Turkish Journal of Electrical Engineering & Computer Sciences

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

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 3133 – 3149
© TÜBİTAK
doi:10.3906/elk-2104-161

A model of service differentiation burst assembling and padding for improving transmission efficiency in OBS networks

Van Hoa LE[1,*], **Hong Quoc NGUYEN**[2], **Thanh Chuong DANG**[3], **Viet Minh Nhat VO**[1]
[1]Hue University, Hue City, Vietnam.
[2]University of Education, Hue University, Hue City, Vietnam.
[3]University of Sciences, Hue University, Hue City, Vietnam.

Received: 24.04.2021 • **Accepted/Published Online:** 10.09.2021 • **Final Version:** 30.11.2021

Abstract: Service differentiation is an indispensable requirement for transmission in optical burst switching (OBS) networks, which can be based on offset-time, burst-length, or both, offset-time and burst-length. The offset time based approach sets a large offset time for high priority bursts and a small offset time for low priority bursts. Whereas, with burst length based approach, high priority bursts are short in size and low priority bursts are long in length. A combination of these two approaches promises to provide flexible service differentiation. The paper proposes a model of service differentiation burst assembling and padding, in which the assembly time threshold is set to reduce the end-to-end delay, but does not effect the burst length prediction accuracy; the length of generated bursts are flexibly adjusted based on feedbacked void size; and the high-priority burst with short length is padded by the lower priority data. Simulation results and analysis show that the proposed model is more efficient than previous similar models in terms of bandwidth utilization, byte loss, throughput fairness and estimation error.

Key words: OBS network, adaptive assembly, adjusted burst lenght, feedbacked void size, controlled padding

1. Introduction

The explosion in the number of users and bandwidth-consuming applications on Internet has caused many challenges for data transmission models over backbone networks. Recent advances in wavelength division multiplexing (WDM) technology have allowed each fiber to reach terabytes of bandwidth. Transmission models through optical fibers have practically attracted a lot of research and development. Optical channel switching (OCS) is an example of this transmission model that has been widely deployed in practice. However, the OCS model also reveals limitations such as low efficiency in bandwidth usage, lack of flexibility in sharing resources and poor adaptation to changes in traffic. Switching models with finer granularity, such as packet switching, are therefore the next trend of optical switching. Optical packet switching (OPS), which is inspired by electronic packet switching, is theoretically ideal, but the required optical technologies, such as optical buffering and optical packet switching, are still immature to it soon happened. An alternative approach called optical burst switching (OBS), which can be seen as a mix of OCS and OPS, has been proposed [1][2]. This is a high-speed data switching technique of the future and is receiving a lot of attention from both optical academia and industry [3].

An important feature of the OBS network is that the burst control packet (BCP) is sent ahead on a

*Correspondence: levanhoa@hueuni.edu.vn

dedicated control channel to reserve resources; after an offset-time, the corresponding burst is followed on one of available data channels (Figure 1). Because the resource is reserved by BCP, the burst does not suffer any delay at intermediate nodes, so there is no need to require optical buffers. On the other hand, since the burst length is quite large compared to the carried packets, using switches at microsecond speed is not reduce bandwidth utilization efficiency. However, this way of transmission also puts a great pressure on how BCP can reserve resources in time at intermediate nodes, ensuring smooth switching the following burst.

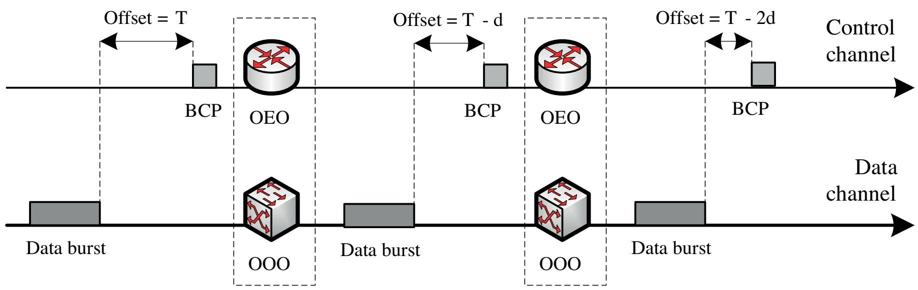

Figure 1. Offset time is gradually reduced as it passes each hop.

A key problem in OBS networks is the burst contention, where a burst arriving at an output port may not find available resources due to being overlapped with other scheduled bursts, while the output channels remain free bandwidth. The cause of the overlap can be due to the improper setting of offset times, the lack of burst transmission synchronization, or the excessive length of generated bursts. Offset time is often determined based on the burst route, while synchronous transmission is difficult to achieve in backbone networks. Hence, burst length adjustment is the most feasible and can be done through burst assembly.

Burst assembly is the operation of aggregating data at ingress nodes, whose techniques include timer-based assembly [4], burst length-based assembly [5]; and both timer- and burst length-based assembly [6]. The length of completed bursts depends on the incoming traffic density and the timer/the length thresholds which are set at assembly queues. If the completed burst is too long, it has little chance of being scheduled into voids, which are idle bandwidth intervals between scheduled bursts, and more data loss for each dropped burst. However, information about void size is usually only available at core nodes, which it should be sent back to the ingress node to adjust the length of aggregated bursts.

In the other side, the burst length needs to be equal to or longer than a minimum threshold B_{min} so that the burst could be switched with existing optical switches [7, 8]. If a burst is smaller than B_{min}, it will be padded with padding bytes; but this is inefficient in terms of fiber bandwidth utilization. An improved approach has been proposed in [9] have, in which data from the low priority queue, instead of padding bytes, is padded the high priority burst; this not only makes more efficient bandwidth utilization, but also gives low priority data the privileges of high priority bursts.

This paper proposes a model of service differentiation burst assembling and padding, in which the offset time is used as a service differentiation label and to reduce contention between bursts that share the same wavelength channel; the assembly time threshold is set to reduce the end-to-end delay, but does not effect the burst length prediction accuracy; the length of completed bursts are flexibly adjusted based on feedbacked void sizes to increase their scheduling probability; and the high-priority burst with short length is padded by the lower priority data.

The main contributions of the paper include:

- Proposing a method of setting the service differentiation time threshold, in which the offset time is nested in the assembly time and the extra offset time of the high-priority burst is greater than the length of the low-priority burst in order to reduced contention between bursts when sharing the same wavelength channel;
- Proposing a flexible burst length threshold adjustment scheme based on feedbacked void size, where an improved structure of NACK and a void size feedback process from the core node to the ingress node is defined;
- Proposing an adaptive assembly technique, that is simultaneously based on a time threshold and a length threshold, where the threshold length is flexibly adjusted according to the void size value feedbacked from core nodes;
- Proposing a burst padding technique, in which the target not only ensures that the completed burst is greater in length than a minimum threshold, but also fits the generated voids. The padding is done by getting data from the low-priority queue to pad the higher priority burst. This method both ensures no padding bytes are used, which increases fiber bandwidth utilization efficiency, and also provides the opportunity for low-priority data to enjoy the privileges of higher priority bursts.

