Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
() : –
© TÜBİTAK
doi:10.3906/elk-1710-45

A model of QoS differentiation burst assembly with padding for improving the performance of OBS networks

Viet Minh Nhat VO[1,*], **Van Hoa LE**[1], **Hoang Son NGUYEN**[2], **Manh Thanh LE**[2]
[1]Hue University, Hue, Vietnam
[2]Hue University of Sciences, Hue, Vietnam

Received: 0410.2017 • **Accepted/Published Online:** 03.04.2018 • **Final Version:** ..201

Abstract: Burst assembly is an operation at the ingress node of optical burst switching (OBS) networks that aggregates incoming packets from various access networks into larger carriers, called bursts. Depending on the density of incoming packets and the preset time or length thresholds, the completed bursts may have various lengths, but they must be at least equal to a minimum value ($B_{\min}$) to facilitate the switching in existing physical optical switches. If a completed burst is smaller than $B_{\min}$, it should be padded by padded bytes and it results in bandwidth utilization inefficiency. One solution to the problem is increasing the assembly time so that completed bursts must be longer than or at least equal to $B_{\min}$. However, increasing the assembly time will result in increased end-to-end delay. This article proposes a model of QoS differentiation burst assembly with padding for improving the performance of OBS networks.

Key words: Optical burst switching networks, burst assembly, QoS differentiation, burst padding, bandwidth utilization efficiency, delay reduction, throughput fairness

1. Introduction

Optical burst switching (OBS) is considered to be a viable solution of optical packet switching for the next-generation Internet [1], in which the bandwidth of fibers is effectively exploited by WDM technologies [2]. At the ingress node of OBS networks, incoming traffic (e.g., IP packets, ATM cells, or Ethernet frames) from various access networks is aggregated into larger carriers, called bursts, through an operation called burst assembly. A control packet is sent on a dedicated control channel prior to its burst by an offset time. At intermediate nodes, bandwidth is reserved on outgoing links according to the information carried in the control packet ahead of the real arrival of its burst so that the burst always remains in the optical domain until reaching its destination.

Burst assembly is an important operation at ingress nodes of OBS networks. Several burst assembly algorithms have been proposed, which are classified into two categories: timer-based assembly that produces variable-length bursts at periodic time intervals [3] and length-based assembly that sends fixed-length bursts into the core network at nonperiodic time intervals [4]. These algorithms are too rigid for real-time incoming packets to accommodate their assembly. Therefore, some hybrid algorithms based on both time and length thresholds have been proposed [5,6].

Completed bursts are required to be at least equal to a minimum value ($B_{\min}$) to facilitate the switching in existing physical optical switches [7]. For bursts that are smaller than $B_{\min}$, they must be padded by padded bytes. This padding is not effective in terms of bandwidth utilization efficiency. One solution to the problem

[*] Correspondence: vvmnhat@hueuni.edu.vn

is increasing the assembly time so that completed bursts are longer than or at least equal to $B_{\min}$. However, increasing the assembly time will result in increased end-to-end delay. This article, therefore, proposes a model of QoS differentiation burst assembly with padding for improving the performance of OBS networks in terms of bandwidth utilization efficiency, delay reduction, and throughput fairness.

The remainder of this paper is organized as follows: Section 2 analyzes the previous models of QoS differentiation burst assembly, in which the approaches for delay reduction are focused on. Based on their drawbacks, a model of QoS differentiation burst assembly with padding is proposed in Section 3 for improving the performance of OBS networks. Simulation results are compared in Section 4. Section 5 gives the conclusion.

2. Related works

QoS differentiation is an essential requirement for transport networks, including OBS networks. Provision of QoS differentiation could be performed at ingress edge nodes, at core nodes, or at both edge and core nodes [8]. Within this article, edge-based QoS differentiation is addressed.

There are two approaches proposed for edge-based QoS differentiation: offset time-based differentiation (OTD) [9,10] and burst length-based differentiation (BLD) [11,12]. With OTD, an extra offset time is assigned to high-priority (HP) bursts, which results in an earlier resource reservation for them (Figure 1a). Based on the extra offset time, OTD allows isolating HP bursts from low-priority (LP) bursts, but with the condition that the extra offset time has to be as long as at least a few average LP burst lengths to achieve perfect isolation [13]. In BLD, due to short bursts being more likely to fit in the voids generated between scheduled bursts, HP packets are aggregated into short bursts for enhancing the performance of HP packets relative to LP packets in terms of the loss probability (Figure 1b).

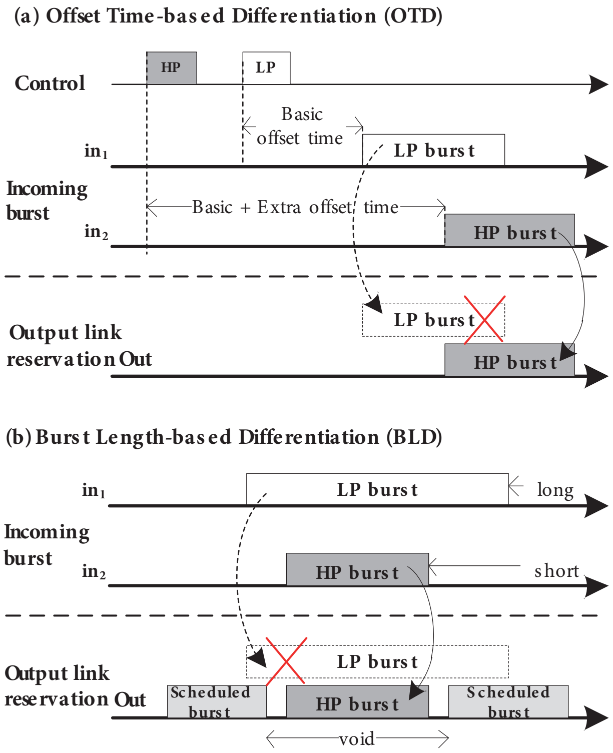


Figure 1. Illustration of (a) offset time-based and (b) burst length-based differentiation.

