

## A model of optimal burst assembly for delay reduction at ingress OBS nodes

Viet Minh Nhat VO<sup>1,\*</sup>, Van Hoa LE<sup>1</sup>, Hoang Son NGUYEN<sup>2</sup>

<sup>1</sup>Department of Information Technology, Hue University, Hue, Vietnam

<sup>2</sup>Department of Mathematics, Hue University of Science, Hue, Vietnam

Received: 27.08.2016

Accepted/Published Online: 09.02.2017

Final Version: 05.10.2017

**Abstract:** Burst assembly plays an important role in reducing the end-to-end delay of packets transported through optical burst switching (OBS) networks. Several methods have been proposed to reduce the delay of the packets buffered at ingress OBS nodes. However, these have created significant estimation errors, which result in wasting the reserved bandwidth or increasing the delay of excess packets. In this paper, we propose a model of optimal burst assembly for delay reduction, which minimizes the estimation error, eliminates the excess packets, and decreases the blocking probability of scheduling.

**Key words:** Ingress optical burst switching node, burst assembly, delay reduction, statistics-based estimation, Engset model

### 1. Introduction

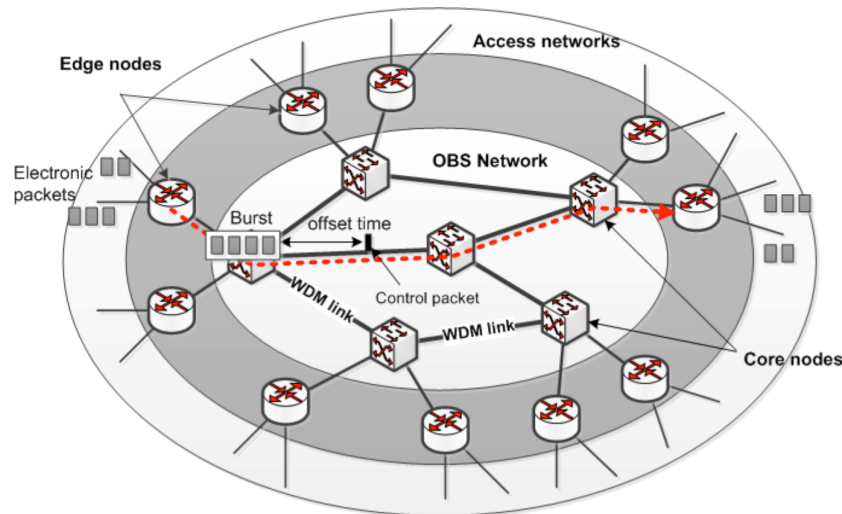
Optical burst switching (OBS) [1] is a promising technology for implementing the next generation optical Internet, in order to meet the rapid growth of Internet traffic and the increasing deployment of new services (e.g., VoIP, video on demand, cloud computing). The implementation of OBS technology exploits the bandwidth of fiber networks more efficiently to create a flexible and configurable network infrastructure at burst granularity and to handle the bursty traffics generated by the mentioned services.

Figure 1 shows an example of OBS networks, in which their edge and core nodes are connected via WDM links. Ingress nodes are responsible for collecting the electronic packets from access networks (e.g., IP packets) and aggregating them into the larger carriers, called bursts. When a threshold of time or burst length is reached, a burst control packet (BCP) is sent ahead on a dedicated control channel to reserve the required bandwidth and configure the core nodes along a path from source to destination. Its burst follows after an offset time on the chosen data channel and is all-optically switched at the core nodes on this path.

The end-to-end delay of passing a packet through an OBS network is mainly due to four components: (1) assembly delay, (2) offset time, (3) burst forwarding delay at core nodes, and (4) propagation delay in the core network. The last two delays are usually dependent on the chosen path and bandwidth availability; therefore, they cannot be reduced with an implemented protocol. Only the first two delays, i.e. the assembly delay and the offset time, which also is called buffering delay, could be reduced.

The basic models of burst assembly, namely the timer-based model [2], the length-based model [3], and the hybrid model [4,5], do not study reducing the buffering delay; however, the models of burst assembly for delay reduction [6–11] try to eliminate the offset time by sending the BCP early, before the completion of its

\*Correspondence: [vvmnhat@hueuni.edu.vn](mailto:vvmnhat@hueuni.edu.vn)



**Figure 1.** An example of OBS networks, where their edge and core nodes are connected via WDM links.

burst. However, it needs to estimate the length of the completed burst, because this information must be carried in the BCP. Various estimation methods have been used in [6–11]. However, these models of burst assembly have the following limitations: (1) they always have a certain estimation error that has not yet been minimized; (2) the estimation error of the current assembly has not been used effectively for the next assemblies; (3) the chosen thresholds have a significant impact on the estimation error, yet they have not been studied yet. In this paper, we propose a model of optimal burst assembly for delay reduction, which addresses the mentioned drawbacks.