The following sections of the paper are organized as follows: Section 2 presents analysis of related works; based on the analysis results, Section 3 proposes model of service differentiation burst assembling and padding in OBS networks; an simulation-based comparison and analysis are performed in Section 4 and, finally, the conclusion is in Section 5.

2. Related works

Service differentiation is an indispensable requirement for transmission in OBS networks. The packets coming from access networks, depending on their quality of service (QoS) requirements, are transported with various privileges over OBS networks. Specifically, the packets with the same destination and the same QoS requirement are aggregated in one burst and labeled with a ”priority label”. In OBS networks, the most commonly used ”priority label” is offset-time. Therefore, the authors in [10, 11] proposed an offset time-based service differentiation scheme (shown in Figure 2), where an extra offset-time is added for the high-priority burst, while keeping the basic offset-time for the low-priority burst. This offset time service differentiation schema was standardized in the JET protocol [12].

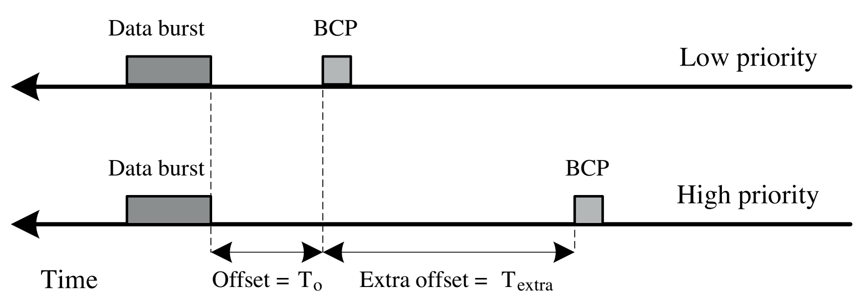

Figure 2. Offset time-based service differentiation scheme.

However, the main disadvantage of the offset time-based service differentiation scheme is that it increases the delay of high-priority bursts. As the recommendation in [10], the extra offset time of the high-priority burst needs three to five times the offset-time of the low-priority burst to achieve a complete isolation (e.g., in byte loss rate) between two consecutive priority classes. This constraint obviously significantly increases the delay of the high-priority burst.

To reduce the burst delay, the authors in [13] have proposed the Prediction and Offset QoS Assembly (POQA) scheme, where the offset-time is nested in the assembly time. As shown in Figure 3, with the traditional assembly model (Figure 3a), the control packet is sent only when the assembly threshold is reached and then the completed burst is sent after an offset time. But with the POQA scheme (Figure 3b), the offset-time is included in the assembly time; the control packet are thus sent before the corresponding burst is completed. However, since the burst length information needs to be carried in the control packet to reserve resources at core nodes, it is necessary to estimate the completed burst length. POQA therefore used the adaptive auto-regressive method to estimate burst length at the time of sending BCP, which is based on the packet arrival rate in the past and in the estimation time period. However, the estimation always suffers from certain error, which has an impact on the performance of POQA.

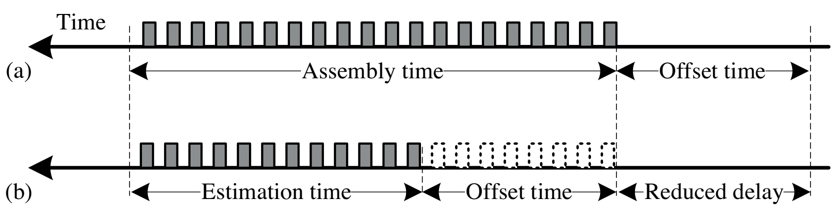

Figure 3. Solution of nesting the offset time into the assembly time to reduce transmission delay.

When implementing the POQA scheme for queues with different priority classes (Figure 4), the higher the priority queue, the greater the offset time. With the time of sending the control packets at $t_1 = T_a(i) - T_o(i))$ (where $T_a(i)$ and $T_o(i)$ are assembly time and offset-time of class i, respectively), the higher priority burst is sent the sooner. Moreover, each priority class has a maximum threshold of assembly time $T_a(i)$, $i = 0..n-1$, so the number of priority classes (n) will also be limited to ensure that a complete QoS separation.

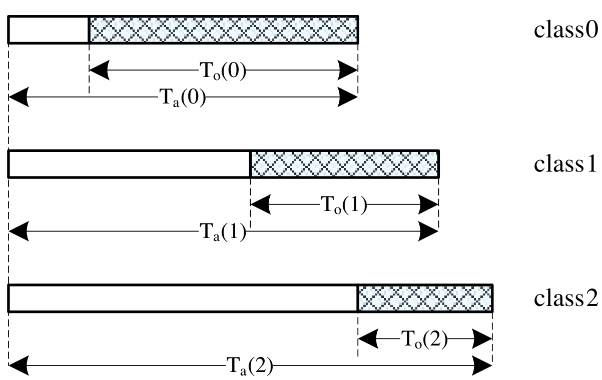

Figure 4. An example of the different assembly times and offset times for service differentiation.

The completed burst length must be equal to or greater than a minimum threshold B_{min} in order to be handled by existing optical switches. In cases where the length of the completed burst is shorter B_{min}, the burst must be padded by padding bytes. This approach is obviously inefficient in terms of fiber's bandwidth utilization. To solve this problem, the author in [9] has proposed a model that combines burst assembly with padding, named the QoS differentiation burst assembly with padding (QDBAP) scheme, which increases the bandwidth utilization efficiency for the high QoS burst, while reducing the delay of low QoS data (since they are sent earlier). An example of this burst padding method is shown in Figure 5, in which burst padding is performed by taking packets from the low QoS queues to pad for the higher QoS burst. If the packet density of a data flow i ($i = 0..n-1$) arriving at the ingress node is high, the completed burst length is longer B_{min} and thus no further processing is required. However, if the arriving packet density of flow i is low, the time threshold $T_a(i)$ quickly reaches but the completed burst length (B_i) is smaller the minimum threshold ($B_i < B_{min}$); it is necessary to get data from queue j to pad burst i, $j > i$ (from the low QoS queue with to the higher QoS burst) to avoid using padding bytes. This approach is not only efficient in terms of fiber's bandwidth utilization, but also reduces the delay of padded data and also their end-to-end delay.

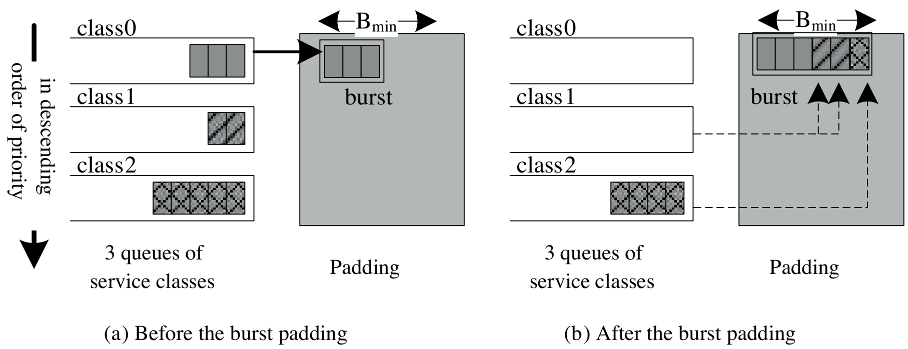

Figure 5. An example of the burst padding method with 3 classes: (a) before padding and (b) after padding.

