not greater than zero. The result is that turn 2 in the scheduling now jumps to the low-priority queues. Because Q1 has a higher weight than queue 0, the scheduling cycle services Q1, which removes one 1000-byte packet from the queue, leaving its deficit counter at a value of 0. The next queue to be scheduled in turn 2 is Q0, and it removes 800 bytes before running into a negative credit state, as illustrated in Figure 4.

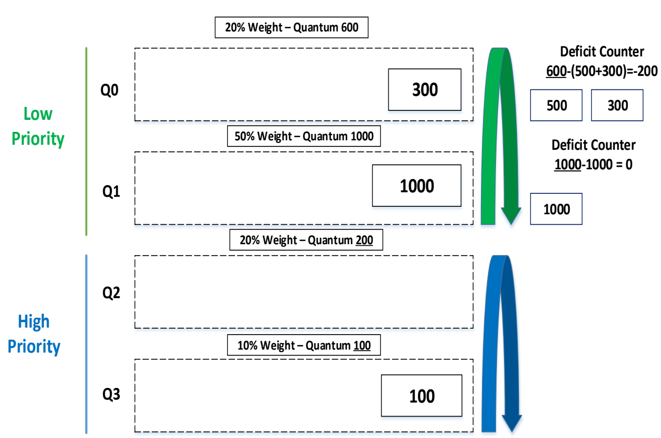

Fig. 4. MDWRR scheduling, turn 2 Q0 and Q1

Now we have a scenario in which one of the priority-high queues, queue 3, and the low-priority queue 0 are in negative credit state. If we honor the state of the queues, in the next scheduling turn, the queue to get scheduling service is queue 1 because it is not in negative credit state. Because queue 2 has no packets in the queue, the two queues in negative credit state can use queue 2′ s weight and thereby update their credit states based on available weights. Because a queue can receive service only once the deficit counter is greater than zero, the queue needs to be bypassed in some scheduling rounds to increase their credits. If no other queues with positive credit state have packets in their queues, this refill of credits happens faster because the punished queues can reuse the available weights not being used by other queues.

V. MDWRR Pseudo Code

The pseudo code in this section does not describe the operation of any specific DWRR implementation. Although each implementation will differ from this model, reviewing the examples and tracing the pseudo code will make it easier to understand the specific design decisions that are required to make for implementations.

The array variable Deficit_Counter is initialized to zero. In this example, the queues are numbered 1 to n, where n is the maximum number of queues on the output port:

FOR i = 1 to n /* Visit each queue index */
Deficit_Counter[i] = 0 /* Initialize Deficit_Counter[i] to 0 */
ENDFOR

The function Enqueue (i) places newly arriving packets into its correct queue and manages what is known as the ActiveList. The ActiveList is maintained to avoid examining empty queues. The ActiveList contains a list of the queue indices that contain at least one packet. Whenever a packet is placed in a previously empty queue, the index for the queue is added to the end of the ActiveList by the function InsertActiveList(i). Similarly, whenever a queue becomes empty, the index for the queue is removed from the ActiveList by the function RemoveFromActiveList(i).

Enqueue(i)
i = the index of the queue that will hold the new packet
IF (ExistsInActiveList(i) = FALSE) THEN /*IF i not in ActiveList */ InsertActiveList(i) /* Add i to the end of ActiveList */ Deficit_Counter[i] = 0 /*Initialize queue Deficit_Counter[i] to 0*/
ENDFOR
Enqueue packet to Queue[i] /* Place packet at end of queue i */
END Enqueue