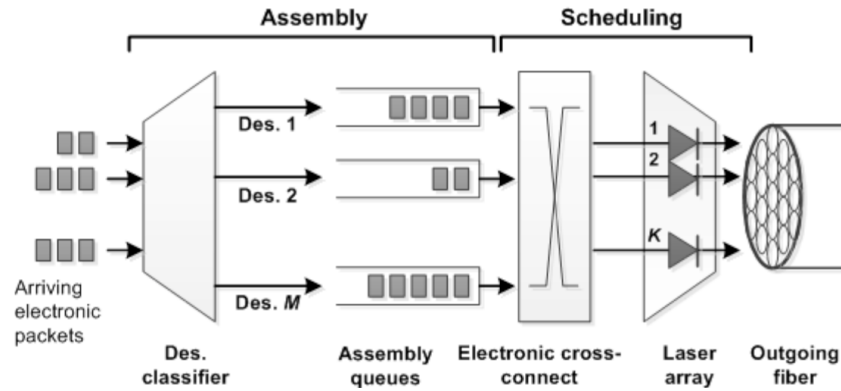
The remainder of this paper is organized as follows: Section 2 begins with a brief overview of basic burst assembly and then analyzes the previous proposals of burst assembly for delay reduction. Section 3 describes our model of optimal burst assembly for delay reduction and compares and analyzes the simulation results. Section 4 analyzes the blocking probability of our model and the numerical results. Section 5 gives the conclusion.

## 2. Related works

### 2.1. Basic models of burst assembly

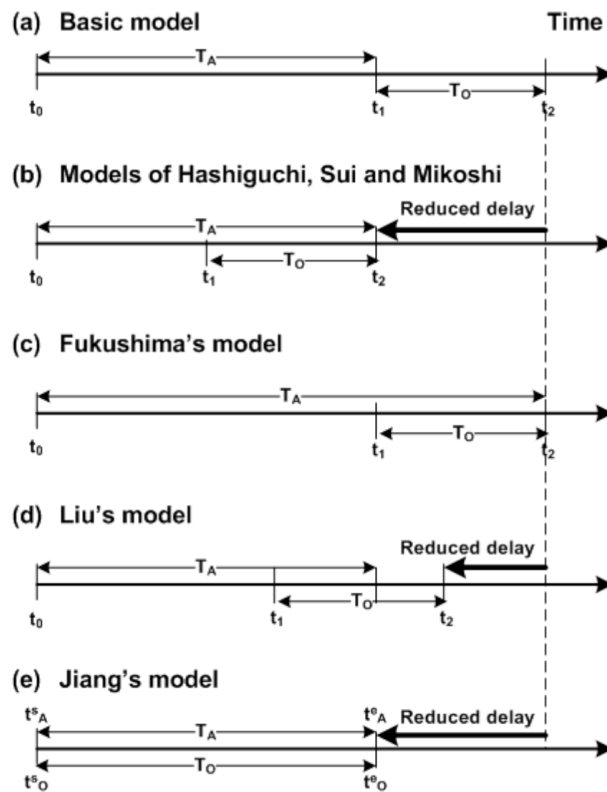
Burst assembly is one of two main operations implemented at an ingress OBS node. As shown in Figure 2, the arriving electronic packets are classified based on their destinations. There are two basic models of burst assembly: the time-based model [2] and the length-based model [3]. In the time-based model, a timer is initiated when the first packet arrives at an empty queue, and a burst is completed when the timer reaches a predefined time threshold ( $T_a$ ). In the length-based model, the maximum number of packets in a burst or maximum in-bytes burst size is defined as a length threshold ( $L_a$ ). A burst is completed when this threshold is reached.

The time-based model limits the average assembly delay, but can generate very small-size bursts in low incoming traffic. Conversely, the length-based model ensures the length of completed bursts, but can cause a huge delay. For this reason, hybrid models that are based on both thresholds of time and length have been proposed [4,5], in which a burst is completed when one of these two thresholds is reached. However, whether based on time, length, or both, these basic assembly models suffer from a buffering delay that includes the assembly delay ( $T_a$  or  $T < T_a$  when the length threshold is reached first) and the offset time ( $T_o$ ), as shown in Figure 3a. This delay can be significant for packets bounded by their round trip time; therefore, reducing



**Figure 2.** Two main operations at an ingress OBS node include the assembly of electronic packets into a burst and the scheduling of the completed burst on a channel of outgoing fiber.

the buffering delay is required. There are several proposals of burst assembly for delay reduction, which are analyzed in the next section.



**Figure 3.** Reduced delays in the previous models of burst assembly for delay reduction (from (b) to (e)) in comparison to the basic model (a).

### 2.2. Previous models of burst assembly for delay reduction

The first model of burst assembly for delay reduction is proposed by Hashiguchi [6]. In this model, the BCP is sent at time  $t_1$  before the burst completion (Figure 3b). In this way, the burst is sent at time  $t_2$  without waiting

for an offset time, as in the basic assembly models. This means that the packets assembled in the current burst are reduced by a delay of  $T_o$ . In order to estimate the length of the completed burst, Hashiguchi used a method based on the average rate of packets arriving in the estimation period of  $T_a - T_o$ . The estimated length is thus given by

$$L_e = \alpha \times L_w \times \frac{T_a}{T_a - T_o}, \quad (1)$$

where  $L_w$  is the length of current burst in the estimation period and  $\alpha$  is a control parameter.

