Short Article

Group Scheduling for MultiChannel in OBS Networks

Nguyen Hong Quoc[1], Vo Viet Minh Nhat[1], Nguyen Hoang Son[2]

[1] Department of Computer Science, Hue University, Hue, Vietnam
[2] Department of Mathematic, Hue University, Hue, Vietnam

Correspondence: Nguyen Hong Quoc, nhquoc@hueuni.edu.vn

Manuscript communication: received 20 November 2013, accepted 19 May 2014

Abstract– **Group scheduling is a scheduling operation of optical burst switching networks in which the burst header packets arriving in each timeslot will schedule their following bursts simultaneously. There have been many proposals for group scheduling (such as OBS-GS, MWIS-OS and LGS), but they consider mainly to schedule the arriving bursts which have the same wavelength on an output data channel. Another suggestion is GreedyOPT which considers the group scheduling for multichannel with the support of full wavelength converters, but it is not optimal. This article proposes another approach of group scheduling which is more optimal and has a linear complexity.**

Keywords– **OBS, wavelength conversion, group scheduling, multichannel, linear complexity**

1 Introduction

Optical burst switching (OBS) is considered as an effective alternative of optical packet switching, when there is a requirement of changing from optical circuit-switching to optical packet-switching to exploit the bandwidth of the optical fibers better. Moreover, due to the limitations of current optical technologies which cannot produce optical buffers (like RAM) and optical packet switches at nanosecond, OBS is then the most viable model of packet switching in the optical environment [1].

OBS networks have a typical characteristic that a burst header packet (BHP) is sent on a dedicated channel, called the control channel, to perform resource reservation; after an offset-time, its data burst follows on a separate data channel (see Figure 1). Due to the resources reserved by its BHP, the following burst will not incur any delay at each intermediate node; then it does not need the optical buffers. The resource reservation at each intermediate node is a part of scheduling operation [1].

There are many scheduling algorithms having been proposed, and they can be classified into two main categories: online and offline (in batch) algorithms. For the first ones, each arriving BHP calls a scheduling algorithm to reserve immediately the needed resources for its following burst. The typical representatives of this kind of scheduling algorithms are LAUC [2] and LAUC-VF [3] In the case of group scheduling; the BHPs arriving in each timeslot will schedule their following bursts simultaneously. OBS-GS [4], MWIS-OS [5] and LGS [6] are the proposals for group scheduling.

Figure 1. Typical model of resource reservationin OBS networks

In LAUC, a value of latest available unscheduled time (LAUT) of an output channel is used to compare with the arrival time of a burst; a channel is selected for scheduling the burst if the distance from its LAUT to the arrival time is smallest. LAUC-VF is an extension of LAUC, in which the gaps are created between the bursts scheduled on a channel are also considered in scheduling. Figure 2b describes the scheduling results of LAUC for the case of the arriving bursts as shown in Figure 2a.

In [4] and [5], the set of arriving bursts are mapped into an interval graph in which each burst corresponding to a vertex and two vertices are adjacent (having a connecting edge) if the corresponding bursts overlap together. In the case of MWIS-OS, each vertex is added one more weight which represent the burst length. The problem of group scheduling optimization will then be converted into finding a largest independent set (with MWIS-OS as the largest weighted independent set) in the interval graph. With that approach, the scheduling results of OBS-GS are the bursts of b_2, b_4, b_7 and b_8 (Figure 2c), while the scheduled bursts with MWIS-OS are b_2, b_3 and b_6 (Figure 2d). However, the complexity of this approach is NP-complete [7].