Whenever an index is at the head of the of the ActiveList, the function Dequeue () transmits up to Deficit_Counter[i] + Quantum[i] worth of bytes from queue. If, at the end of the service round Queue[i] still has packets to send, the function InsertActiveList(i) moves the index i to the end of the ActiveList. However, if Queue[i] is empty at the end of the service round, the Deficit_Counter[i] is set to zero and the function RemoveFromActiveList(i) removes the index i from the ActiveList.

```
Dequeue ()
    While (TRUE) DO
        IF (ActiveList is NotEmpty) THEN
            i = the index at the head of the ActiveList
            Deficit_Counter[i] = Deficit_Counter[i] +
                                    Quantum[i]
            WHILE (Deficit_Counter[i] > 0 AND NOT
              Empty (Queue[i])) DO
              PacketSize = Size (Head (Queue[i]))
              IF (PacketSize <= Deficit_Counter[i]) THEN
                  Transmit packet at head of Queue[i]
                  Deficit_Counter[i] = Deficit_Counter[i] –
                  PacketSize
/* To find out the largest traffic volume */
                /*Declare largest as integer*/
              Set largest to 0
               FOR EACH value in A DO
                   IF A[n] is greater than largest THEN
                     largest ← A[n]
                   ENDIF
               END FOR
                  Dequeue the largest first
              ELSE
                          Break /*exit this while loop*/
              ENDIF
          ENDWHILE
          IF (Empty (Queue[i])) THEN Deficit_Counter[i] = 0
              RemoveFromActiveList(i)
          ELSE
              InsertActiveList(i)
           ENDIF
        ENDIF
    ENDWHILE
END Dequeue
```

Using this pseudo code to implement a DWRR queue scheduling discipline for deployment in a production network has inherent limitations:

(I) Continuing to service multiple packets from a single queue until Deficit_Counter[i] is less than the size of the packet at the head of the Queue[i] can introduce jitter, making it difficult for this implementation to support real-time traffic.

(II) The inability of the model to support a negative Deficit_Counter[i] means that if Queue[i] does not have enough credits to transmit a packet, then the queue may experience bandwidth starvation, because it is not allowed to transmit during the current nor, perhaps, subsequent service rounds until it has accumulated enough credits.

VI. BENEFITS OF MDWRR

Modified weighted round-robin scheduling discipline that addresses the limitations of WRR. Deficit algorithms are able to handle packets of variable size without knowing the mean size. The main benefits are stated below-

(A) Provides protection among different flows, so that a poorly behaved service class in one queue cannot impact the performance provided to other service classes assigned to other queues on the same output port,

(B) Overcomes the limitations of WRR by providing precise controls over the percentage of output port bandwidth allocated to each service class when forwarding variable-length packets,

(C) Overcomes the limitations of strict PQ by ensuring that all service classes have access to at least some configured amount of output port bandwidth to avoid bandwidth starvation; and

(D) Implements a relatively simple and inexpensive algorithm, from a computational perspective, that does not require the maintenance of a significant amount of per-service class state.

VII. LIMITATIONS OF MDWRR

MDWRR is unable to use the total allocated network bandwidth even in burst traffic. The limitations are-

(A) Highly aggregated service classes mean that a misbehaving flow within a service class can impact the performance of other flows within the same service class. However, in the core of a large IP network, it required to schedule aggregate flows, because the large number of individual flows makes it impractical to support per-flow queue scheduling disciplines.

(B) MDWRR does not provide end-to-end delay guarantees as precise as other queue scheduling disciplines do.

(C) MDWRR may not be as accurate as other queue scheduling disciplines. However, overhigh-speed links, the accuracy of bandwidth allocation is not as critical as over low-speed links.

VIII. CONCLUSION

We assume that high volume traffic should get first priority. To achieve both oversubscription and low latency PON is by implementing a DBA algorithm. A DBA algorithm minimizes latency, improves utilization and should respond quickly to the changing traffic patterns. We have modified the Deficit Weighted Round Robin Algorithm by adding a query that will check the largest traffic volume and will dequeue it first. MDWRR provides all of the benefits of WRR, while also addressing the limitations WRR by supporting the accurate allocation of bandwidth when scheduling variable-length packets.