Regardless of the use of OTD or BLD, a completed burst is only sent after a total delay of assembly time (T_a) and offset time (T_o), generally called buffering delay (Figure 2a). Decreasing the buffering delay results in reduced end-to-end delay [14]; therefore, this article focuses on the approach of offset time-based QoS differentiation burst assembly for delay reduction.

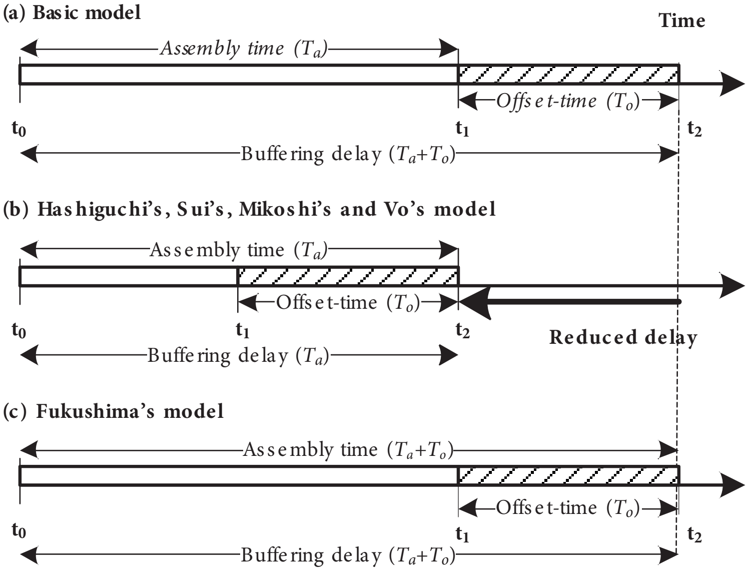


Figure 2. Comparison of the reduced delay among BADR models.

Several models of burst assembly for delay reduction (BADR) have been proposed [15–18], in which the general idea is to include the offset time in the assembly time (Figure 2b). In [19], Fukushima et al. also proposed a BADR model, in which the packets arriving during the offset time are allowed to be aggregated into the current burst. If the assembly time is generalized as the period in which all incoming traffic is aggregated into the same burst, Fukushima et al.'s model is similar to the models in [15–18], but with the larger assembly time of $T_a + T_o$ (Figure 2c).

In order to support QoS differentiation, Sui et al. [18] proposed a model of multiple QoS queues with differential assembly times and offset times, called POQA (Prediction and Offset QoS Assembly), so that the HP queue has a short buffering delay and long offset time, while the LP queue has a long buffering delay and short offset time (Figure 3a). Therefore, HP bursts not only favor early resource reservation but also reduce the buffering delay. However, finding the best values of assembly time and offset time for differential priority queues to achieve perfect isolation is an important issue that needs more study. In the offset time-based QoS differentiation model, Fukushima et al. [19] set differential offset times but kept the same assembly time for priority bursts (Figure 3b). As a result, the higher the priority is, the higher the buffering delay is.

In [20,21], Garg and Kaler proposed a model of edge and core node-based QoS differentiation. Two queues are organized at edge nodes for two priority classes. The control packets are sent as soon as the first packet arrives at the empty queue. Since the burst length should be carried in the control packet, the linear predictive filter (LPF)-based technique is used to estimate the burst length. If the completed length is less than the estimated length, the control packet pretransmission is successful, but if the completed length is greater than the estimated length, another control packet is resent after an offset time. This increases the used extra bandwidth and increases the amount of processed control packets. The QoS differentiation proposed in [20,21] is