The assembly that produces large bursts reduces the number of bursts (and also BCPs) circulating in the network; but the amount of data loss per dropped burst is increased. Another negative effect of large bursts is the low probability of scheduling with void-filling because these bursts is larger than voids. Therefore, it is necessary to adjust the burst length to fit at least one of the voids. To do this, it is required to feedback the void size information from core nodes to the edge node for adjusting the burst length.

The next section presents a model of service differentiation burst assembly and padding, in which four functional modules are integrated to address the above problems in order to improve the transmission efficiency in OBS networks.

3. Model of service differentiation burst assembly and padding

The model of service differentiation burst assembly and padding, abbreviated as SDBAP, operates on a combination of functional modules added at the input node and the core node, as shown in Figure 6. Specifically, at the ingress node, incoming data is fed into assembly queues based on their destination and QoS requirements. The burst assembly is deployed at each queue based on a hybrid technique with a pair of preset time threshold and length threshold. The method of setting the service differentiation time threshold of each priority queue is described in subsection 3.1, while the length threshold flexible adjustment approach is analyzed in detail in subsection 3.2. The assembly queues are controlled by two functional modules: the module of service differentiation time setting and the module of length threshold adjustment. In addition, to ensure the completed burst is longer than a minimum threshold (B_{min}), another functional module, called the padding control module, is added to padding packets from low-priority queues to higher priority burst. These three function modules operate at the ingress node (Figure 6).

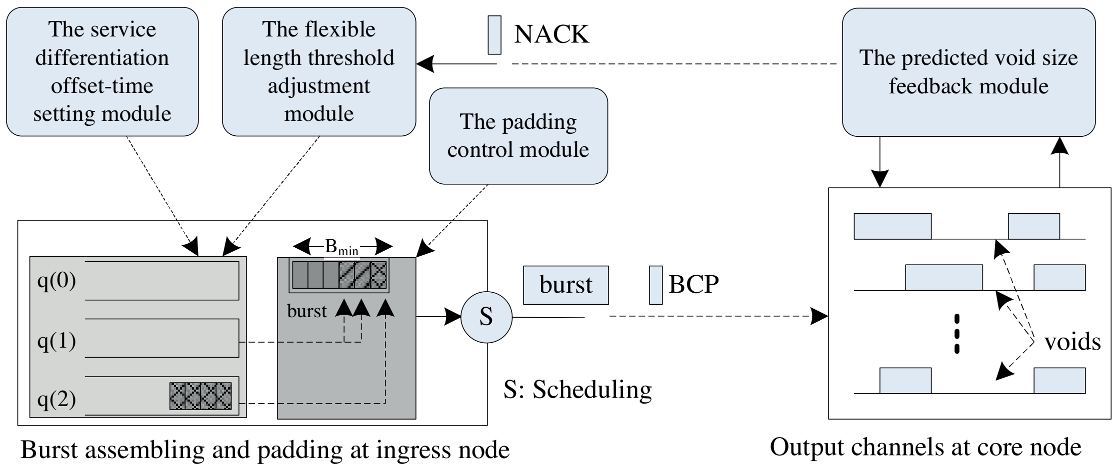

Figure 6. The model of service differentiation burst assembly and padding.

To feedback the void size information from core nodes to the ingress node, another function module, called the predicted void size feedbacking module, is responsible for collecting the size of generated voids and sending the predicted size of the next voids to the ingress node. The operation of these functional modules are shown in the following subsections.

3.1. Service differentiation time setting module

Service differentiation in SDBAP is based on offset time, where the low-priority burst maintains the basic offset time, while the high-priority burst is added an extra offset time. As a result, the high-priority burst suffers from an extra delay. In order to reduce this delay, the method of nesting the offset time into the assembly time, which means sending control packets early, as in [9, 13] is chosen for implementation in SDBAP. The estimated length is then determined based on the completed burst length (L_j) of $M-1$ previous assembly times and the current length of assembly queue (L_M) by Equation 1.

$$L_e = \sum_{j-1}^{M-1} w_j L_j + w_M L_M \frac{T_a}{T_a - T_o}, \tag{1}$$

where w_j is the weight of assembly time j and $\sum w_j = 1$. where $i = [0, 1, 2]$ and $i = 0$ corresponds to the highest priority class.

The extra offset time also needs to be large enough to achieve a complete isolation (e.g., in byte loss rate). SDBAP uses the approach that sets the extra offset time of the high-priority burst greater than the length of the low-priority burst (as shown in Figure 7) to avoid a contention between the priority bursts which share the same wavelength channel [9]; however the extra offset time is also small enough not to increase the delay of the high-priority burst. In SDBAP, the extra offset time set is equal to the maximum possible length of low-priority bursts. With the combination of nesting the offset time into the assembly time and the extra

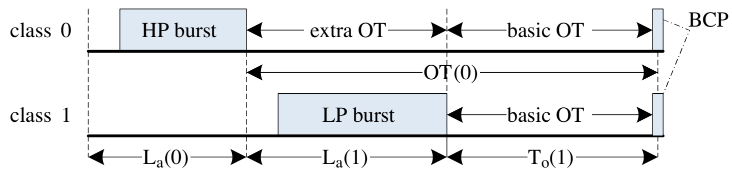

Figure 7. The offset time of the high-priority burst must be greater than the sum of the offset time and the length of the low-priority burst.

offset time of the high-priority burst be greater than the length of the low-priority burst, the assembly time threshold is determined by Equation 2.

$$T_a(i) = T_e(i) + T_o(i) = T_o(i+1) + L(i+1) \tag{2}$$

3.2. Length threshold flexible adjustment module

The assembly technique used in the SDBAP model is based on both time and length thresholds. Setting these threshold values will affect the completed burst length. Specifically, if packet arrival density to the ingress node is high and the time threshold is large, the completed burst has a great length and a low probability of being scheduled to voids at core nodes; but if incoming data is low and the time threshold is small, the completed burst is small and their length may not reach B_{min}. Hence, the assembly thresholds should be flexibly adjusted. In SDBAP, since the offset time is nested in the assembly time, the extra offset time of the high-priority burst should be greater than the length of the low-priority burst and the estimation time should not be too small to reduce the estimation accuracy. Adjusting of the completed burst length can be accomplished through turning of the length threshold.

The high-priority burst needs to be small in length to have a high probability of scheduling with void filling at core nodes,but should be greater than B_{min}. Furthermore, in order to effectively exploit the bandwidth when scheduling with void filling, bursts need to fit in voids. In other words, setting of the threshold length should be based on the void size and therefore there is a need to feedback the void size from core nodes to the ingress node [14]. The void size feedback technique is described in the following subsections.

3.3. Predicted void size feedback module

The predicted void size feedbacking is performed by core nodes, in which the predicted void size feedbacking module collects the size of generated voids to predict the size of the next voids. The predicted void size information is then sent back to the related ingress node every time a burst is dropped at the core node. The NACK packet is used to carry this value. Specifically, in the original NACK packet, the CTRL PDU field remains 6 idle bytes [15], so 4 of 6 bytes it can be used to carry the predicted void size value. This is reasonable since only 4 bytes are used to carry the burst length value. The modified structure of NACK packet is shown in Figure 8.