The models given by Sui [7] and Mikoshi [8] are similar to that reported by Hashiguchi, yet different from their estimation methods. Specifically, Sui estimates the burst length by using an adaptive autoregressive (AAR) linear filter, with the burst lengths measured in  $M - 1$  previous assemblies and the amount of packets arriving in the estimation period. Hence, the estimated length is given by

$$L_e = \sum_{i=1}^{M-1} w(i)L(i) + \alpha \times L_w \times \frac{T_a}{T_a - T_o}, \quad (2)$$

where  $L(i)$  is the length measured at the  $i$ th assembly ( $1 \leq i \leq M$ ) and  $w(i)$  is its impact weight. Note that  $\alpha = w(M)$  and  $\sum_{i=1}^M w(i) = 1$ .

Mikoshi estimates the length of completed burst based on the Jacobson/Karels algorithm [12] with some changes, as follows: firstly, the estimation error  $E(n)$  of the  $n$ th assembly is the difference between the measured length  $L(n)$  and the estimated length  $L_e(n)$ . Secondly, this estimation error is used to calculate the estimated length of the  $n + 1$ st assembly. Thirdly, a parameter  $D(n + 1)$ , called the deviation of the  $n + 1$ st assembly, is calculated based on the deviation  $D(n)$  and the estimation error  $E(n)$  of the  $n$ th assembly. This quantity is also involved in determining the final estimated length. The equations used to estimate the burst length in Mikoshi's model are given by

$$\begin{aligned} E(n) &= L(n) - L_e(n) \\ L_e(n + 1) &= L_e(n) + \alpha \times E(n) \\ D(n + 1) &= D(n) + \alpha \times (|E(n)| - D(n)) \\ L_e(n + 1) &= \mu \times L_e(n + 1) + \phi \times D(n + 1) \end{aligned}, \quad (3)$$

where  $\alpha$ ,  $\mu$ , and  $\phi$  are the weight factors.

In his model, Fukushima [9] allows aggregating the packets that arrive during the offset time into the current burst (Figure 1c), and suggests an estimation formula based on the average rate ( $\lambda_{avg}$ ) of  $M$  latest-arriving packets, as follows:

$$L_e = L + \lambda_{avg} \times T_o, \quad (4)$$

where  $L$  is the length measured in the period of  $T_a$ .

If we generalize the assembly duration as the period in which all arriving packets are aggregated into the current burst, Fukushima's model is similar to the models given by Hashiguchi, Sui, and Mikoshi, but with a larger assembly period of  $T_a + T_o$ .

Liu's model [10] is similar to Fukushima's as well, but with a hybrid assembly model. Specifically, the BCP will be sent when the timer reaches a preset minimum time threshold ( $T_a$ ) or a preset minimum length threshold ( $L_{min}$ ). Liu proposes an estimation method based on the difference between the average rate of the

packets arriving in the current assembly ( $\lambda_{cur}$ ) and that of the previous assembly ( $\lambda_{pre}$ ). The estimated length is thus given by

$$L_e = L + \left( \lambda_{pre} + (\lambda_{cur} - \lambda_{pre}) \times \frac{T_o}{T_w + T_o} \right) \times T_o, \quad (5)$$

where  $T_w + T_o$  is the assembly period. If the timer reaches the minimum time threshold first,  $T_w = T_a$  and, thus, Liu's model is equivalent to the basic time-based model (Figure 3a). In the case of reaching the minimum length threshold first, as shown in Figure 3d, the packets assembled in the current burst are reduced by a delay of  $t_1 + T_o - T_a$ .

With Jiang's model [11], the BCP is sent as soon as the first packet arrives at an empty queue; thus, the offset time of  $T_o$  is equal to the time threshold of  $T_a$ . Jiang uses a hybrid model in which both thresholds of time and burst length are calculated flexibly in the latest assembly. Specifically, the current time threshold is calculated as the average of  $M$  previous time thresholds with the following formula:

$$T_a = \frac{\sum_{i=1}^M T_a(i)}{M} \quad (6)$$

and the current length threshold is adjusted step-by-step ( $step = step \pm 1$ ), depending on increase/decrease in incoming traffic, with the following formula:

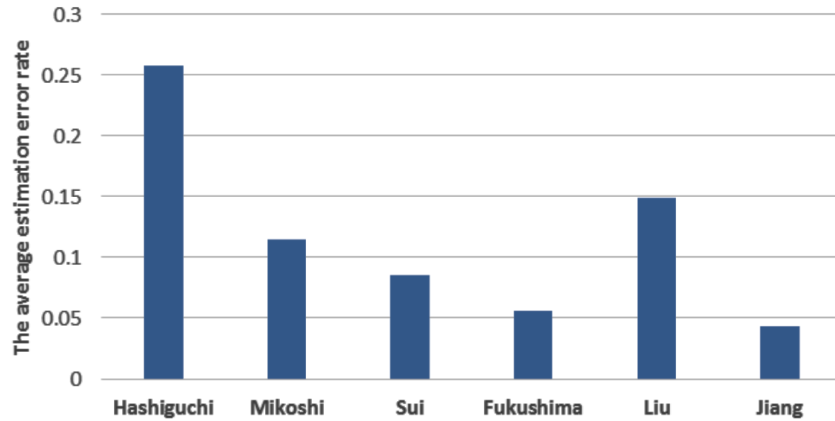
$$L_e = L_{min} + step \times (L_{max} - L_{min})/N, \quad (7)$$

where  $N$  is the total adjustment steps in a range of minimum and maximum burst length [ $L_{min}$ ,  $L_{max}$ ].