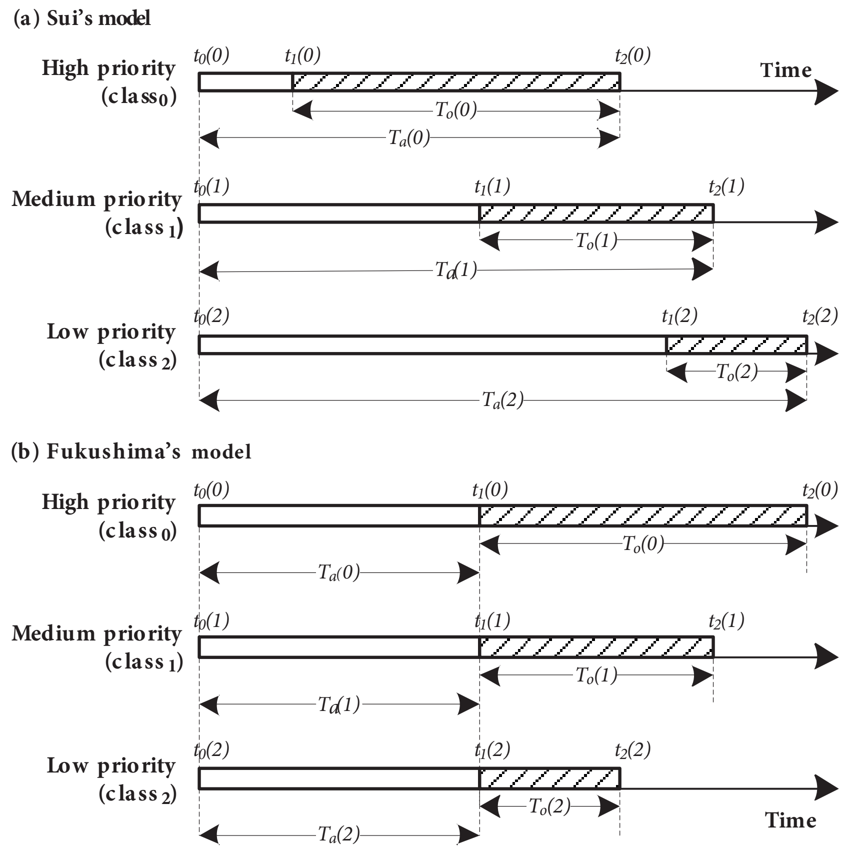


Figure 3. QoS differentiation models of (a) Sui et al. and (b) Fukushima et al.

primarily based on scheduling priorities (at edge and core nodes), where LP bursts are only scheduled after the completed scheduling of HP bursts in a time-window Δt. If the scheduling time for each burst is insignificant, then Δt is the time required to complete the scheduling of HP bursts; Garg and Kaler's BADR model is thus similar to Fukushima et al.'s model in which the offset-time of LP bursts is T_o, while that of HP bursts is $T_o + \Delta t$. Note that $T_o = T_a$ in this case.

Obviously, Sui et al.'s model is better than that of Fukushima et al. and Garg and Kaler in early resource reservation and reduced buffering delay for HP bursts. However, similar to other burst assembly models, Sui et al.'s model also produces short bursts with the low incoming packet density and a preset short time threshold. If the length of these bursts is shorter than a minimum value ($B_{\min}$), they must be padded by padded bytes, which results in bandwidth usage inefficiency. Therefore, it is necessary to have solutions to this problem.

In the next section, a model of offset time-based QoS differentiation burst assembly with padding is proposed to increase bandwidth utilization efficiency, reduce buffering delay, minimize estimation error, and improve the throughput fairness of QoS classes.

3. Model of QoS differentiation burst assembly with padding

3.1. Structure of ingress nodes

The structure of ingress nodes that supports QoS differentiation burst assembly with padding is proposed as in Figure 4. Incoming packets are first classified by their destinations (e.g., egress nodes). The packets of the same destination are then allocated into multiple queues based on their QoS classes. Assuming that incoming packets belong to n QoS classes ($class_i$, $i = 0, 1 \ldots n-1$), n queues (q_i) are thus implemented. The packets in each queue are next aggregated into a burst if its timer or length threshold is reached. The completed burst could be finally padded if its length is shorter than $B_{\min}$.

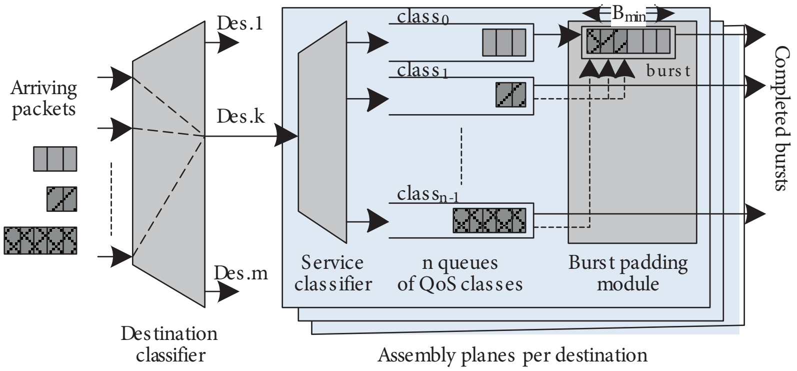


Figure 4. The structure of an ingress node for the model of QoS differentiation burst assembly with padding.

Our proposed model of QoS differentiation burst assembly with padding is an improvement of the OBADR algorithm in [17] and Sui et al.'s model in [18]. OBADR is an algorithm of two-phase burst assembly, in which the assembly based on time threshold (the time to send the control packet) is in Phase 1 and the assembly based on length threshold (the estimated length) is in Phase 2. As proved in [17], using the estimated length as the assembly threshold in Phase 2 minimized the estimation error, but this error always exists due to the error of used estimation methods and the various lengths of incoming packets. In our proposed model of burst assembly with padding, $B_{\min}$ is used as a length-threshold option of OBADR in Phase 2, which significantly helps minimize the estimation error (see Section 4.4).

From Sui et al.'s model, our model adds a constraint for perfect isolation. Consider 2 successive priority classes, $class_i$ and $class_{i+1}$, where $class_0$ is the highest priority class. Perfect isolation between the $class_i$ burst and $class_{i+1}$ burst is reached if:

$$T_o(i) \geq \alpha \times \lambda_{i+1} \times T_a(i+1), \tag{1}$$

where α, $\alpha \in \mathrm{N}^+$ and $\alpha > 1$, is a multiple factor and λ_{i+1} is the rate of packets arriving at queueq_{i+1}.

Note that if the packet length is presented as the time duration of occupied bandwidth (ON), the incoming rate of packets (λ_{i+1}) could be presented as the ratio of ON/(ON + OFF), where OFF is the time duration of idle bandwidth; therefore, $0 \leq \lambda_{i+1} \leq 1$.