The predicted void size is calculated based on the history of the size of previously generated voids. Specifically, the predicted value at time $t+1$ ($v_t + 1$) is calculated based on N previous observations.

$$v_{t+1} = w_t v_t + w_{t-1} v_{t-1} + ... + w_{1-N+1} v_{1-N+1}, \tag{3}$$

where v_t is the void size at time t and w_t is its weight, $\sum w_t = 1$. The size values of these consecutive voids can be considered as a form of time series, so the moving average method can be applied. In fact, the

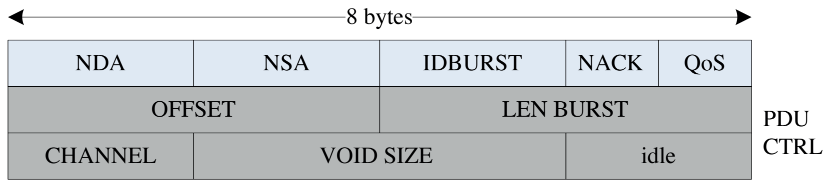

Figure 8. The modified structure of NACK packet with 4 bytes dedicated to carry the predicted void size.

closer the observations contain more information for the impending values, so they need a greater weight than the past observations. The method of exponentially weighted moving averages (EWMA) method is then used and Equation 3 is rewritten to Equation 4.

$$v_{t+1} = wv_t + w(1-w)v_{t-1} + w(1-w)^2 v_{t-2} + ... \tag{4}$$

where $w = 1/N$.

At the ingress node, every time it gets NACK packet, the length threshold flexible adjustment module extracts the predicted void size value to recalculate a new length threshold for the next assembly.

3.4. Padding control module

The padding control module is called when the length of a completed burst has not reached B_{min}. The principle of burst padding is to get packets from the low-priority queue to pad the high-priority burst. The padding position is at the beginning of the burst (see Figure 5), because this position usually has a higher dropping probability compared to other positions, if a contention occurs [16, 17]. Another improvement of this module is that padding continues after the threshold B_{min} has been reached until either the length threshold is reached or the low priority queue is empty. This way of padding maximizes the amount of low-priority data enjoying the privileges of the high-priority burst. However, compared to QDBAP, the padding operations in SDBAP are more complex, thus consuming more computation time.

In summary, the SDBAP algorithm, in the case of the ingress node having three priority queues, is described in detail as follows (Figure 9):

1. Set the time threshold $T_a(i)$, where $i = [0, 1, 2]$, as Equation 2;
2. If receiving the feedbacked void size value $v(i)$, set the length threshold for the corresponding queue: $L_a(i) = v(i)$; if not, initializing the default value: $L_a(0) = B_{min}$, $L_a(i+1) = 1.5 \times L_a(i)$;
3. If $timer(i) = T_a(i)T_o(i)$, estimate the burst length $L_e(i)$ based on Equation 1 and send a BCP; reset the new length threshold: $L_a(i) = L_e(i)$;
4. If the length threshold $L_a(i)$ is reached, the data from $q(i)$ is aggregated into the corresponding priority a burst $b(i)$ and sent it;
5. If $timer(i) = T_o(i)$, data from $q(i+1)$ is padded to the burst $b(i)$ until the threshold $L_a(i)$ is reached ($L(i) \geq L_a(i)$); the completed burst is the sent;
6. If $q(i+1)$ is empty and $L(i) < L_a(i)$, go to Step 5 to continue padding with the next queue: $i = i + 1$.

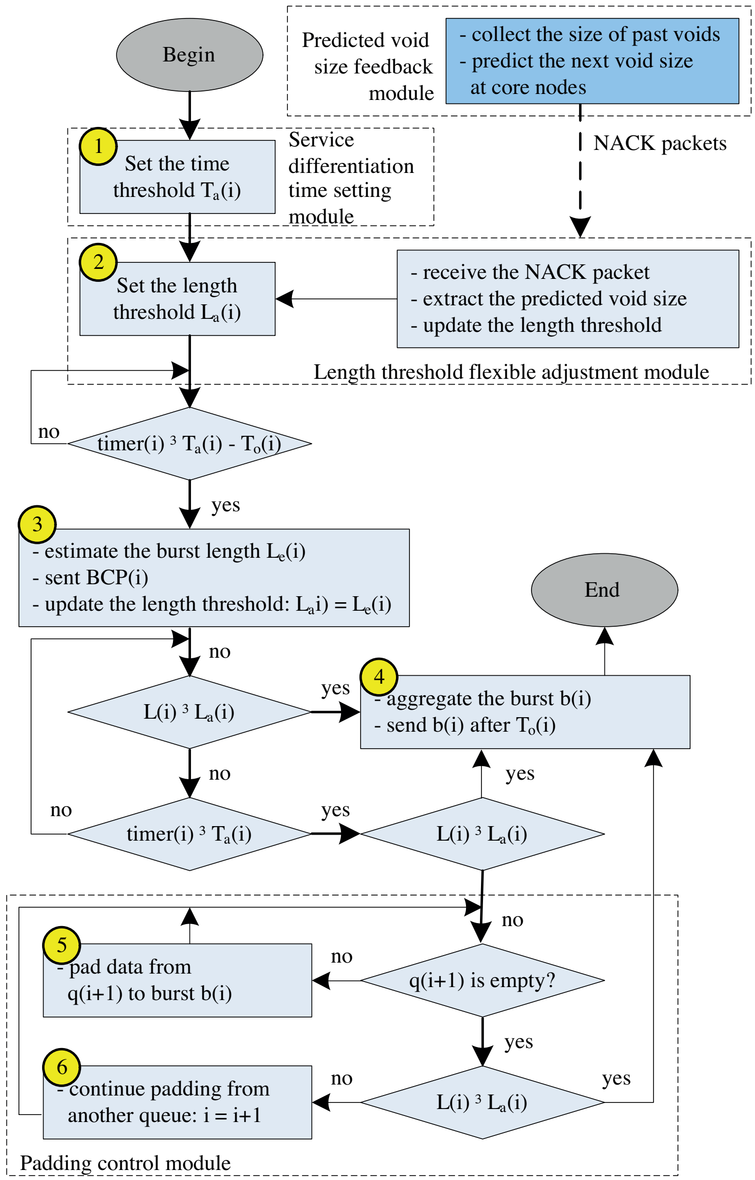

Figure 9. The algorithm of service differentiation burst assembly and padding.

