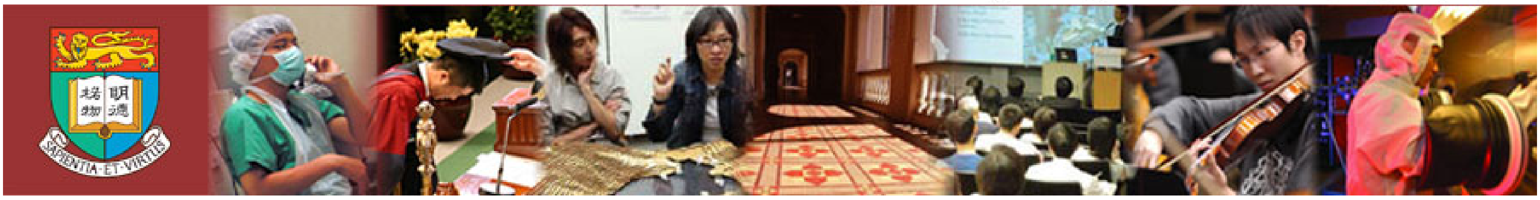

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

Improving the performance of optical burst switching with large control overhead

C. Y. Li,[1] P. K. A. Wai,[1] and Victor O. K. Li[2]
[1]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University,
Hong Kong, China, {enli, enwai}@polyu.edu.hk
[2]Department of Electrical and Electronic Engineering, The University of Hong Kong,
Hong Kong, China, vli@eee.hku.hk

Abstract—**In optical burst switching (OBS) system, the throughput decreases rapidly with increase in control packet processing time T_{cp}. The negative impact of T_{cp} will become significant as the optical fiber transmission rate increases. By analyzing the relationship between the throughput and T_{cp}, we attempt to improve the throughput. Different possible solutions are discussed. We found that using extra random offset time can significantly improve the throughput at the expense of increase in data burst delay.**

Index Terms—**optical burst switching, optical packet switching, control overhead, offset time**

I. INTRODUCTION

Without practical optical signal processing devices and optical buffers, all-optical packet switching is still in the early research stage [1]. Recently, optical burst switching (OBS) becomes popular because it can provide connectionless type of transmission services with low optical hardware complexity [2-6]. The traffic between OBS nodes are *data bursts*, each consisting of multiple packets. One-way reservation is used for the data transmission in OBS networks. When a data burst is generated at a node, a *control packet* is sent immediately to the destination of the data burst. The control packet reserves the resources at the nodes on the path of the data burst. No acknowledgment is sent back in order to minimize the delay time of sending out the data burst at the source. After an *offset time*, the node sends out the data burst following the same routing path of the control packet. If the reservation by the control packet is successful, the data burst will pass through all nodes on the path without any processing and optical-to-electrical (O/E) conversion. No optical buffers are required in the intermediate nodes, thus significantly simplifying the implementation of optical routing nodes.

One-way reservation is used in OBS networks as it is assumed that the average data burst transmission time can be much shorter than the propagation delay between nodes [2]. Also, the processing time T_{cp} of the control packets at intermediate nodes can be significant when compared to the transmission time of the data bursts. In these situations, feedback acknowledgment (two-way reservation) approach will be very inefficient. Using one-way reservation, the large propagation delay between nodes will no longer be a major concern for system performance. OBS networks can therefore perform better than the connection-oriented wavelength-routed optical networks.

Since the offset time T_{off} between the transmission of a data burst and its control packet must be larger than the total processing time of the control packet on the path, simple OBS channel reservation schemes such as Horizon [3] and just-in-time (JIT) [4] become inefficient in transmission bandwidth utilization when T_{cp} increases. Although sophisticated channel reservation schemes such as the last available unused channel with void filling (LAUC-VF) [5] scheme are more bandwidth efficient, and in principle can reuse all the idle transmission capacity in the time interval of T_{off}, the system throughput will still degrade significantly if T_{cp} is large. To eliminate the negative impact of control overhead, one may delay the data bursts at the inputs of intermediate nodes to compensate the control packet processing time but fiber delay lines (FDLs) are required [2], [3].