To favor the (HP) $class_i$ burst, its assembly time must be shorter than the assembly time of the (LP) queueq_{i+1}, as in the following equation:

$$T_a(i) < T_a(i+1). \tag{2}$$

From Eqs. (1) and (2), the constraint for perfect isolation is inferred as follows:

$$\alpha \times \lambda_{i+1} < \frac{T_o(i)}{T_a(i)}. \tag{3}$$

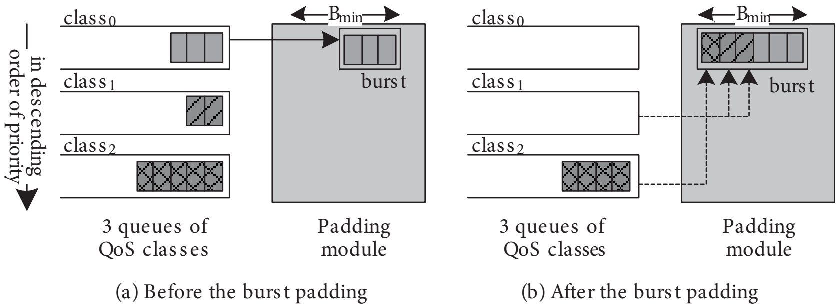
Therefore, the assembly time and the offset time of queue q_i must be set relative to the rate of packets arriving at queue q_{i+1} and the multiple factor α.

3.2. Padding policies

When the rate of $class_i$ packets arriving at queue q_i is high, the length of completed bursts (b_i) is equal to or longer than $B_{\min}$ and nothing is done, but if the rate of incoming packets is low, the assembly time of queue q_i $(T_a(i))$ is reached before, and completed bursts b_i are smaller than $B_{\min}$, there is a need to move the packets from lower priority queues q_j to pad completed burstsb_i, where $j > i$.

Our padding policies are proposed as follows:

1. The packets from lower priority queues are moved to higher priority bursts (Figure 5), because the short assembly time of the HP queue could make the completed burst length shorter than $B_{\min}$. If the padding is successful, the padding packets will have reduced assembly delay in comparison with their original assembly time.



(a) Before the burst padding

(b) After the burst padding

Figure 5. Illustration of the padding policies with 3 classes.

2. The packets chosen for padding from LP queues are in the manner of first come, first served. This means that early incoming packets in LP queues are moved first, which decreases their assembly delay.

3. The padding packets are concatenated to the tail of higher priority bursts (Figure 5), because the drop probability of the tail is higher than that of other parts (head or middle) of the overlapped burst [12,22]. The padding keeps the lower drop probability for HP packets, while it leaves higher drop probability for LP packets.

4. The packets from the queues whose control packets have been sent are not chosen for padding. This is to avoid more added complexity for the burst assembly unit and more wasteful bandwidth by resending another control packet to correct the information of burst length carried in a previous control packet.

With the implementation of the above proposed padding policies, the benefits include: 1) guaranteeing the length of a completed burst that is not shorter than $B_{\min}$; 2) decreasing the assembly delay of the padding packets; 3) reducing the required bandwidth for LP bursts, but not increasing the load of HP bursts; 4) minimizing the estimation error of the burst assembly algorithm; and 5) improving the throughput fairness of QoS classes.

3.3. Algorithm of QoS differentiation burst assembly with padding

Our algorithm of QoS differentiation burst assembly with padding (QDBAP) is proposed as follows.

The complexity of the QDBAP algorithm for one cycle of burst assembly depends mainly on the number of packets (m) arriving in the assembly time and the padding if the completed burst is smaller than $B_{\min}$

Algorithm QDBAP (QoS Differentiation Burst Assembly with Padding**)**

```
Input:    - $T_a(i)$, $T_o(i)$          // assembly time and offset time of queue $q_i$;
Output:   - $B(i)$                      // set of completed bursts.
Begin
    $B(i) \leftarrow \varnothing$; $b(i) \leftarrow \varnothing$;      // $b(i)$ is the burst aggregated of queue $q_i$
    while a packet $p$ arrives at the queue $q_i$ do
        if $b(i) = \varnothing$ then       // queue $q_i$ is empty
            $t(i) \leftarrow s_P$;            // set the timer of queue $q_i$ to the incoming time of $p$ ($s_P$)
            $T_a(i) \leftarrow T_a(i) + s_P$;
            $t_1(i) \leftarrow T_a(i) - T_o(i)$; // determine the time $t_1(i)$ for sending BCP
        end if
        $b(i) \leftarrow b(i)+\{p\}$;         // aggregate $p$ into $b(q_i)$
        if $t(i) \geq t_1(i)$ then         // if timer $t(q_i)$ reaches $t_1(q_i)$, send BCP
            $L(i) \leftarrow |b(i)|$;         // current length of queue $q_i$
            $L_e(i) \leftarrow (1-\alpha)\times L_{avg}(i)+\alpha\times L(i)\times\frac{T_a(i)}{T_a(i)-T_o(i)}$
            if ($L_e(i) < B_{\min}$) then $L_e(i) \leftarrow B_{\min}$   // set the estimation length to $B_{\min}$
            $t_1(i) \leftarrow \infty$;
        end if
        if ($t(i) \geq T_a(i)$) then       // if timer $t(i)$ reaches $T_a(i)$, send burst
            $L(i) \leftarrow |b(i)|$;            // length of completed burst
            if ($L(i) < B_{\min}$) then // if burst length < $B_{\min}$, pad the burst
                $j \leftarrow i+1$;
                while ($L(i) < B_{\min}$) & ($j \leq n$) & ($t(j) < t_1(j)$) do
                    if ($L(j) \geq B_{\min} - L(i)$) then
                        $L(j) \leftarrow L(j) - (B_{\min} - L(i))$;
                        $L(i) \leftarrow B_{\min}$;
                        Reset the timer of queue $q_j$;
                    else
                        $L(i) \leftarrow L(i) + L(j)$;
                        $L(j) \leftarrow 0$;
                        Turn off the timer of queue $q_j$;
                    end if
                    $j \leftarrow j + 1$;
                end while
            end if
            $B(i) \leftarrow B(i) + \{b(i)\}$;      // a new burst is completed
            $b(i) \leftarrow \varnothing$;
            $L_{avg}(i) \leftarrow L(i)$;            // update the average length of completed burst
        end if
    end while
    return $B(i)$
```