LGS also has the same objective as MWIS-OS to find the maximum total size of scheduled bursts, but it is modeled in the form of dynamic programming, in which each burst is mapped to a vertex of a graph and two vertices are adjacent if a latter burst can be scheduled with the nearest former burst without being

1859-378X–2013-3407 © 2013 REV

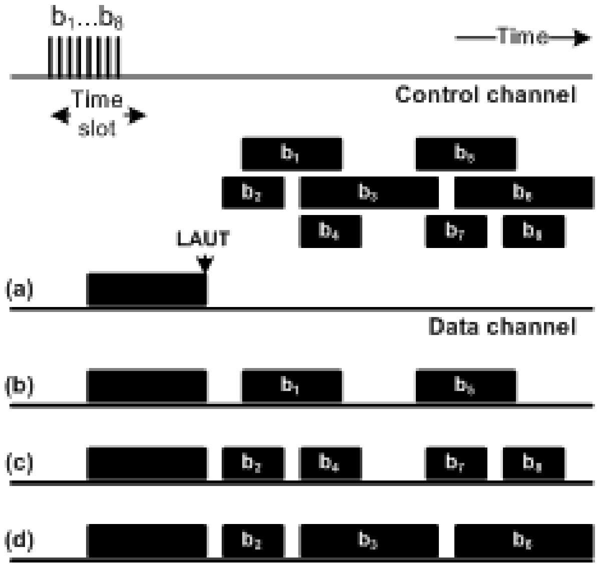


Figure 2. An example of the arriving bursts and the possible solutions of scheduling: (b) LAUC, (c) OBS-GS and (d) MWIS-OS

overlapped. The group scheduling of LGS also becomes the problem of finding the longest path in this graph. However, the complexity of LGS which is proven in [5] is only linear.

However, all the three algorithms of group scheduling only consider the scheduling on an output channel, with the assumption that all arriving bursts in each timeslot have the same wavelength. A group scheduling algorithm for multichannel with the support of full wavelength converters has also been proposed in [8], called GreedyOPT, in which the analyses show that it has the linear complexity ($O(mn)$) and an effective rate of lost data, but it is still not optimal due to the principle of first-fit. In this paper, we propose a group scheduling algorithm for multichannel with the support of full wavelength converters, which is more optimal based on the byte loss probability and the algorithm complexity.

The structure of the paper is as follows: section 3 presents our proposal of group scheduling algorithm for multichannel with the support of full wavelength converters. The simulation results will be analyzed in section 4 and section 5 shows the conclusion.

2 Our Approach of Group Scheduling Algorithm for Multichannel

Consider a set of arriving BHPs (corresponding to their following bursts) $I = \{b_1, b_2, \ldots, b_n\}$, where n is the total of BHPs. Assuming that the set of available channels at an output port is $W = \{1, 2, \ldots, m\}$, where m is the number of output channels. The information carried on each arriving BHP includes the start (s_i) and end time (e_i) of the burst b_i $(i = 1, 2, .., n)$. An arriving burst will be scheduled on an output channel if $s_i \geq LAUT$. Two bursts will not be scheduled on the same output channel if they are overlapped, i.e. $s_i < s_j < e_i$ or $s_j < s_i < e_j$.

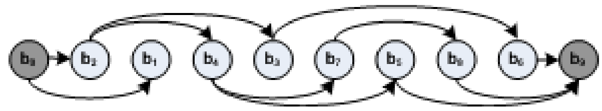


Figure 3. Group scheduling problem is formulated in the problem of dynamic programming

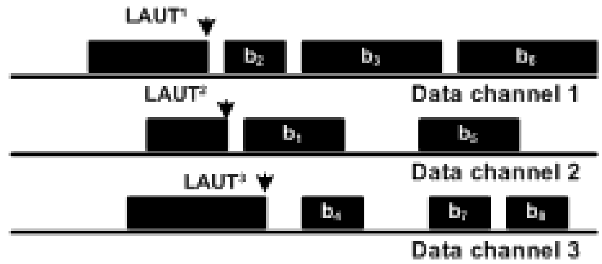


Figure 4. A scheduling solution of LGS-MC with the case of arriving bursts in Figure 2a

According to the approach of LGS, the set of arriving bursts, as depicted in Figure 2a, is be mapped into the graph as shown in Figure 3, after rearranging them in ascending of their end time. Each edge between two vertices carries an accumulated weight of the scheduled bursts. For LGS, the weight of an edge is the total accumulated length of the previous scheduled bursts plus the length of the next scheduled burst. Note that we add two vertices b_0 and b_9 (non overlapping with any other burst) so that the end time of b_0 is less than the start time of all other bursts and the start time of b_9 much be greater than the end time of all other bursts.