The control packet processing time T_{cp} will become increasingly significant because the data burst transmission time will decrease with the increase in the transmission rate of optical fiber, and one cannot lengthen the data bursts by too much. When trying to improve the system performance, it is important to preserve the low optical hardware complexity characteristics of OBS. In addition, a thorough understanding of the relationship between T_{cp} and the OBS performance is required. So far, control overhead is assumed to be small in OBS networks and is often neglected in the derivation of OBS schemes and performance evaluation. When T_{cp} is not set to zero, however, the focus is almost always on the whole offset time T_{off} and little attention has been given to the role of T_{cp} in the system performance.

In this paper, we study the relationship between T_{cp} and OBS performance and investigate means to improve OBS performance at large T_{cp}. In Section II, we review the degradation of OBS network performance when the control overhead increases. We observe that the system throughput degrades rapidly when the control packet processing time T_{cp} increases but it becomes insensitive to the increase of T_{cp} when T_{cp} is larger than the data burst transmission time L. In Section III, we study the

This research is supported by a grant from The Hong Kong Polytechnic University (Project Number 1-BBZB).

978-1-4244-2075-9/08/$25.00 ©2008 IEEE

role of T_{cp} in the throughput performance degradation. We then discuss methods that improve the throughput performance in Section III. In Section IV, we show that the throughput can be significantly increased by using the random extra offset but the data burst delay will also increase. We conclude in Section V.

II. Review of OBS Throughput Performance

Figure 1 shows the throughput performance of OBS in an 8 × 8 Manhattan Street Network (MSN) [7] with LAUC-VF channel reservation scheme and different control overhead. The OBS with Horizon [3] and JIT [4] channel reservation schemes are not included because these two schemes do not reuse the transmission bandwidth in the time interval of T_{off} between the data burst and control packet. It is therefore not surprising that their throughput performance decreases when T_{cp} increases, i.e., T_{off} increases. In contrast, LAUC-VF channel reservation scheme in principle can reuse all the idle transmission bandwidth. Intuitively, LAUC-VF should not be sensitive to T_{cp}. However, simulations often show that LAUC-VF throughput performance severely degrades if T_{cp} is large, especially in networks having paths with large number of hops.

Solid lines with crosses, circles, and squares in Fig. 1 represent the system throughputs when the control packet processing times are 0.1, 1.0 and 10 times of the data burst transmission time. For comparison, the dashed line corresponds to $T_{cp} = 0$. In the simulations, the offset time between a data burst and its control packet is given by

$$T_{\text{off}} = H \times T_{cp} + T_{sw}, \tag{1}$$

where H is the number of hops to the destination from the current location of the control packet [2]. Hence, H is equal to the total hop count of the path when the control packet is at the source, and decreases by one every time the control packet passes an intermediate node. The time T_{sw} is the required switch reconfiguration time at each node. Equation (1) assumes that a control packet leaves a node immediately after the node finds a channel for the corresponding data burst. There is no need to wait for the switch to reconfigure. Without special node architecture and switch reconfiguration algorithms, T_{sw} can also cause large throughput loss [6]. In this paper, we will focus on the impact of T_{cp} and set $T_{sw} = 0$ in all simulations. For ease of discussion, we normalized T_{cp} by the data burst transmission time in the following discussion. Other assumptions of the simulations are given in Section IV.

In Fig. 1, the OBS LAUC-VF with $T_{cp} = 0.0$ have a maximum throughput of 0.22 at a loading of 0.4 and reduces to 0.2 when the loading approaches one. On the other hand, OBS LAUC-VF with $T_{cp} = 0.1$ and 1.0 have the maximum throughputs of 0.18 and 0.16, respectively, at loadings of 0.3 and 0.2, and have a throughputs of 0.05 and 0.01, respectively, when the loading approaches one. Obviously, large T_{cp} has a severe negative impact on the OBS system throughput performance. However, when T_{cp} increases beyond 1.0, the throughput performance of LAUC-VF is similar to that of $T_{cp} = 1.0$ even if $T_{cp} >> 1$, e.g., $T_{cp} = 10.0$ in Fig. 1. The throughput of LAUC-VF appears to be insensitive to the value of T_{cp} when T_{cp} is larger than the data burst transmission time.