(lines 17−−31). In the case that the completed burst is not smaller than $B_{\min}$, no padding is required and the complexity of QDBAP is O(m). However, if the completed burst is smaller than $B_{\min}$, it needs to be padded (lines 19-30) and the complexity of QDBAP is O($m \times n$), where n is the number of priority queues.

4. Simulation and analysis

An ingress OBS node that supports the QoS differentiation burst assembly with padding as shown in Figure 4 is considered. Incoming packets are assumed to have a Poisson distribution, in which their lengths vary in the range of [500, 1000] in bytes and belong to three priority classes, $class_0$, $class_1$, and $class_2$, in descending order of priority. Three queues, q_0, q_1, and q_2, are thus implemented, in which their assembly times are set to $T_a(0) = 0.4$ ms, $T_a(1) = 0.45$ ms, and $T_a(2) = 0.5$ ms, respectively, and their offset times are $T_o(0) = 0.3$ ms, $T_o(1) = 0.25$ ms, and $T_o(2) = 0.2$ ms, respectively.

As discussed in Section 3.1, the rate of packets arriving at queue q_i is referred to as the ratio of occupied bandwidth (ON) to total bandwidth (ON + OFF); thus, $0 \leq \lambda_i \leq 1$ and $i = 0, 1, 2$. To satisfy the constraint in Eq. (3), the multiple factor α is set to 2 and the rates of packets arriving at queues q_0, q_1, and q_2 are set to an arbitrary value, a value less than 0.333, and a value less than 0.25, respectively.

With the same packet load of 0.2 arriving at three queues, Table 1 shows the average length of completed bursts, in which $class_0$ bursts have the smallest length. The $B_{\min}$ value of 30,000 bytes is therefore chosen for our simulations because it is consistent with the suggestion in [4] where the minimum burst length is in the range of [1.25, 30] in Kbytes. Also with this choice, if the rate of $class_0$ packets decreases while the rates of $class_1$ and $class_2$ packets increase, more completed $class_0$ bursts are smaller than $B_{\min}$ and they need to be padded by the packets from queues q_1 and q_2.

Table 1. The average lengths of completed bursts.

Simulation time (s)		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Length burst (bytes)	$class_0$	32,500	32,400	32,500	32,400	32,500	32,600	32,500	32,600	32,500	32,500
	$class_1$	38,600	38,500	38,800	38,500	38,600	38,600	38,500	38,500	38,600	38,600
	$class_2$	42,400	42,600	42,500	42,500	42,500	42,600	42,500	42,600	42,600	42,500

Two scenarios of simulation are considered: Scenario 1 from 0 s to 0.5 s with the same rate (0.2) of packets arriving at three queues and Scenario 2 from 0.6 s to 1.0 s, in which the packet rate is changed to 0.1, 0.25, and 0.25 at queues q_0, q_1, and q_2, respectively. The algorithms of QDBAP and POQA [18] are compared basing on the following criteria:

- Wasteful bandwidth, which is measured by the number of padded bytes.

- Reduced delay rate (R_{rd}) of the padding packets, which is determined by

$$R_{rd} = \frac{\sum_{i=1}^{N} (1 - d_{pad}/d)}{N},$$

where N is the number of padding packets, d is the original assembly delay of padding packet if it is not moved, and d_{pad} is its real assembly delay after padding.

- Average estimation error, which is calculated by

$$R_E = \frac{\sum_{i=1}^{M} \left(\left| L - L_e \right| / L \right)}{M},$$

where M is the number of successive burst assemblies, L is the measured length, and L_e is the estimated length of a burst in one cycle of burst assembly.

- Throughput fairness of QoS classes, which is generally evaluated by throughput fairness index (TFI). Based on the formula of Jain et al. [23], TFI is proposed as

$$TFI = \frac{\left(\sum_{i=1}^{n} \sigma_i y_i \right)^2}{n \sum_{i=1}^{n} \left(\sigma_i y_i \right)^2},$$

where σ_i is the weight factor per class, $0 < \sigma_i < 1$ and $\sum_{i=1}^{n} \sigma_i = 1$, and $y_i = Ld_{class_i} / Bw_{class_i}$ is the ratio of the real burst load (Ld_{class_i}) to the provided bandwidth (Bw_{class_i}).

4.1. Comparisons and evaluations based on wasteful bandwidth

In Scenario 1, with the incoming packet density equal for all three priority classes, the number of padded bytes used for POQA and QDBAP was negligible because no generated bursts were smaller than $B_{\min}$, except for the $class_0$ bursts due to their short assembly time. By padding the packets from queues q_1 and q_2 to $class_0$ bursts, no padded bytes were used in QDBAP (see Figure 6). However, when reducing the incoming $class_0$ packets' density in Scenario 2, all completed $class_0$ bursts were smaller than $B_{\min}$. Many padded bytes should be used in POQA, but very few are needed in QDBAP by moving the packets from queues q_1 and q_2 to pad $class_0$ bursts. However, since the 4th policy (see Section 3.2) only allows this moving to be done if the control packets of queues q_1 and q_2 are not sent, there are still some $class_0$ bursts that are smaller than $B_{\min}$ (see Figure 7), so some padded bytes are still required.