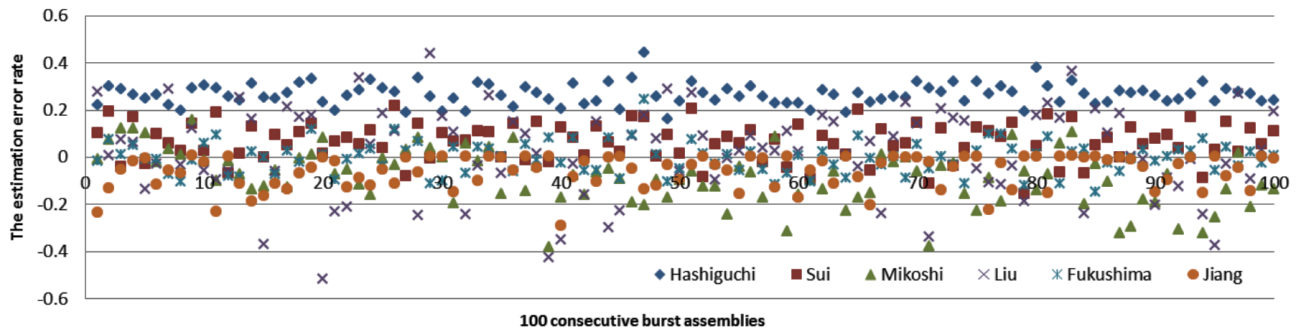
The estimation error of Jiang's model can be minimized when the length threshold is reached first. The error is zero if all packets are the same size, or the length threshold is the common multiple of all arriving packet sizes. However, if the time threshold is reached first, there exists an estimation error of  $L - L_e$ .

In short, the previous models of burst assembly for delay reduction attempt to involve the offset time in the assembly delay. However, they still show the following disadvantages:

- (1) Their estimation errors are significant, as shown in Figure 4, which causes reserved bandwidth waste if the estimated length is longer than the completed one. In the case of the estimated length being smaller than the completed one, the excess packets will be assembled into the next burst. As a result, these packets are subjected to a supplemental delay, which is equal to the buffering delay.
- (2) Most previous models have not effectively used the estimation error for the next assemblies. In [8], Mikoshi uses the estimation error of  $L - L_e$  to directly adjust the estimated length for the next assembly. However, this approach is not unproblematic, as the estimation error does not converge on a minimum value (e.g., zero). Figure 5 shows the distribution of the estimation error rate in 100 consecutive burst assemblies among the previous models.
- (3) In order to estimate the completed length, some models base the measured lengths (in Sui's model) on the time thresholds of  $M$  last assemblies (in Jiang's model), or on the average rate of  $M$  last-arriving packets (in Fukushima's model). These statistics-based approaches help more accurate estimations; however, they suffer from huge computational cost, especially when the amount of packets arriving at ingress OBS nodes is very high. Therefore, it is essential that the calculation volume is reduced by decreasing the estimation time properly, and estimation accuracy must be ensured.



**Figure 4.** A simulation-based comparison of the average estimation error rate (RE) between the previous models of burst assembly for delay reduction, where  $R_E = \frac{\sum_{i=1}^M (|L(i) - L_e(i)| / L(i))}{M}$ .



**Figure 5.** Distribution of the estimation error rates in 100 consecutive burst assemblies among the previous models, where that of Jiang is the nearest to the horizon.

- (4) Most previous models use a fixed time threshold ( $T_a$ ) and then try to estimate the length of the completed burst ( $L_e$ ). Jiang has used an improved hybrid model, in which the time threshold and the length threshold are adjusted flexibly. However, because  $T_a$  is calculated as the average of the time thresholds of  $M$  last assemblies, it does not reflect the latest trend of arriving traffics. Furthermore, the step-by-step adjustment of  $L_e$  will not meet up with the burstiness of the traffic and short peaks.

The following section describes in detail our model of optimal burst assembly for delay reduction that addresses the mentioned drawbacks.

### 3. Model of optimal burst assembly for delay reduction

#### 3.1. Algorithm of two-phase burst assembly

Our model of burst assembly for delay reduction is based on the idea of sending the BCP early at time  $t_1$ , and its burst is sent at time  $t_2$ . Thus, the packets assembled in the current burst are reduced by a delay of  $T_o$  (Figure 3b).

Our model is based on the algorithm of two-phase burst assembly, as follows:

- Phase 1 (time-based assembly): As the first packet arrives at an empty queue, a timer is triggered. The BCP is sent only when the timer reaches time  $t_1$ ; the estimation time is thus equal to  $T_a - T_o$ . The

estimated length ( $L_e$ ) is then calculated by using our improvement of the TW-EWMA algorithm [13], which is presented in the next section.