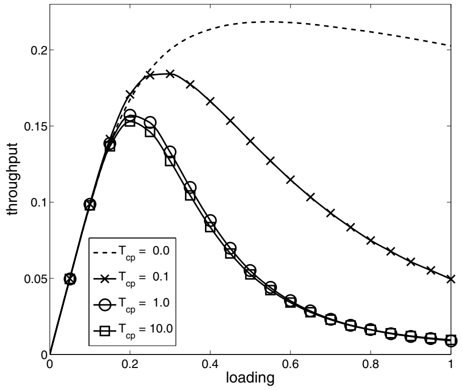

Fig. 1 The OBS throughput performance of an 8 × 8 Manhattan Street Network (MSN) with LAUC-VF for different control overheads. Solid lines with crosses, circles, and squares are the system throughputs when the control packet processing times are 0.1, 1.0 and 10 times of the data burst transmission time respectively. The dashed line represents the throughput for $T_{cp} = 0$. The setting and assumptions of the simulations are described in Section IV.

It seems that the results observed in Fig. 1 are due to the priority effect of the offset time T_{off} [8]. Assuming random control packet arrival, a data burst is more likely to be blocked by a data burst with a large T_{off} than by one with a small T_{off}. According to Eq. (1), data bursts with larger hop count paths will find it easier to reserve the output channels at nodes in the early stage of their journey, but they are also likely to be blocked when they are close to their destinations. Much transmission bandwidth is therefore wasted. A larger T_{cp} seems to enhance this offset time priority effect. Although this explanation seems valid for the throughput curves of $T_{cp} = 0.0$, 0.1 and 1.0, it is not sufficient to explain the curve of $T_{cp} = 10.0$ shown in Fig. 1. A more fundamental OBS performance model based on the control packet processing time T_{cp} is required.

III. The Model

Figure 2 shows two control packets CP_1 and CP_2 and their corresponding data bursts DB_1 and DB_2. We assume that the two control packets arrive at the input ports of a node N_j and request the same output port O_k. If there are multiple channels available at the output port, LAUC-VF chooses the idle time gap in the channels that can accommodate and has the start time closest to the arrival of the data burst [5]. For simplicity, we assume that there is only one wavelength channel per output port and all data bursts have constant transmission time L. The discussion can be easily generalized to multiple channel OBS with variable burst length. In Fig. 2, we assume that the arrival time of the control packets CP_1 and CP_2 are t_1 and t_2, respectively, where $t_1 \leq t_2$. The offset times associated with the control

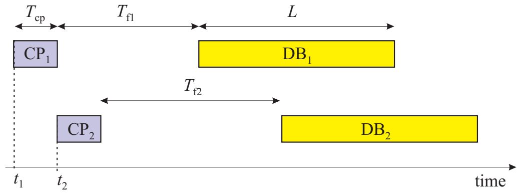

Fig. 2 Two control packets CP$_1$ and CP$_2$ and their corresponding data. bursts DB$_1$ and DB$_2$.

packets are T_{f1} and T_{f2} respectively. We assume a first-come-first-serve (FCFS) discipline. Since the control packet CP$_1$ arrives at the node earlier, CP$_1$ will block the channel request of CP$_2$ if t_1, t_2, T_{f1}, T_{f2}, and L satisfy

$$t_2 + T_{f2} - L < t_1 + T_{f1} < t_2 + T_{f2} + L. \quad (2)$$

Eq. (2) simply gives the condition that part of DB$_1$, either its head or tail, will overlap with DB$_2$ if the request of CP$_1$ is accepted. We consider the two cases (i) $T_{f1} \le T_{f2}$, and (ii) $T_{f1} > T_{f2}$ separately. In case (i), we can simplify Eq. (2) to