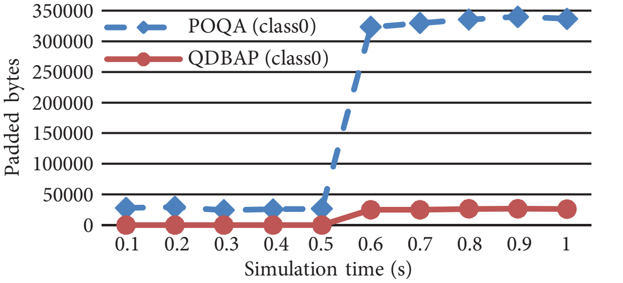


Figure 6. Comparison of the number of padded bytes between POQA and QDBAP.

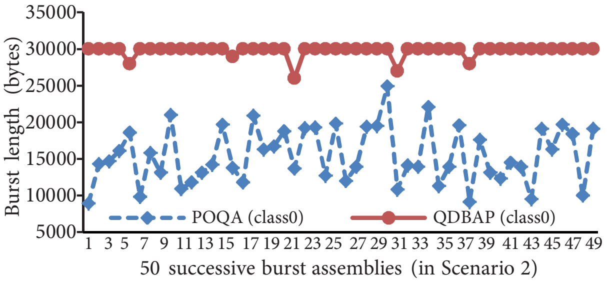


Figure 7. The length of completed $class_0$ bursts with QDBAP and POQA in 50 successive burst assemblies (in Scenario 2).

For $class_1$ and $class_2$, due to the large time threshold and the increased density of packets arriving at queues q_1 and q_2 in Scenario 2, the completed bursts are greater than $B_{\min}$ and no padded byte is used for either POQA or QDBAP.

4.2. Comparisons and evaluations based on reduced delay rate

The padding packets have reduced assembly delay because the assembly time of HP bursts is always shorter than that of LP queues. Specifically, if a packet from queue q_j is moved to a $class_i$ burst $(i < j)$, the reduced delay is at least equal to $T_a(i) - T_o(j)$. Table 2 shows the significantly reduced delay rate of the padding $class_1$ packets, approximately 0.68 in Scenario 1; however, there was a slight reduction in Scenario 2. The reason is that the later the packet arrives, the shorter the waiting time in the queue is, and therefore the reduced delay is less if it is moved for padding. With the padding $class_2$ packets, no delay is reduced in Scenario 1 because no packet is moved for padding, but in Scenario 2, many packets are moved, resulting in a high rate of reduced delay. There is no reduced delay for $class_0$ packets.

Table 2. The reduced delay rate of the padding packets of $class_1$ and $class_2$.

Simulation time (s)		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Reduced delay rate	$class_1$	0.688	0.691	0.688	0.690	0.688	0.550	0.551	0.550	0.551	0.550
	$class_2$	0	0	0	0	0	0.682	0.681	0.682	0.682	0.682

In our simulation, all packets moved to $class_0$ bursts are from queuesq_1 and q_2. As shown in Table 3, the ratio of moved $class_1$ packets is approximately 5% in Scenario 1, but nearly 22% in Scenario 2, while that of moved $class_2$ packets is only about 16%. This is due to our moving policy, which moves sequentially from high to low priority.

Table 3. The moved packet ratio from queuesq_1 and q_2.

Simulation time (s)		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Moved packet ratio(%)	q_1	4.9	5.0	4.8	4.7	5.0	21.7	21.7	21.9	21.8	21.8
	q_2	0	0	0	0	0	15.8	15.8	15.8	15.7	15.9

Padding also contributes to the reduced delay of queuesq_1 and q_2. As shown in Table 4, the reduced delay rate of queue q_1 is less (approximately 0.5) in Scenario 1, while there is no delay reduction in queue q_2, but in Scenario 2, the reduced delay rate of queues q_1 and q_2 are increased approximately 0.22 and 0.1, respectively. This shows the advantage of the proposed padding policies

Table 4. The reduced delay rate of queuesq_1 and q_2.

Simulation time (s)		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Reduced delay rate	q_1	0.049	0.050	0.048	0.047	0.050	0.217	0.217	0.219	0.218	0.218
	q_2	0	0	0	0	0	0.097	0.097	0.097	0.096	0.097

4.3. Comparisons and evaluation based on average estimation error

The estimation error mainly occurs in queueq_0. As shown in Figure 8a, QDBAP has lower estimation error than POQA, especially in Scenario 2. The reason is when the rate of incoming $class_0$ packets is low, a completed $class_0$ burst that cannot reach the length of $B_{\min}$ is recognizable at the time of sending the control packet $(t_1(0))$, so the estimated length of $class_0$ bursts is assigned to $B_{\min}$ (line 15 of the QDBAP algorithm), instead of estimating it, and, as a result, the estimation error is zero. However, as shown in Figure 8a, this value does not reach zero in Scenario 2. The reason is that some completed $class_0$ bursts are smaller than $B_{\min}$ (see Figure 7) and the packets from queues q_1 and q_2 cannot be moved to $class_0$ bursts by the 4th policy (see Section 3.2); this creates a small estimation error for QDBAP in Scenario 2.