Based on the above additional modules, the SDBAP model increases the efficiency of data transmission over the OBS network. However, SDBAP suffers from increased computational complexity and resource (memory) cost to be operable. Specifically, SDBAP (and QDBAP as well) suffers from the additional complexity due to the padding operation, which is $O(p \times q)$, where p is the number of packets in queue $i+1$ that are taken to pad burst i and q is the number of queues $(i = 1..q)$. In fact, the number of queues is small ($q = 3$ in the above algorithm), so the remaining complexity is $O(p)$. In addition to burst padding, SDBAP collects the size of past and current voids to estimate the size of next voids, thus requiring an additional memory of $N \times VOIDSIZE$ bytes for buffering, where N is the observation window size and $VOIDSIZE = 4$ (see Figure 8). The complexity of the void size estimation algorithm is $O(N)$. However, these two operations perform at two different nodes: padding is performed at the edge node, while void size estimation is performed at the core node. Therefore, the complexity of SDBAP is $max(O(p), O(N))$.

Note that since all three models of POQA, QDBAP and SDBAP perform burst length estimation due to early sending of the control packet, they all suffer the complexity of $O(M)$ for this operation, where M is the number of previous assembly times (see subsection 3.1). The complexity of POQA, QDBAP and SDBAP is therefore $O(M)$, $max(O(M), O(p))$ and $max(O(M), O(p), O(N))$, respectively.

A comparison between SDBAP versus POQA and QDBAP is shown in Table 1.

Table 1. Comparison between SDBAP vs. POQA and QDBAP.

	POQA	**QDBAP**	**SDBAP**
Offset time adjustment	yes	yes	yes
Offset time nested in assembly time	yes	yes	yes
Burst length prediction	no	no	yes
Burst length adjustment based on feedbacked void size	no	no	yes
Burst length greater than B_{min}	no	yes	yes
Padding conditions	no padding	completed burst length $< B_{min}$	(completed burst length $<$ predicted burst length) or ((predicted burst length $< B_{min}$) and (completed burst length $< B_{min}$))
Responsible node	ingress	ingress	ingress and core
System complexity	$O(M)$	$max(O(M), O(p))$	$max(O(M), O(p), O(N))$
Additional memory (bytes)	unused	unused	$N * VOIDSIZE$

The following are the simulation results and analyzes to evaluate the effectiveness of the SDBAP model.

4. Simulation and analysis

The tested network is an NSFNET with 14 nodes and the simulation environment is NS2 [18] with obs-0.9a package.

Assuming that ingress nodes can provide three service classes (class0, class1 and class2), there are 3 assembly queues ($q(i)$, $i = 0, 1, 2$) arranged accordingly. The assembly parameters corresponding to three queues are shown in Table 2, where the difference of the two time thresholds between two consecutive priority classes is 0.05 ms to ensure isolation between priority bursts. Similarly, the difference of the two offset times between two consecutive priority classes is also 0.05 ms. Setting this same difference on the queues ensures the same burst assembly delay (the sum of assembly time and offset time) on the queues.

Packets arriving at the ingress node are assumed to have a Poisson distribution with sizes in range [500, 1000] bytes. With the normalized load arriving at all 3 queues is 0.2 (the lowest load), the average sizes (in bytes) of all completed bursts are as shown in Table 3. With the suggestion in [19], the chosen minimum burst length is $B_{min} = 30000$ bytes; since this value approximates the average burst size generated at class0.

The simulation is divided into 2 phases: Phase 1 from 0 to 0.5 s, the normalized load arriving at 3 queues

Table 2. Simulation parameters.

	class0	**class1**	**class2**
Queue	$q(0)$	$q(1)$	$q(2)$
Timer of each queue	$timer(0)$	$timer(1)$	$timer(2)$
Time threshold (ms)	$T_a(0) = 0.4$	$T_a(1) = 0.45$	$T_a(2) = 0.5$
Offset time (ms)	$T_o(0) = 0.3$	$T_o(1) = 0.25$	$T_o(2) = 0.2$
Length threshold (ms)	$L_a(0) = B_{min}$	$L_a(1) = B_{min}$	$L_a(2) = B_{min}$
Estimated length (ms)	$L_e(0)$	$L_e(1)$	$L_e(2)$

Table 3. The average size (in bytes) of completed bursts.

Simulation time (s)	**class0**	**class1**	**class2**
0.1	32500	38600	42400
0.2	32400	38500	42600
0.3	32500	38800	42500
0.4	32400	38500	42500
0.5	32500	38600	42500
0.6	32600	38800	42600
0.7	32500	38500	42500
0.8	32600	38500	42600
0.9	32500	38600	42600
1.0	32500	38600	42500

is 0.2 and Phase 2 from 0.5 to 1s, the loads change to 0.1, 0.25 and 0.25 at queues 0, 1, 2, respectively. The goal of splitting into 2 phases is to compare the efficiency of service differentiation burst assembly models when varying the incoming loads of different priority flows. Three considered service differentiation burst assembly models are POQA, QDBAP and SDBAP.

The simulation objectives include:

1. Comparing the burst/byte loss rate of the priority classes. In fact, burst loss rate can be a suitable measure to evaluate the performance of OBS networks, but because a burst is the result of a collection of many packets inside, byte loss rate would be a more accurate scale. In subsection 4.1, both burst and byte loss rates will be analyzed and evaluated simultaneously;

2. Comparing the number of used padding bytes if the completed burst is less than the threshold B_{min}; and

3. Comparing the throughput fairness between priority classes, which is measured by the throughput fairness index (TFI) as Equation 5 [20].

$$TFI = \frac{(\sum_{i=1}^{n} \sigma_i y_i)^2}{n \sum_{i=1}^{n} (\sigma_i y_i)^2}, \tag{5}$$

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

4. Comparing the estimation error, which is the difference between the estimated length and the measured length of the completed burst, calculated by Equation 6

$$R_E = \frac{(\sum_{i=1}^{M}(|L - L_e|/L)}{M}, \tag{6}$$

where M is the number of consecutive assembly times, L is the measured length, and L_e is the estimated burst length per assembly time.

5. Comparing the average burst formation time of POQA, QDBAP and SDBAP.

4.1. Comparison of the burst/byte loss rate

As shown in Figures 10 and 11, there is no significant difference between burst loss rate (Figure 10) and byte loss rate (Figure 11) for class0, class1, class2 and all 3 classes. The burst/byte loss rate of SDBAP for each class and all 3 classes is always the smallest. This thanks to padding and the length threshold adjustment, which create the bursts that fit well with voids and increase the successful scheduling rate. As a result, the burst/byte loss rate is reduced.