$$t_2 + (T_{f2} - T_{f1}) - L < t_1 \le t_2, \quad (3)$$

because $t_2 + (T_{f2} - T_{f1}) + L$ is larger than t_2 and $t_2 \ge t_1$. Since CP$_2$ (DB$_2$) will not be blocked if $T_{f2} > (T_{f1} + L)$, Eq. (3) explains the mechanism of OBS systems that support priority traffic classes with extra offset time: a high priority data burst will not be blocked by any low priority data burst if its offset time is larger than the sum of the data burst length and the maximum offset time of the low priority traffic. In case (ii), i.e., $T_{f1} > T_{f2}$, Eq. (2) can be rewritten as

$$t_2 - (T_{f1} - T_{f2}) - L < t_1 < \min[t_2, t_2 - (T_{f1} - T_{f2}) + L]. \quad (4)$$

From Eqs. (3) and (4), the length of the intervals in which CP$_1$ will block CP$_2$ (DB$_2$) are given as $\max[(L - |T_{f1} - T_{f2}|), 0]$ and $\min[(L + |T_{f1} - T_{f2}|), 2L]$, respectively. Please note that the time interval in which blocking will occur for case (ii) ($T_{f1} > T_{f2}$) is always larger than that of case (i) ($T_{f1} \le T_{f2}$). Also, adding a constant extra offset time to all data bursts will not affect the system throughput because the constant extra offset time does not change the results in Eqs. (3) and (4).

In order to study the role of T_{cp} in the throughput performance of OBS, we substitute Eq. (1) into Eqs. (3) and (4). The interval within which t_1 occurs such that CP$_1$ will block CP$_2$ (DB$_2$) becomes $\max[(L - |H_1 - H_2| \times T_{cp}), 0]$ for $H_1 \le H_2$ and $\min[(L + |H_1 - H_2| \times T_{cp}), 2L]$ for $H_1 > H_2$. Figure 3 plots the length of the interval within which t_1 occurs such that CP$_1$ will block CP$_2$ (DB$_2$) for different ($H_2 - H_1$) and T_{cp}. The blocking time interval and T_{cp} in Fig. 3 are normalized by the data burst transmission time L. The lines with pluses and asterisks are the time interval length of $T_{cp} = 0.1L$ and $0.2L$, respectively. The line with circles is that of $T_{cp} \ge L$. From Fig. 3, the blocking time interval is L when $H_2 = H_1$. For $H_2 > H_1$ the time interval decreases linearly and becomes zero when $(H_2 - H_1) \ge L/T_{cp}$. As T_{cp} increases, the range of ($H_2 - H_1$) for non-zero blocking time interval decreases rapidly and becomes zero for $T_{cp} > L$. For $H_2 < H_1$, the interval increases linearly and becomes $2L$ when $(H_2 - H_1) \le -L/T_{cp}$. As T_{cp} increases, the blocking time interval becomes $2L$ for all ($H_2 - H_1$) < 0 when $T_{cp} > L$. Figure 3 shows that a larger T_{cp} not only strengthens the offset time priority effect, but also increases the probability of a blocked data burst to be blocked by data bursts with larger hop counts.

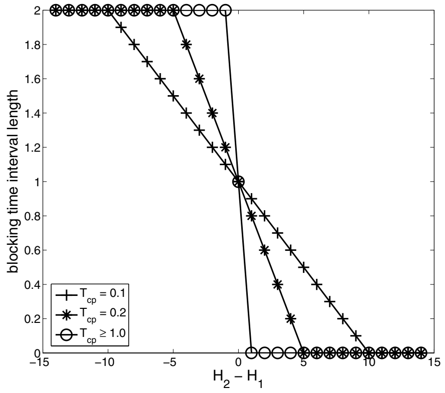

Fig. 3 The length of the blocking time interval for CP$_1$ to block CP$_2$ (DB$_2$) versus ($H_2 - H_1$) for different T_{cp}. The blocking time interval and T_{cp} are normalized by the data burst transmission time L.