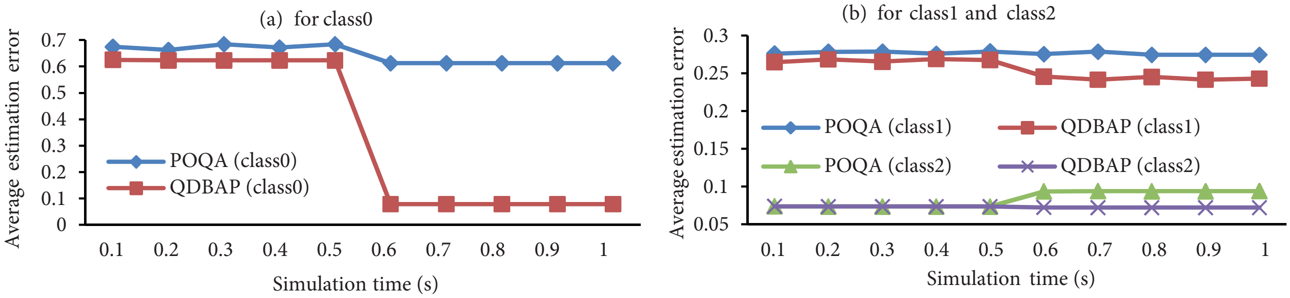


Figure 8. Comparison of the average estimation errors between POQA and QDBAP.

Estimation errors also appear in $class_1$ and $class_2$ (see Figure 8b), but not much. The estimated error of $class_1$ is higher than that of $class_2$ due to the shorter estimation time of $class_1$ [18]. For both $class_1$ and $class_2$, due to the padding, the number of completed $class_1$ and $class_2$ bursts in QDBAP is less than that in POQA; the result is that the estimation error of QDBAP is less than that of POQA.

4.4. Comparisons and evaluation based on throughput fairness

The link (channel) bandwidth utilization capacity in OBS networks never reaches 100% because voids always exist between scheduled bursts. A coefficient $\alpha = 0.7$ is thus added to the rate of maximum usable bandwidth per link [24]. The coefficient is also proved through our simulation (Table 5), in which the maximum achieved throughput per link is 0.72 on average. Therefore, $\alpha = 0.7$ is chosen in our simulations.

Table 5. The maximum achieved throughput per link with various incoming loads.

Incoming load/bandwidth	0.5	0.6	0.7	0.8	0.9	1.0
Maximum achieved throughput	0.48616	0.572186	0.67152	0.72123	0.7213	0.71945

Assuming that the bandwidth provided to the three QoS classes is equal, $Bw_{class_i} = 0.2333$. In Scenario 1, the incoming packet load per priority class is 0.2; thus, the load of all bursts completed by POQA is also 0.2. As a result, $y_0 = y_1 = y_2$ (Figure 9a) and $TFI = 1$. With QDBAP, due to some $class_0$ bursts whose lengths are shorter than $B_{\min}$, a little padding is required, which results in $y_0 \approx y_1 \approx y_2$ (see Figure 9b) and $TFI \approx 1$.

In Scenario 2, the load of incoming packets $class_0$, $class_1$, and $class_2$ is varied to 0.1, 0.25, and 0.25, respectively; the values of y_0, y_1, and y_2 are thus varied by POQA, which causes a big difference among y_i values (Figure 9a). The flow of $class_0$ bursts is under its provided bandwidth, but the flows of $class_1$ and $class_2$ bursts are over their provided bandwidth. This results in the TFI value being reduced to 0.89 on average. However, with QDBAP, by moving the packets from queues q_1 and q_2 to $class_0$ bursts, the real load of $class_1$ and $class_2$ bursts is reduced, while the load of $class_0$ bursts increases. This helps to balance the values of y_0, y_1, and y_2 (Figure 9b), which results in a better value of TFI (0.99 on average).

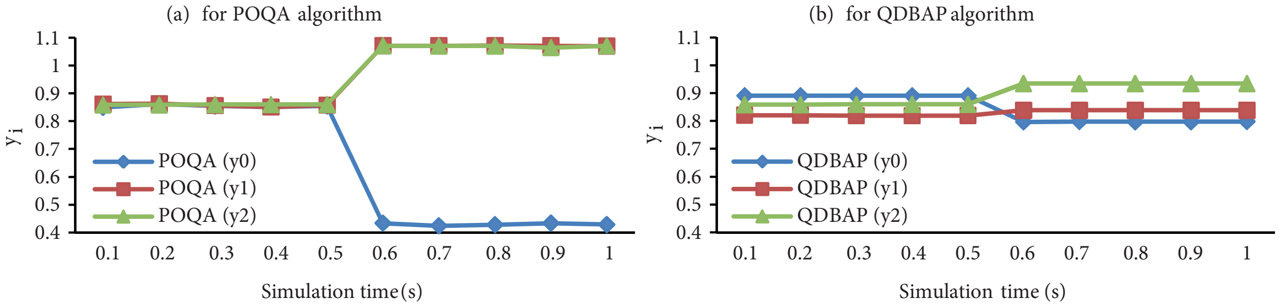


Figure 9. Comparison of the throughput fairness (based on $y_i = Ld_{class_i}/Bw_{class_i}$) between POQA and QDBAP.

Another positive effect of the padding of the QDBAP algorithm is that it reduces the real packet rate at low priority queues. Based on Eq. (3), this reduction further confirms the condition of perfect isolation between successive priority classes in our model.