REFERENCES

[1] Tae Il Jeong, Jae Ho Jeong, and Sung Jo Kim, "An Efficient Scheduling Mechanism Using Multiple Thresholds for Multimedia Traffic in ATM Switching Nodes," 22nd IEEE Conf. on Local Computer Networks (LCN'97), Nov. 1997.

[2] Michael P. McGarry, Martin Maier, and Martin Reisslein, "Ethernet PONs: A Survey of Dynamic Bandwidth Allocation (DBA) Algorithms," IEEE Communications Magazine, Vol. 42, No. 8, (August-2004), pp. 1-15.

[3] "Traffic Management Specification Version 4.0," ATM Forum, Apr. 1996.

[4] Ferguson, Paul and Huston, Geoff. Quality of Service: Delivering QoS on the Internet and in Corporate Networks. John Wiley & Sons, January 1998. (ISBN 0471243582).

[5] Shreedhar, M. and Varghese, G. "Efficient Fair Queueing Using Deficit Round Robin." Proc. ACM SIGCOMM '95, Vol. 25, No. 4 (October 1995):231–242.

[6] M. Katevenis et al., "Weighted Round-Robin Cell Multiplexing in a General-Purpose ATM Switch Chip," IEEE J. of Selected Areas in Comm., vol. 9, no. 8, Oct. 1991, pp. 1265-1279.

[7] Demers, A., Keshav, S., and Shenker, S. "Analysis and Simulation of a Fair-queueing Algorithm." Proc. ACM SIGCOMM '89 (September 1989): 1–12.

[8] Partridge, Craig. Gigabit Networking. Addison-Wesley Pub Co., January 1994. (ISBN 0201563339).

[9] V. Hottmar and B. Adamec, "Analytical Model of a Weighted Round Robin Service System," Journal of Electrical and Computer Engineering, vol. 2012, Article ID 374961, 6 pages, 2012.

[10] Chuck Semeria, "Supporting Differentiated Service Classes: Queue Scheduling Disciplines" Juniper Networks, Inc, Part Number: 200020-001 12/01.

[11] Ji-Young Kwak, Ji-Seung Nam, and Doo-Hyun Kim, "A Modified Dynamic Weighted Round Robin Cell Scheduling Algorithm," ETRI Journal, Volume 24, Number 5, October 2002, pp. 360-372.

[12] Queuing and Scheduling (QOS-Enabled Networks) Part 3 [Online]. Available: http://what-when-how.com/qos-enabled-networks/queuing-and-scheduling-qos-enabled-networks-part-3/

[13] Deficit round robin [Online]. Available: https://en.wikipedia.org/wiki/Deficit_round_robin

Mr. Md. Hayder Ali was born in Dhaka, Bangladesh, in 1985. He received the B. Sc and M. Sc in Computer Science & Engineering, M. Sc in Information and Communication Technology, Commonwealth Executive MBA and now a Ph. D researcher in Computer Science & Engineering. He is member of Bangladesh Computer Society (BCS), The Institution of Engineers, Bangladesh (IEB), IEEE and ISOC, IAENG, IAENG Society of Computer Science.

He was a fellow of APRICOT, APAN-45,46,47 and 48, SANOG and bdREN. He has more than 23 National and International training, 19 publications and 6 international conference speakers including SANOG, APRICOT, APAN.

He has about 11 years' experience in IT industries. From 2014 to till date, he is working with National University as an Assistant System Analyst. He is a core member of network planning, operation and management team in National University.

Dr. Mohammad Hanif Ali was born in Dhaka, Bangladesh, in 1956. He received the B. Sc (Hons) and M. Sc in Physics from Jahangirnagar University. He received his PhD degree from UK.

He joined in Bangladesh Atomic Energy Commission in 1982 as a Scientific Officer. He was promoted to senior scientific officer in 1987 and become principal scientific officer in 1995.

Dr. Hanif joined in the Department of Computer Science and Engineering in 1999 as an Associate Professor. He was promoted to Professor in 2003 and working in the same department till date.