- Phase 2 (length-based assembly): The process of burst assembly is still continued, but now it is based on the estimated length threshold of  $L_e$ . The burst is only completed once this threshold is reached.

With our proposed model, the estimation error is minimized. In fact, it is zero when all arriving packets have the same size or  $L_e$  is the common multiple of all arriving packet sizes. In the event that the arriving packets are of various sizes, the condition for completing a burst is  $L_e - max_p \leq |b| \leq L_e$ , where  $|b|$  is the current length of burst  $b$  and  $max_p$  is the possible maximum size of arriving packets. It is clear that this approach will cause a bit of wasted bandwidth when the estimated length is larger than the completed one. However, it ensures that no excess packet is moved to the next burst.

### 3.2. An improvement of the TW-EWMA algorithm

The original TW-EWMA algorithm [13] estimates the rate of packets arriving in a time window instead of all arriving packets. It is significant for OBS networks in terms of computation costs, due to the huge amount of packets arriving at ingress OBS nodes. With the time window of  $T_a - T_o$ , the estimated rate of packets arriving in the period of  $T_o$  is given by

$$\lambda_e = (1 - \alpha) \times \lambda_{avg} + \alpha \times \frac{N}{T_a - T_o}, \quad (8)$$

where  $\alpha$  is a weight factor,  $\lambda_{avg}$  is the average rate of previous arriving packets, and  $N$  is the number of packets arriving in the time window.

In [13],  $\alpha$  was fixed at 0.3. This fixed value has an obvious negative impact on the estimated results, when the burstiness of traffic is popular in real networks. We improve the TW-EWMA algorithm by adjusting flexibly the weight factor  $\alpha$ , which varies depending on the increase/decrease in incoming traffic, with the formula

$$\frac{\alpha}{1 - \alpha} = \frac{\lambda_{cur}}{\lambda_{avg}} \Rightarrow \alpha = \frac{\lambda_{cur}}{\lambda_{avg} + \lambda_{cur}} \quad (9)$$

The estimated length in the end of Phase 1 is thus given by

$$L_e = L_w + T_o \times \lambda_e, \quad (10)$$

where  $L_w$  is the length measured in the period of  $T_a - T_o$ .

### 3.3. Algorithm of optimal burst assembly for delay reduction

Our algorithm of optimal burst assembly for delay reduction (OBADR) operates as follows: the packets arriving at an assembly queue are assumed to be taken from a list  $S_q$ , in which the information about each packet includes its arrival time and size. When the time for sending the control packet  $t_1$  is reached (Line 10), the estimated burst length  $L_e$  is calculated (Line 15) based on the rate of current arriving packets  $\lambda_{cur}$  (Line 13) and the average rate of previous arriving packets  $\lambda_{avg}$  (Line 13). The weight factor  $\alpha$  can be adjusted according to the changes of these two rates (Line 17). The next phrase of the OBARD algorithm is a burst assembly based on the estimated length ( $L_e$ ), in which a burst will be completed when the queue length  $|b|$  falls in the interval  $[L_e - max_p, L_e]$  (Line 19). This condition is intended to ensure that the size of the completed burst is always equal

to or less than the estimated length, so there will be no excess packets. The details of the OBADR algorithm are described as follows:

---

**OBADR Algorithm.**

---

**Input:**  $T_a$  // time threshold

$T_o$  // offset time

$S_q$  // list of packets arriving in queue

$max_p$  // possible maximal length of packets

**Output:**  $S_{burst}$  // list of completed bursts

**Begin**

```

1   $\alpha \leftarrow 1$ ; // initial value of weight factor  $\alpha$ 
2   $\lambda_{avg} \leftarrow 0$ ; // average rate of previous arriving packets
3   $S_{burst} \leftarrow \emptyset$ ;
4   $t_1 \leftarrow T_a - T_o$ ;
6  While ( $S_q \neq \emptyset$ ) do
7     $p \leftarrow$  first packet in queue;  $S_q = S_q \setminus \{p\}$ ;
8     $T_q \leftarrow s_p$ ; //  $s_p$  is the arriving time of packet  $p$ 
9     $b \leftarrow b + \{p\}$ ; // assembling packet  $p$  into burst  $b$ 
10   If ( $T_q \geq t_1$ ) then // Phase 1: send the control packet
11      $L \leftarrow |b|$ ; // current burst length
12      $\lambda_{cur} \leftarrow L / (T_a - T_o)$ ; // rate of packets arriving in  $T_a - T_o$ 
13      $\lambda_{avg} \leftarrow (1 - \alpha) * \lambda_{avg} + \alpha * \lambda_{cur}$ ;
14      $L_e \leftarrow L + T_o * \lambda_{avg}$ ; // estimation length of burst
15      $t_1 \leftarrow \infty$ ;
16      $\alpha \leftarrow \lambda_{cur} / (\lambda_{cur} + \lambda_{avg})$ ; // adjusting the weight factor  $\alpha$ 
18   End if
19   If ( $L_e - max_p \leq |b| \leq L_e$ ) then // Phase 2: send the burst
20      $L \leftarrow |b|$ ; // completed burst length
21      $S_{burst} \leftarrow S_{burst} + \{b\}$ ; // add a new burst into  $S_{burst}$ 
22      $b \leftarrow \emptyset$ ;
23      $t_1 \leftarrow T_q + |p|$ ;
25   End if
26 End while
27 Return  $S_{burst}$ ;