Figure 3 describes the scheduling problem of the bursts in Figure 2a which is converted to a problem of dynamic programming, in which the optimization of group scheduling of the arriving bursts on an output channel of LGS becomes the problem of finding the path from b_{n+1} to b_0 so that the accumulated weight is the largest. With LGS-MC, the optimization of group scheduling of the arriving bursts on m output channels becomes m' ($m' \leq m$) times of running the LGS algorithm, in which after each execution, the identified bursts that create an optimal combination of group scheduling will be deleted from the set of arriving bursts. This process is repeated until there is no burst in the set of arriving bursts or no output channel available any longer. In the case of the arriving bursts in Figure 2a, assume that there are 3 available output channels; the scheduling results of LGS-MC are now as shown in Figure 4.

Note that the output channels should be arranged in ascending of their LAUT to ensure that no burst can be scheduled optimally in a later output channel but it was removed from the list of arriving bursts in a previous step of scheduling by the violated condition of $s_i < LAUT$.

2.1 Description of the LGS-MC Algorithm

Input: $I = \{b_1, b_2, \ldots, b_n\}, W = \{1, 2, \ldots, m\}$ where $n, m \in Z^+$.

Output: The set of scheduled bursts I'.

Method:

Step 1. Arrange a set of output channels W in ascending of their $LAUT^w$ $(w = 1, 2, \ldots, m)$.

Step 2. For each channel $w \in W$, perform the following steps:

Step 2.1. Remove the bursts which have the arriving time less than $LAUT^w$ of the current channel ($s_i < LAUT^w$): $I = I/\{b_i\}$; $n = |I|$.

Step 2.2. Arrange the bursts of I in ascending of their end time (e_i).

Step 2.3. Create the list of $index(j)$ $(j = 1, 2, \ldots, n)$, where the value of $index(j)$ is the index of the previous burst which does not overlap burst j. If no such burst exists, set $index(j) = 0$.

Step 2.4. Determine the maximal total length $C(j)$ when the burst j is considered, which is calculated by the following equation:

$$\begin{cases} 0 & \text{if } j = 0 \\ Max\{C(j-1), l_j + C(index(j))\} & \text{if } j > 0 \end{cases}$$

where l_j is the length of the burst j ($l_j = e_j - s_j$) and the function $Max()$ returns the maximal total length if the burst j is scheduled or the previous value of $C(j-1)$.

Step 2.5. Trace the set of consecutive scheduled bursts which has the maximal total length.

Step 2.5.1. Set $j = n$ and $cost = C(n)$.

Step 2.5.2. While $(j > 0)$ do

- if $cost = C(j-1)$ then $j = j - 1$.
- if not, schedule the burst j, set$j = index(j)$; $cost = C(j)$; $I' = I' \cup \{b_j\}$ and $I = I \setminus \{b_j\}$.

Step 3. If there exists an unscheduled burst ($I \neq \emptyset$) and an available output channel w ($w \leq m$), then return to Step 2 with $w = w + 1$ and $n = |I|$.

Note that Step 2 of this algorithm is the LGS algorithm in [6]. It seems that our algorithm gives a set of scheduled bursts I'.