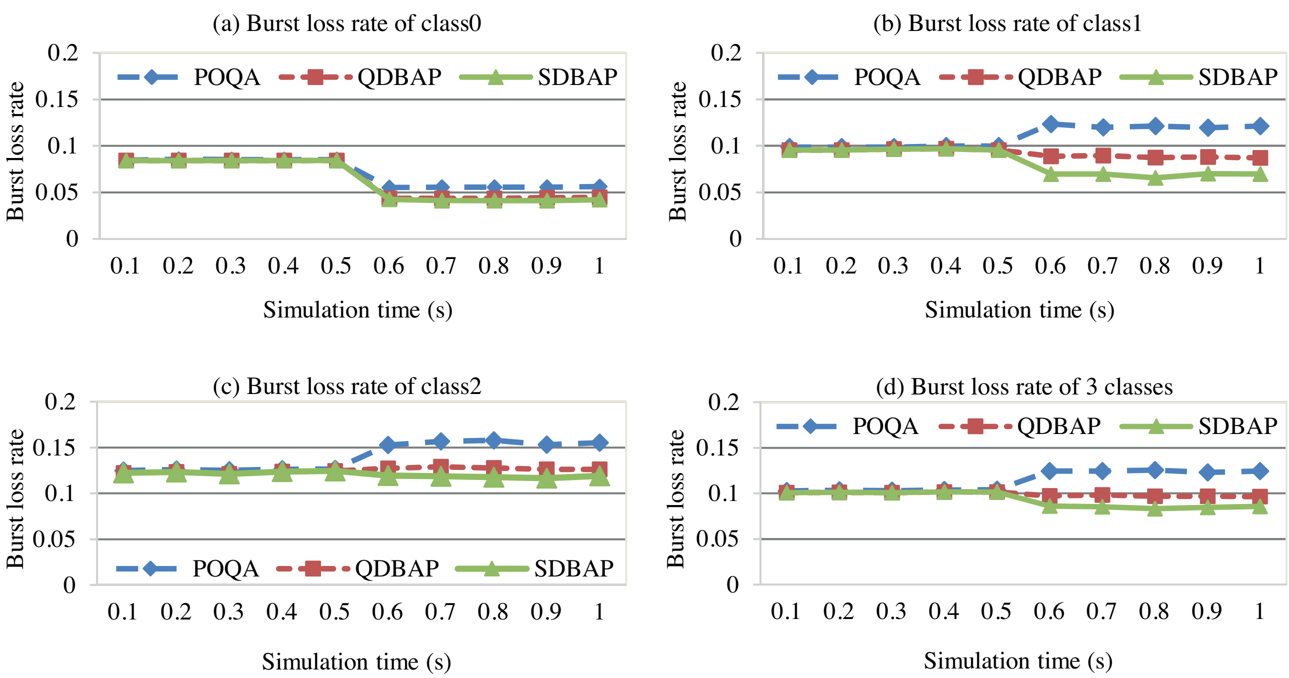

Figure 10. Comparison of the burst loss rate of each class and all 3 classes between POQA, QDBAP and SDBAP.

4.2. Comparison of the number of padding bytes

In SDBAP (and also QDBAP), data from low-priority queues is get to pad the high priority-burst if its length has not reached B_{min}. However, if the low-priority queue is empty, padding bytes should be used. Figure 12 shows a comparison of the number of used padding bytes in POQA, QDBAP and SDBAP, where the number of used padding bytes in SDBAP is much lower than that in POQA, but still a little higher than that in QDBAP.

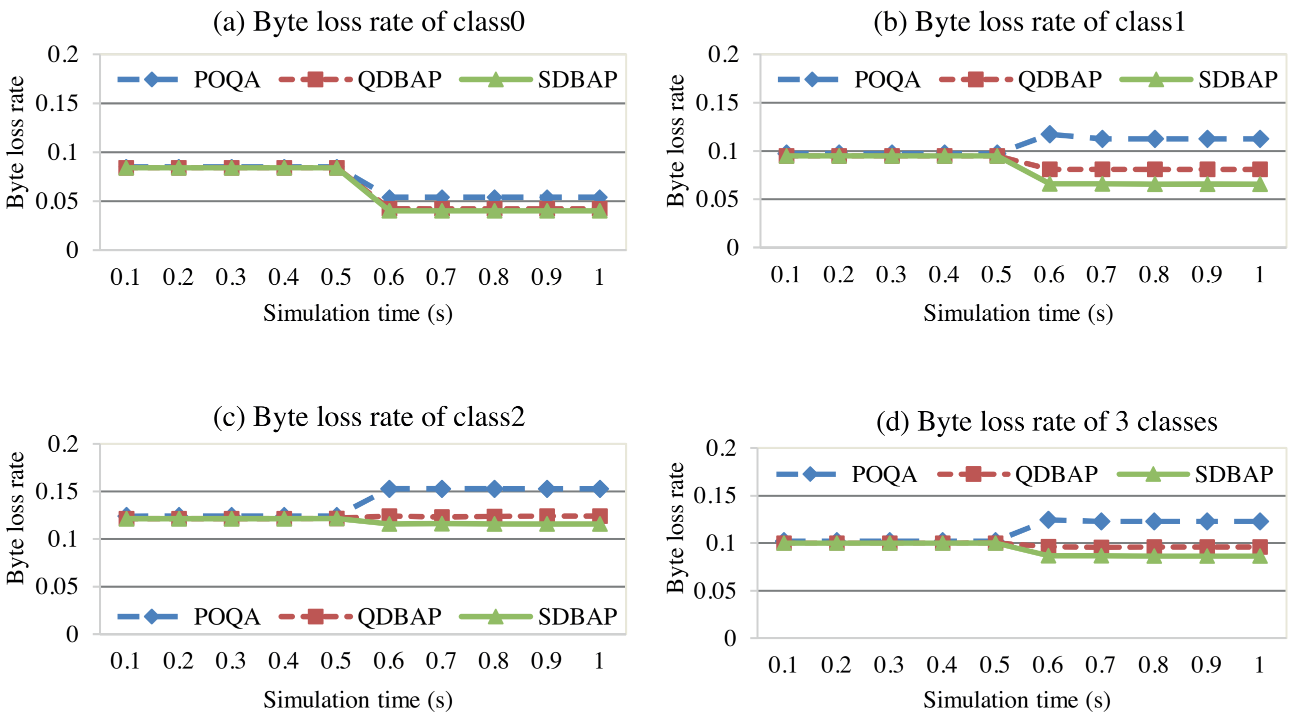

Figure 11. Comparison of the byte loss rate of each class and all 3 classes between POQA, QDBAP and SDBAP.

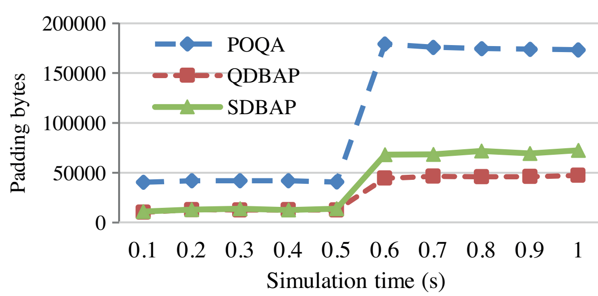

Figure 12. Comparison of the padding bytes between POQA, QDBAP and SDBAP.

The reason is that POQA does not have the padding mechanism, so a high number of padding bytes must be used; while QDBAP stops padding if the burst length reaches B_{min}. With SDBAP, padding is continued when the completed burst is greater than B_{min} and stops only when this length threshold is reached or low-priority queues are empty. Thus, as shown in Figure 13, the completed bursts are almost greater than B_{min}, but in some cases there still exist the bursts smaller than B_{min}. This is explained by the fact that at low-priority queues, since part of data has been used to pad to the higher priority burst, the rest is sometime not enough to be aggregated into a burst and padding bytes must then be used such that the completed burst is at least equal to B_{min}. Although SDBAP uses a larger number of padding bytes than QDBAP, the number of completed bursts is less due to the maximization of completed burst size and as a result, fewer bursts are generated, thus reducing the burst loss/ bytes in the network.