```

**End**

---



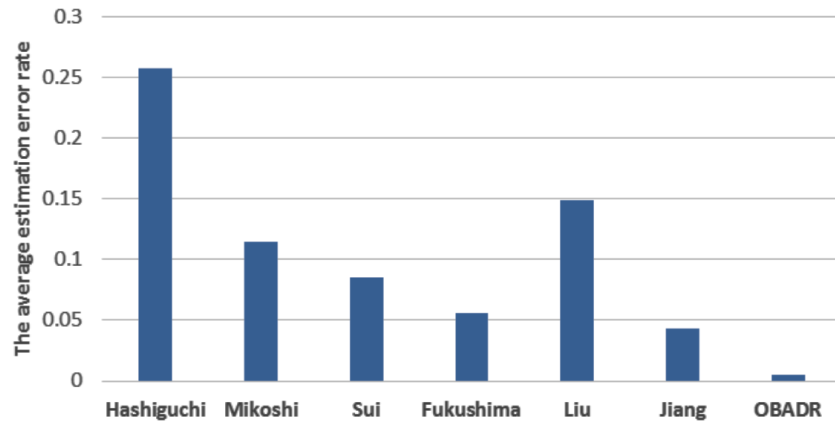
The complexity of the OBADR algorithm for one time of burst assembly is  $O(\log N)$ , where  $N$  is the number of packets assembled in a burst.

### 3.4. Simulation results and analysis

The OBADR algorithm is implemented in NS2 with the support of package obs0.9a, on a PC of 2.4 GHz Intel Core, 2 CPU, and 2G RAM. The packet arrival process is assumed to be Poisson and the packet sizes are uniformly distributed in the interval  $[500, \dots, 1000]$ . The parameters of the burst assembly include the time threshold  $T_a = 6$  ms, the offset time  $T_o = 2$  ms, and the load of 0.5 Erlang. The simulation goals are:

1. Comparing the average estimation error rate of the OBADR model to the previous models;
2. Comparing the distribution of the estimation error rate between the OBADR model and Jiang's model, which are the best among the previous models; and
3. Comparing the OBADR model and the previous models in terms of number of excess packets in 100 consecutive burst assemblies.

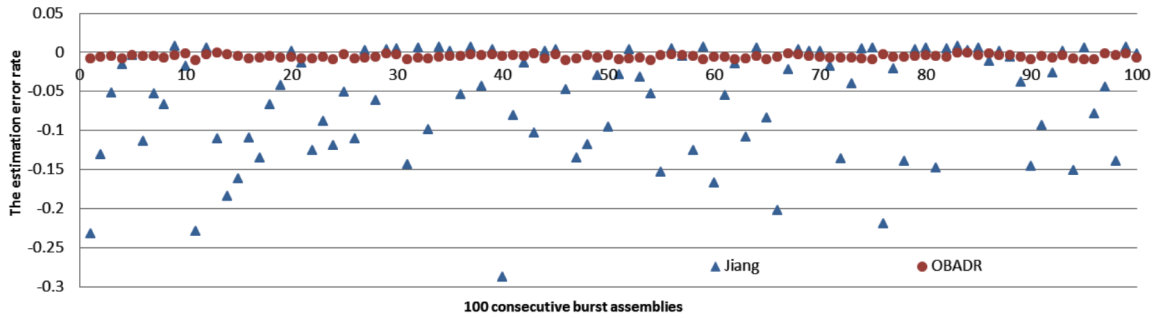
As shown in Figure 6, the OBADR model results in the least average estimation error rate, due to the burst length being estimated accurately by the improved TW-EWMA algorithm in Phase 1 and then using this length as the assembly threshold in Phase 2.



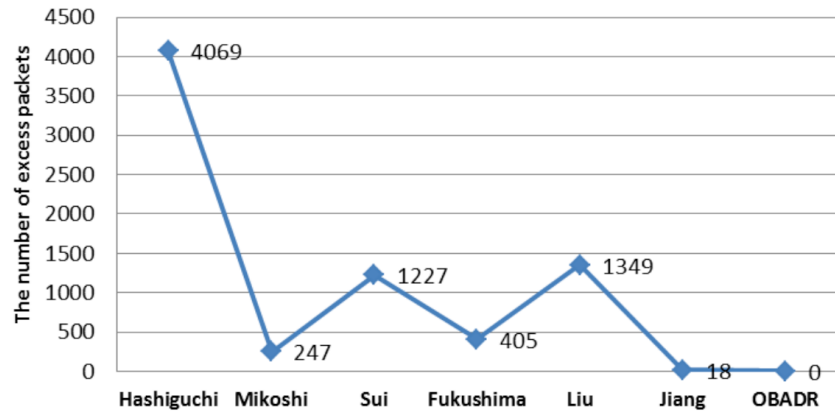
**Figure 6.** Simulation-based comparison of the average estimation error rate between the previous models and the OBADR model, in which the latter shows the lowest error rate.