5. Conclusion

This article has proposed a model of burst assembly with padding, which supports QoS differentiation at ingress OBS nodes. This is a model of BADR integrated with padding policies for avoiding the use of padded bytes. The proposed padding policies have the advantages of increasing the bandwidth utilization efficiency, reducing the buffering delay, minimizing the estimation error, and improving the throughput fairness of QoS classes. Through simulation results, the QDBAP algorithm presented its advantages in comparison with the POQA algorithm. However, QDBAP is only effective if the load of HP traffic is low and the load of LP traffic is high. In the opposite case, padding is no longer required for the operation of burst assembly.

References

[1] Maier M, Reisslein M. Trends in optical switching techniques: a short survey. IEEE Netw 2008; 22: 42-47.

[2] Kiyoshi F, Ichiro H, Toshiharu ITO, Kazuhiko K. Research and development of high-capacity optical link technologies. J Natl Inst Inf Commun Technol 2006; 53: 37-44.

[3] Ge A, Callegati F, Tamil LS. On optical burst switching and self-similar traffic. IEEE Commun Lett 2000; 4: 98-100.

[4] Cao X, Li J, Chen Y, Qiao C. Assembling TCP/IP packets in optical burst switched networks. In: IEEE 2002 Global Telecommunications Conference; 26 March 2003; Taipei, Taiwan. New York, NY, USA: IEEE. pp. 2808-2812.

[5] Yu X, Chen Y, Qiao C. A study of traffic statistics of assembled burst traffic in optical burst switched networks. Proc SPIE 2002; 4874: 149-159.

[6] Kitayama K, Koga M, Morikawa H, Hara S, Kawai M. Optical burst switching network testbed in Japan. In: OFC/NFOEC 2005 Optical Fiber Communication Conference; 6 September 2005; Anaheim, CA, USA. pp. 3-5.

[7] Al Amin A, Nishimura K, Shimizu K, Takenaka M, Tanemura T, Onaka H, Hatta T, Kasukawa A, Tsuji S, Kondo Y et al. Development of an optical-burst switching node testbed and demonstration of multibit rate optical burst forwarding. J Light Technol 2009; 27: 3466-3475.

[8] Akar N, Karasan E, Vlachos KG, Varvarigos EA, Careglio D, Klinkowski M, Josep Solé-Pareta. A survey of quality of service differentiation mechanisms for optical burst switching networks. Opt Switch Netw 2010; 7: 1-11.

[9] Yoo M, Qiao C, Dixit S. Optical burst switching for service differentiation in the next-generation optical internet. IEEE Commun Mag 2010; 39: 98-104.

[10] Dolzer K, Gauger C. On burst assembly in optical burst switching networks - A performance evaluation of just-enough-time. Teletraffic Science and Engineering 2001; 4: 149-160.

[11] Barakat N, Sargent EH. On optimal ingress treatment of delay-sensitive traffic in multi-class OBS systems. In: Proceedings of the 3rd International Workshop on Optical Burst Switching; October 2004; San Jose, CA, USA.

[12] Vokkarane VM, Haridoss K, Jue JP. Threshold-based burst assembly policies for QoS support in optical burst-switched networks. Proc SPIE/IEEE 2002; 4874: 125-136.

[13] Kim S, Choi J, Yoon B, Kang M. Optimal offset-time decision for QoS in optical burst switching networks. J Commun Networks 2007; 9: 312-318.

[14] Hu G, Kohn M. Evaluation of packet delay in OBS edge nodes. In: IEEE 2006 International Conference on Transparent Optical Networks; 20 November 2006; Nottingham, UK. New York, NY, USA: IEEE. pp. 66-69.

[15] Hashiguchi T, Wang X, Morikawa H, Aoyama T. Burst assembly mechanism with delay reduction for OBS networks., In: OSA 2003 Conference on the Optical Internet; 13–16 July 2003; Melbourne, Australia. pp. 664-666.

[16] Mikoshi T, Takenaka T. Improvement of burst transmission delay using offset time for burst assembly in optical burst switching. In: IEEE 2008 Asia-Pacific Symposium on Information and Telecommunication Technologies; 22–24 April 2008; Bandos Island, Maldives. New York, NY, USA: IEEE. pp. 13-18.

[17] Vo VMN, Le VH, Nguyen HS. A model of optimal burst assembly for delay reduction at ingress OBS nodes. Turk J Elec Eng & Comp Sci 2017; 25: 3970-3982.

18 Sui Z, Zeng Q, Xiao S. An offset differential assembly method at the edge of OBS network. Proc SPIE 2005; 6021: 1-6.

[18] Fukushima U, Chen W, Fujiwara Y, Yokohira T. A burst assembly method to reduce end-to-end delay in optical burst switching networks. WSEAS Transactions on Communications 2009; 8: 894-903.

[19] Garg AK, Kaler RS. Effective resource utilization and QoS oriented integrated scheme for optical burst switching (OBS). J Opt Commun 2010, 31: 218-223.

[20] Garg AK, Kaler RS. An efficient quality of service (QoS) scheme for all optical networks. Opt Fiber Technol 2011, 17: 64-69.

[21] Sarwar S, Zeeshan F, Aslam L, Yousaf MM, Qounain WU. Novel composite burst assembly for OBS-networks. Sindh University Research Journal (Science Series) 2017; 49: 773-778.

[22] Jain R, Chiu DM, Hawe WR. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System. DEC Technical Report TR301. Maynard, MA, USA: Digital Equipment Corporation, 1984.

[23] Orawiwattanakul T, Ji Y, Zhang Y, Li J. Fair bandwidth allocation with distance fairness provisioning in OBS networks. J Light Technol 2010; 27: 3370-3380.