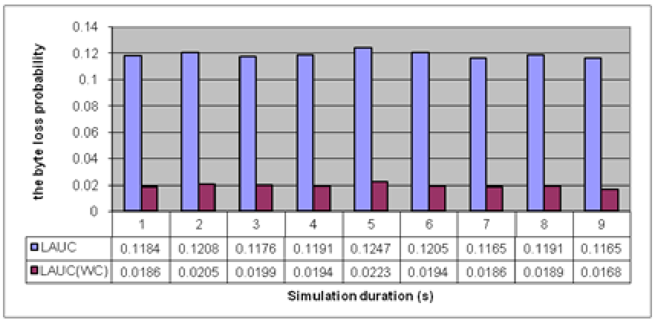
2.2 The Complexity of LGS-MC Algorithm

The complexity of LGS-MC algorithm is determined as follows:

- Step 1: arrange the output channels in ascending of their LAUT having the complexity of $O(mlog_2(m))$.
- Step 2: finding a set of scheduled bursts on an output data channel by using the LGS algorithm which has the complexity of $O(nlog_2(n))$ [6].
- Step 3: the complexity is $O(m)$. Here, Step 2 and Step 3 should be performed alternatively, then their complexity is $O(mnlog_2(n))$; therefore, the complexity of all the algorithm is $O(mnlog_2(n))$. However, because m is a constant, the final complexity is $O(nlog_2(n))$.

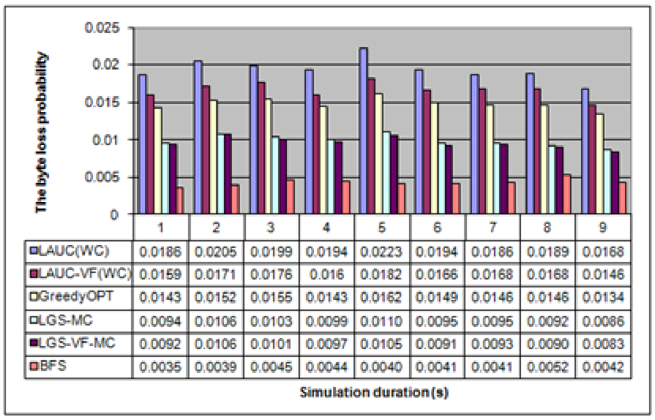
3 Simulation and Analysis

We implement the LGS-MC algorithm (and also the case of void filling: LGS-VF-MC) and compare (based on the lost bytes) with two online scheduling algorithms of LAUC, LAUC-VF with the support of full wavelength converters (LAUC(WC) and LAUC-VF(WC)) and also with the group scheduling algorithm for multi-channel of GreedyOPT. Assume that the data arriving at a core node have the Poisson distribution and the arriving bursts have varied lengths. The simulation time is from 1 to 9 seconds and the timeslot $\tau = 0.0015ms$.



	1	2	3	4	5	6	7	8	9
LAUC	0.1184	0.1208	0.1176	0.1191	0.1247	0.1205	0.1165	0.1191	0.1165
LAUC(WC)	0.0186	0.0205	0.0199	0.0194	0.0223	0.0194	0.0186	0.0189	0.0168

Figure 5. A comparison between LAUC and LAUC(WC) based on the byte loss probability



	1	2	3	4	5	6	7	8	9
LAUC(WC)	0.0186	0.0205	0.0199	0.0194	0.0223	0.0194	0.0186	0.0189	0.0168
LAUC-VF(WC)	0.0159	0.0171	0.0176	0.016	0.0182	0.0166	0.0168	0.0168	0.0146
GreedyOPT	0.0143	0.0152	0.0155	0.0143	0.0162	0.0149	0.0146	0.0146	0.0134
LGS-MC	0.0094	0.0106	0.0103	0.0099	0.0110	0.0095	0.0095	0.0092	0.0086
LGS-VF-MC	0.0092	0.0106	0.0101	0.0097	0.0105	0.0091	0.0093	0.0090	0.0083
BFS	0.0035	0.0039	0.0045	0.0044	0.0040	0.0041	0.0041	0.0052	0.0042

Figure 6. A comparison among LAUC(WC), LAUC-VF(WC), GreedyOPT, LGS-MC and BFS based on the byte loss probability

Figure 5 shows that, with the online scheduling, the case with wavelength conversion (LAUC(WC)) always gives better results than the one without wavelength conversion (LAUC). It is shown by the byte loss probability being less than about 10%. This is obvious because when an arriving burst cannot be scheduled on the desired output channel, it can be scheduled on a different available output channel with the support of wavelength converters.