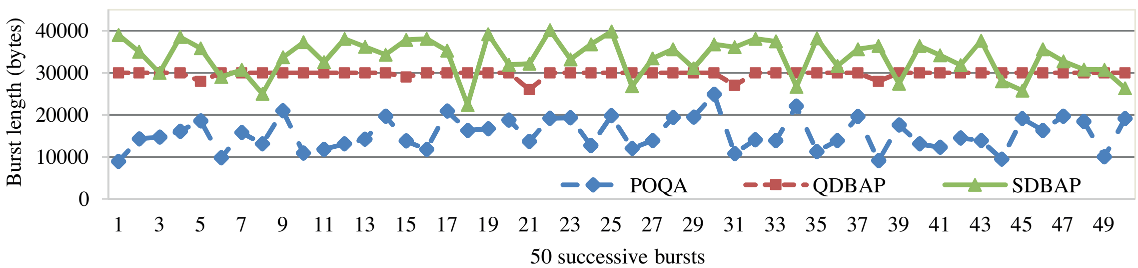

Figure 13. Comparison of the length of 50 consecutive completed bursts between POQA, QDBAP and SDBAP.

4.3. Comparison of the throughput fairness

In OBS networks, the metrics such as burst/byte loss, bandwidth utilization and throughput are often used to evaluate network performance. In addition, for the differentiated priority flows that share the same link, it is necessary to ensure the throughput fairness between them [20]. A suggestion of the throughput fairness index (Equation 5) has been proposed in [20] and this paper continues to use this formula.

In Phase 1, when the incoming load of the priority flows is equal (0.2), the rate of actual load to allocated bandwidth (y_i) is approximately the same, but when there is a change in incoming load at Phase 2, where the high-priority flow decreases (0.1), while the low-priority flow increases (0.25), the value y_i changes significantly. Specifically, with POQA (Figure 14a), the value y_1 is significantly reduced compared to y_2 and y_3. This proves that the ratio of used bandwidth to allocated bandwidth of the high priority flow is quite low. However, this issue was significantly improved in the QDBAP and SDBAP models (Figures 14b and 14c), where, when comparing between QDBAP and SDBA, the value y_i of SDBAP was slightly higher (about 1.5%) than that of QDBAP. These values y_i determine the throughput fairness of POQA, QDBAP and SDBAP through their TFI values (Equation 6). As shown in Figure 14d, the throughput fairness of SDBAP is the best, with its value TFI is close to 1. It is thanks to the padding enhancement and adaptive burst length adjustment that increase the successful scheduling probability of priority classes, thus also increase the fairness throughput.

4.4. Comparison of the estimation error

Estimation error comes from having to specify the burst length to be carried in BCP, while burst aggregation has not been completed. The estimated error is determined based on the difference between the actual measured length from the estimated length shown in Equation 6. As is shown in Figure 15, the estimated error of SDBAP is no greater than that of QDBAP, since they use the same computation method (Equation 1). But, when compared with POQA, the estimated error of SDBAP is significantly smaller. In fact, this estimated error is reduced by immediately resetting the length threshold equal to the estimated value during the time of burst aggregation. However, since the adjustment of the length threshold is come from two predicted values, the predicted void size and the estimated length, the estimation error is inevitable

4.5. Comparison of the average burst formation time

Burst formation time includes burst assembly time and padding time. With POQA model, there is no padding so burst formation time is burst assembly time, but with QDBAP and SDBAP, burst formation time includes burst assembly time and padding time so it is larger than the burst formation time of POQA model. Figure 16 shows the average burst formation time of POQA, QDBAP and SDBAP.

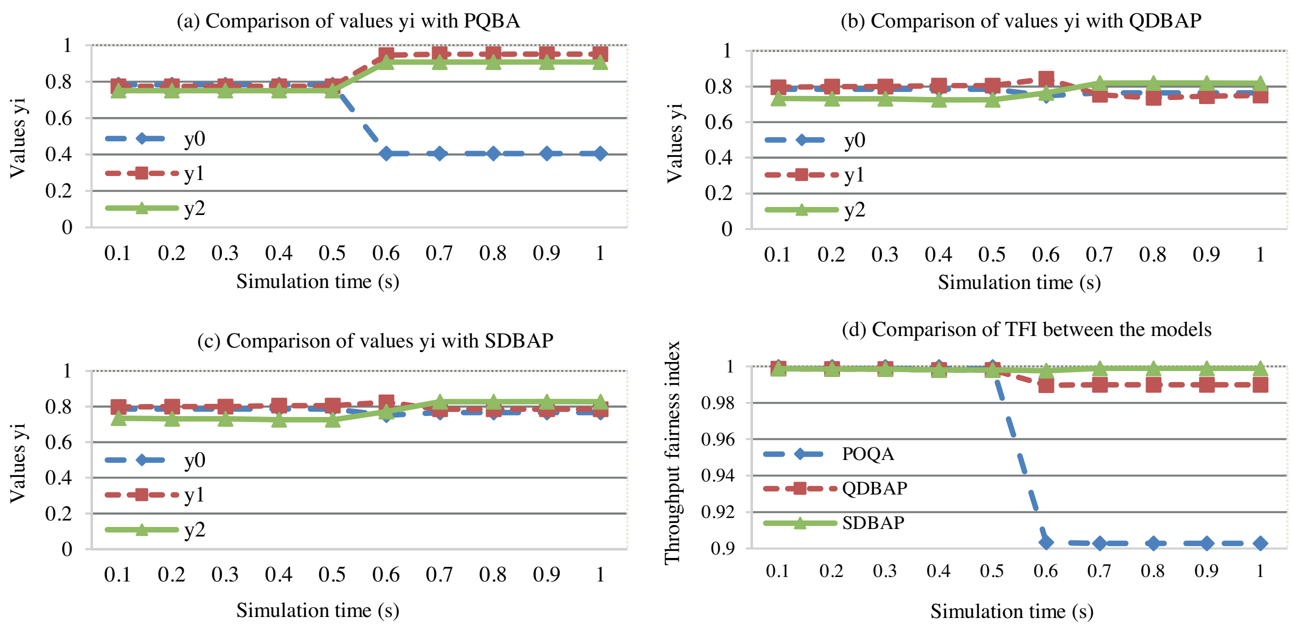

Figure 14. Comparison of the throughput fairness between POQA, QDBAP and SDBAP.

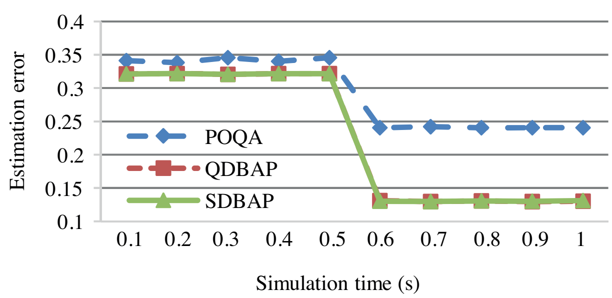

Figure 15. Comparison of the estimation error between POQA, QDBAP and SDBAP.