Specifically, Figure 7 shows the distribution of the estimation error rate of the OBADR model and Jiang's model, in which the estimation error rate of the OBADR model is always negative and approaches zero. This reflects the accurate nature of the OBADR model, i.e. that a burst is only completed once its size is approximate or equal to the estimated length. In fact, the estimation error of the OBADR model is inevitable, because no arriving packets are sized to be a divisor of the estimated length. Only in a few exceptional cases the total length of arriving packets is equal to the estimated length, and then the estimation error is eliminated.

A simulation-based comparison of the number of excess packets between the OBADR model and the previous models is shown in Figure 8, in which no excess packet occurs in the OBADR model. This result was achieved due to the conditions in Line 19 of the OBARD algorithm, which ensures that the completed length is never larger than the estimated length ( $L_e$ ).



**Figure 7.** Comparison of the distribution of the estimation error rates in 100 consecutive burst assemblies between Jiang’s model and the OBADR model, in which the distribution of the latter is the nearest to the horizon.



**Figure 8.** Simulation-based comparison of the number of excess packets between the previous models and the OBADR model, in which the latter generates no excess packet.

#### 4. Impact of the OBADR model on the blocking probability of scheduling

##### 4.1. Engset model

As described in Figure 2, an ingress OBS node performs two sequential operations: assembly and scheduling, in which the output of the first operation is the input of the second one. Specifically, the bursts generated in the assembly are distributed on the available channels of outgoing fiber in the scheduling. With the implementation of the OBADR model in the first operation, we analyze how our model affects the blocking probability of scheduling at the outgoing fiber.

Let  $M$  be the number of sources corresponding to the number of queues, and  $K$  be the number of servers corresponding to the number of available channels at the outgoing fiber (Figure 2). Then we have a  $K$  server loss model with  $M$  sources. As analyzed in [14], with the assembly delay of  $T_a$  and the offset time of  $T_o$ , the  $K$  server loss model with  $M$  sources is equivalent to an Engset model with mean on times  $T_o + B$  and mean off times  $1/\lambda_{avg} + T_a$ , where  $B$  denotes the mean busy period for the output of each queue and  $1/\lambda_{avg}$  is the mean packet interarrival time.  $B$  is given by

$$B = \frac{1}{\mu} + \left( \frac{\lambda_{avg}}{\mu} \right) \times (T_a + T_o), \tag{11}$$

where  $1/\mu$  is the mean packet length (in units of time).

The stationary probabilities  $\{P_k | k = 0 \dots K\}$  for the number of busy channels are given by

$$P_k = \frac{\binom{M}{k} \times \rho^k}{\sum_{q=0}^K \binom{M}{q} \times \rho^q}, \tag{12}$$

where  $\rho \approx (T_o + B)/(1/\lambda_{avg} + T_a)$ . (13)

With the offered burst load of  $O = \rho \sum_{k=0}^M (M - K) \times P_k$  and the carried burst load of  $C = \sum_{k=0}^K k \times P_k$ , the stationary burst blocking probability  $P_{burst}$  is given by

$$P_{burst} = (O - C)/O \tag{13}$$

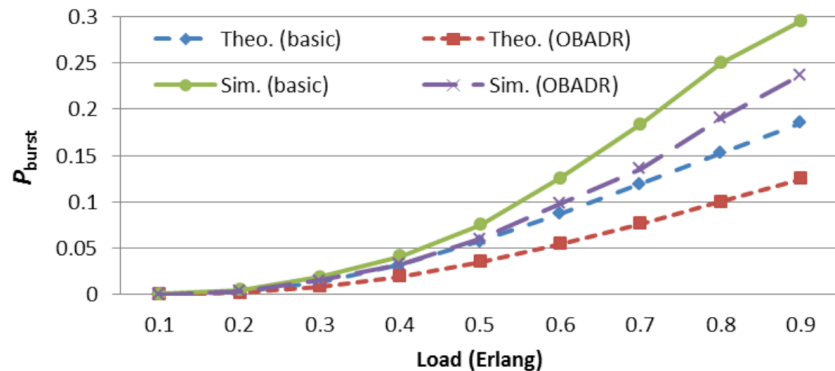
Eq. (13) is the blocking probability of the basic assembly model with a buffering delay of  $T_a + T_o$  (Figure 3a). However, with the OBADR model, the offset time of  $T_o$  is included in the assembly period of  $T_a$ , and so the buffering delay is reduced to  $T_a$ . Eq. (12) is reduced as follows:

$$B = \frac{1}{\mu} + \left(\frac{\lambda_{avg}}{\mu}\right) \times T_a \tag{14}$$

The OBADR model noticeably decreases the mean busy period  $B$  and thus decreases the blocking probability  $P_{burst}$ .