In the case of group scheduling with the support of full wavelength converters, the simulation results in Figure 6 show that our algorithm of LGS-MC (and also the case with void filling of LGS-VF-MC) has better scheduling results in comparison with the online algorithms (LAUC(WC) and LAUC-VF(WC)) and GreedyOPT.

However, when compared with the algorithm of brute force search (BFS), the simulation results show that LGS-MC (and also LGS-VF-MC) is only nearly optimal. Indeed, as described in Section 3.1, LGS-MC is just a crude combination of optimal single cases; the difference is not significant (0.07% in average). Furthermore, the complexity of LGS-MC algorithm is only

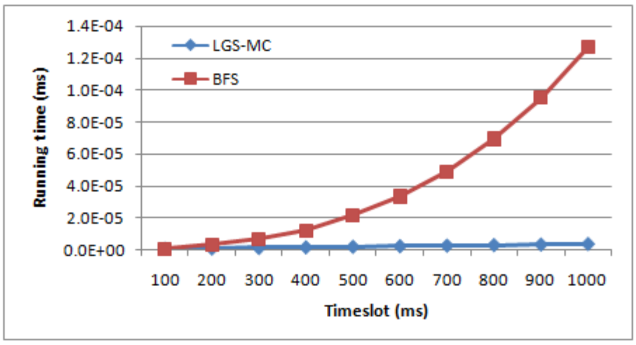


Figure 7. A comparison of the average execution time of LGS-MC and BFS when the timeslot size is changed

$O(nlog_2(n))$, while the BFS is $O(nm)$. This is shown quite clearly in Figure 7 when we measure their average execution time (ms). In short, a compromise between the optimum level and the algorithm complexity needs to be accepted.

4 Conclusion

In this study, we have proposed a group scheduling algorithm (LGS-MC) for multichannel with the support of wavelength converters. The analyses and simulation results show that our algorithm is more optimal than the other former proposals in the rate of lost data and the algorithm complexity. Despite not achieving the optimum level when being compared to BFS, the complexity of LGS-MC is quite small ($O(nlog_2(n))$), therefore, it is feasible to deploy in the optical environment.

References

[1] J. P. Jue and V. M. Vokkarane, *Optical Burst Switched networks*. Springer, 2005.

[2] J. S. Turner, "Terabit burst switching," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 3–16, 1999.

[3] Y. Xiong, M. Vandenhoute, and H. Cankaya, "Control architecture in optical burst-switched WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1838 – 1851, 2000.

[4] S. Charcranoon, T. El-Bawab, H. Cankaya, and J.-D. Shin, "Group-scheduling for optical burst switched (obs) networks," in *IEEE Global Telecommunications Conference, 2003. GLOBECOM '03.*, vol. 5, Dec 2003, pp. 2745 – 2749.

[5] H. Zheng and C. Chen, "Performance analysis of scheduling algorithms in optical burst switching (obs) networks," in *Second International Conference on Innovative Computing, Information and Control. ICICIC '07.*, Sept. 2007, p. 557.

[6] N. H. Quoc, V. V. M. Nhat, and N. H. Son, "An algorithm of group scheduling with void filling in obs core nodes," *Advanced in Computer Science and its Applications, Lecture Notes in Electrical Engineering*, no. 279, pp. 107–114, 2014.

[7] P. Ostergard, "A fast algorithm for the maximum clique problem," *Discrete Applied Mathematics*, vol. 120, pp. 197–207, 2002.

[8] G. Figueiredo and N. da Fonseca, "Algorithm with linear computational complexity for batch scheduling in obs networks," in *IEEE International Conference on Communications (ICC)*, 2011, pp. 1 – 6.