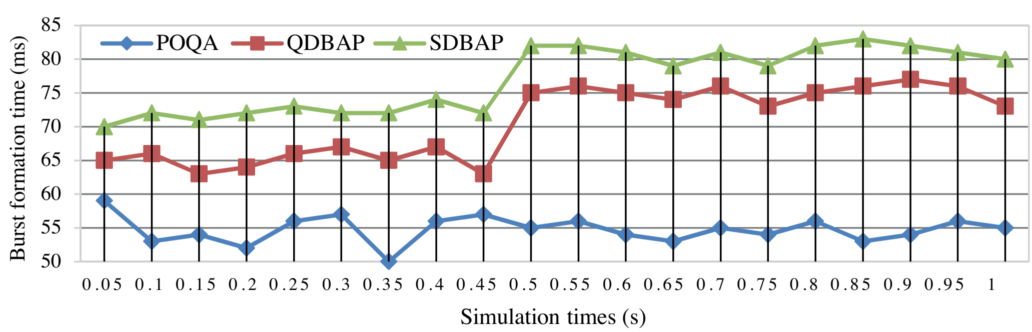

Figure 16. Comparison of the average burst formation time between POQA, QDBAP and SDBAP.

The time (ms) shown in Figure 16 is the average burst formation time per 0.05 ms of simulation, where the SDBAP burst formation time is the longest due to the process of extracting the void size from received NACKs and that of adjusting the length threshold of the assembly algorithm. QDBAP does not have this operation so its burst formation time is less and POQA has neither length threshold adjustment nor padding, so it has the shortest burst formation time.

At the 0.5th second, there is a change in incoming traffic load at the queues so there is a variation in burst formation time. With QDBAP and SDBAP, since more padding is performed, the burst formation time is slightly increased (13%). SDBAP always takes longer than QDBAP (7%) due to the length threshold update operation. POQA has no padding so there is no change in its burst formation time.

5. Conclusion

The model of service differentiation burst assembling and padding (SDBAP) has proven its advantages based on burst loss rate, number of used padding bytes and throughput fairness when compared with other similar models such as POQA and QDBAP. However, SDBAP depends on the result of void size prediction and burst length estimation, so calculation error is inevitable. SDBAP also suffers from a longer execution time due to many additional operations such as length threshold adjustment and padding. Furthermore, this model requires feedback from the core node, which increases the exchange of control packets over the network and increases the computational complexity of the nodes. Further studies on predicting the best point time to send burst packets, changing burst lengths in accordance with voids, and determining appropriate load levels for implementing this model will help to further improve transmission efficiency in OBS networks.

Acknowledgment

This work was supported/partially supported by Hue University under the Core Research Program, Grant No. NCM.DHH.2019.05

References

[1] Yoo SJB. Optical Packet and Burst Switching Technologies for the Future Photonic Internet. Journal of Lightwave Technology 2006; 24 (12): 4468-4492. doi: 10.1109/JLT.2006.886060

[2] Zalesky A. To burst or circuit switch?. IEEE/ACM Transaction Network 2009; 17 (1): 305-318. doi: 10.1109/TNET.2008.923718

[3] Reza MS, Hossain MM, Majumder SP. Contention Problem in Optical Burst Switching Network. In: International Conference on Computational Intelligence and Communication Networks; Bhopal, India, 2010; pp. 239-242. doi: 10.1109/CICN.2010.57

[4] Ge A, Callegati F, Tamil LS. On optical burst switching and self-similar traffic. IEEE Communications Letters 2000; 4 (3): 98-100. doi: 10.1109/4234.831037

[5] Xiong Y, Vandenhoute M, Cankaya HC. Control architecture in optical burst-switched WDM networks. IEEE Journal on Selected Areas in Communications 2000; 18 (10): 1838-1851. doi: 10.1109/49.887906

[6] Vokkarane VM, Zhang Q, Jue JP, Chen B. Generalized burst assembly and scheduling techniques for QoS support in optical burst-switched networks. In: Global Telecommunications Conference GLOBECOM ’02. IEEE; Taipei, Taiwan; 2002; 3: 2747-2751. doi: 10.1109/GLOCOM.2002.1189129

[7] Kitayama K, Koga M, Morikawa M, Hara S, Kawai M. Optical burst switching network testbed in Japan. In: OFC/NFOEC Technical Digest. Optical Fiber Communication Conference; Anaheim, CA, USA; 2005; 3: 3-5. doi: 10.1109/OFC.2005.192713

[8] Amin AA et al. Development of an Optical-Burst Switching Node Testbed and Demonstration of Multibit Rate Optical Burst Forwarding. Journal of Lightwave Technology 2009; 27 (16): 3466-3475. doi: 10.1109/JLT.2009.2015776

[9] Vo VMN, Le VH, Nguyen HS, Le MT. A model of QoS differentiation burst assembly with padding for improving the performance of OBS networks. Turkish Journal of Electrical Engineering & Computer Sciences 2018; 26 (4): 1783-1795. doi:10.3906/elk-1710-45

[10] Yoo M, Qiao C, Dixit S. Optical burst switching for service differentiation in the next-generation optical internet. IEEE Communication Magazine 2001; 39 (2): 98-104. doi:10.1109/35.900637

[11] Hwang IY, Ryou JH, Park HS. Offset-time compensation algorithm-QoS provisioning for the control channel of the optical burst switching network. Lecture Notes in Computer Science 2005; 3391: 362-369.

[12] Yoo M, Qiao C, Just-enough-time (JET): A high speed protocol for bursty traffic in optical networks, In: IEEE/LEOS Technology Global Information Infrastructure; Montreal, Canada; 1997. pp. 26-27.

[13] Sui Z, Zeng Q, Xiao S. An offset differential assembly method at the edge of OBS network. In: Asia-Pacific Optical Communications; Shanghai, China; 2005. 6201.

[14] Vo VMN, Pham TD, Dang TC, Le VH. A mechanism of QoS differentiation based on offset time and adjusted burst length in OBS networks. Turkish Journal of Electrical Engineering & Computer Sciences 2020; 28 (5): 2808-2820. doi: 10.3906/elk-1906-87

[15] Triay J, Rubio J, Cervelló C. An Optical Burst Switching Control Plane Architecture and its Implementation. In: 2th Open European Summer School EUNICE; Stuttgart, Germany; 2006. pp. 1-7.

[16] Vokkarane VM, Haridoss K, Jue JP. Threshold-Based Burst Assembly Policies for QoS Support in Optical Burst-Switched Networks. In: The Convergence of Information Technologies and Communications; Boston, MA, United States; 2002. pp. 125-136. doi: 10.1117/12.475291

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

[18] https://www.isi.edu/nsnam/ns/

[19] Kantarci B, Oktug SF, Atmaca T. Performance of OBS techniques under self-similar traffic based on various burst assembly techniques. Computer Communications 2007; 30 (2): 315-325. doi:10.1016/j.comcom.2006.08.035

[20] Le VH, Vo VMN, Le MT. Throughput-based fair bandwidth allocation in OBS networks. Wiley ETRI Journal 2018; 401 (5): 624-633. doi: 10.4218/etrij.2017-0253