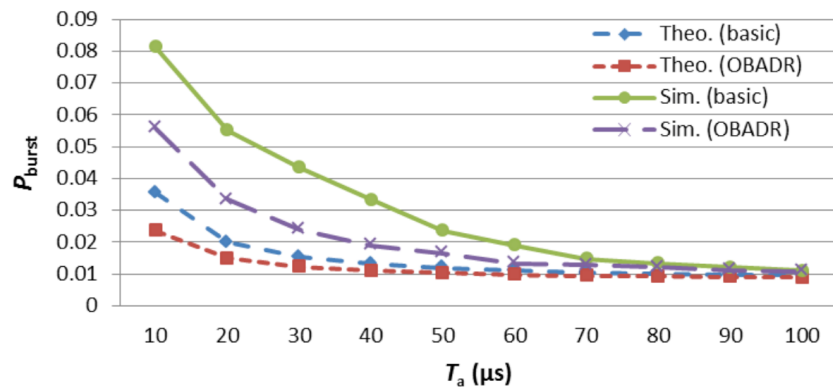
### 4.2. Numerical evaluations

We consider an ingress OBS node with 12 queues and 8 channels at outgoing fiber. The average length of the packets arriving in queues is 1000 bytes. The assembly time ( $T_a$ ) and the offset time ( $T_o$ ) are 0.6  $\mu s$  and 0.2  $\mu s$ , respectively. With the loads varying from 0.1 to 0.9 Erlang, Figure 9 shows a comparison of the blocking probability  $P_{burst}$  between the OBADR model and the basic assembly model, in which the OBADR model has lower  $P_{burst}$  in both theory (plotted by Mathematica) and simulation (implemented in NS2).



**Figure 9.** Comparison between the OBADR model and the basic assembly model in theory and simulation with various loads (Erlang).

With the assembly time varying from 10 to 100  $\mu s$ , Figure 10 shows a comparison of  $P_{burst}$  between the OBADR model and the traditional model in theory and simulation. In this comparison, the OBADR model again has lower  $P_{burst}$  in both theory and simulation.



**Figure 10.** Comparison between the OBADR model and the basic assembly model in theory and simulation with various  $T_a$  ( $\mu s$ ).

## 5. Conclusion

Sending the control packet early, before the burst completion, will reduce the buffering delay at ingress OBS nodes and thus decrease the delay of burst delivery. However, the completed burst length must be estimated at the moment of sending the control packet. In this paper, we have proposed an OBADR algorithm, which eliminates the offset time from the buffering delay and minimizes the estimation error by the burst assembly, based on the estimated length threshold. Additionally, our proposal decreases the blocking probability of scheduling at outgoing fiber. However, with the service differentiation based on the offset time, the approach of including offset time  $T_o$  in the assembly time  $T_a$  will limit the amount of separable services because of the condition of  $T_o < T_a$ . Therefore, it is necessary to research a further improvement of OBADR for OBS networks with the service differentiation.

## References

- [1] Qiao C, Yoo M. Optical Burst Switching (OBS)—A new paradigm for an optical internet. *J High Speed Netw* 1999; 8: 69-84.
- [2] Ge A, Callegati F, Tamil L. On optical burst switching and self-similar traffic. *IEEE Commun Lett* 2000; 4: 98-100.
- [3] Oh S, Kang M. A burst assembly algorithm in optical burst switching networks. In: *IEEE 2002 Optical Fiber Communication Conference and Exhibit*; 17–22 March 2002; Anaheim, CA, USA. New York, NY, USA: IEEE. pp. 771-773.
- [4] 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.
- [5] Chi Y, Junbin H, Zhengbin L, Anshi X. A novel burst assembly algorithm for OBS networks based on data-length time-lag product. In: *IEEE 2005 Asia-Paci?c Conference on Communication*; 5 October 2005; Perth, WA, Australia. New York, NY, USA: IEEE. pp. 319-323.
- [6] 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, VIC, Australia. Washington, DC, USA: OSA. pp. 664-666.
- [7] Sui Z, Zeng Q, Xiao S. An offset differential assembly method at the edge of OBS network. *Proc SPIE* 2005; 6021: 1-6.
- [8] 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.

- [9] 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.
- [10] Liu H, Jiang S. A mixed-length and time threshold burst assembly algorithm based on traf?c prediction in OBS network. *Int J Sens Comput* 2012; 2: 87-93.
- [11] Jiang X, Zhu N, Yuan L. A novel burst assembly algorithm for OBS networks based on burst size and assembly time prediction. *J Comput Inf Syst* 2013; 9: 463-475.
- [12] Peterson LL, Davie BS. *Computer Networks: a Systems Approach*. 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2003.
- [13] Salad K, Haidari F. On the performance of a simple packet rate estimator. In: *IEEE/ACS 2008 International Conference on Computer Systems and Applications*; 31 March–4 April 2008; Doha, Qatar. New York, NY, USA: IEEE. pp. 392-395.
- [14] Zalesky A, Wong EWM, Zukerman M, Le VH, Tucker RS. Performance analysis of an OBS edge router. *IEEE Photonic Tech L* 2004; 16: 695-698.