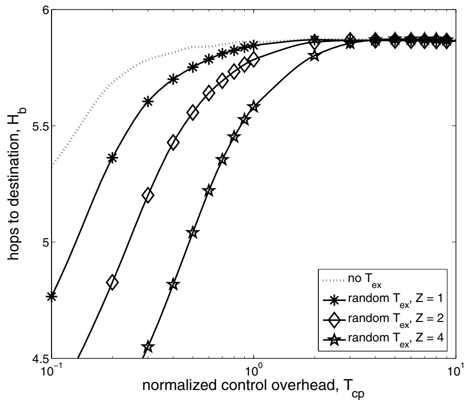

Fig. 4 H_b of an eight channel 8 × 8 MSN with OBS LAUC-VF using different control overhead T_{cp} when the normalized loading is one. The dotted line represents the simulated H_b of LAUC-VF without random extra offset time T_{ex}. Solid lines with asterisks, diamonds, and pentagrams are the H_b of LAUC-VF with randomized T_{ex} with Z = 1, 2, and 4, respectively.

We define H_b as the average number of hops to the destination of the data burst that blocks a data burst at a node. Figure 4 shows the H_b of an eight channel 8 × 8 MSN with OBS LAUC-VF using different control overhead T_{cp} when the normalized loading is one. In Fig. 4, the dotted curve is the simulated H_b of LAUC-VF with different T_{cp}. The value of H_b increases from 5.3 to 5.8 when T_{cp} increases from 0.1 to 1. Thus data bursts with larger H will have higher successful probability in channel reservation w T_{cp} increases. However, this also increases the bandwidth wastage when these data bursts are blocked. Note also that the OBS LAUC-VF throughput becomes insensitive to the change of T_{cp} when $T_{cp} > L$, e.g., the curve of $T_{\mathrm{cp}} = 10$ in Fig. 1. We have similar observations from Fig. 4. H_{b} of OBS LAUC-VF will have a value of around 5.8 even if we increase T_{cp} from 1 to 10.

A. Methods to improve OBS throughput

The performance model explains the reasons of OBS to have poor throughput performance as shown in Fig. 1 when T_{cp} is large. In the following, we briefly discuss different methods to improve the OBS throughput.

1) *Constant offset time*: Data bursts with different values of H will have the same probability to block each other if the offset time is set to constant. However, FDLs are required at the inputs of the nodes which will increase the complexity of OBS node architecture [2], [3].
2) *Dynamic offset time adjustment*: In principle, we can dynamically set the initial offset time between data bursts and control packets according to the network traffic condition to reduce the data burst discard rate. Similar to adaptive routing [2], the improvement in performance highly depends on the accuracy of the traffic distribution information. This method is not suitable for systems with highly dynamic traffic and large propagation delay between nodes.
3) *Window-based channel reservation*: We have investigated a window-based reservation scheme that does not require FDLs [9]. The proposed scheme delays the control packets in a node such that the channel assignment is better than that of the FCFS approach. Since additional delay at a node is equivalent to lengthening T_{cp}, this method is appropriate for the OBS system if the original T_{cp} is larger than the data burst transmission time.
4) *Randomized extra offset time*: Throughput improvement has been observed with an extra randomized offset time [10] and this is attributed to the traffic shaping effect of the data bursts at OBS source nodes. However, we find that the randomized extra offset time also significantly weakens the connection between H and the offset time, and hence reduces the blocking. The randomized extra offset time will be discussed in more detail in the next section.

B. Randomized extra offset time

When a randomized extra offset time is added to Eq. (1), the offset time becomes $T_{\mathrm{off}} = H \times T_{\mathrm{cp}} + T_{\mathrm{sw}} + T_{\mathrm{ex}}$, where T_{ex} is the random extra offset time. The difference of the two offset times T_{f2} and T_{f1} is

$$T_{\mathrm{f2}} - T_{\mathrm{f1}} = (H_2 - H_1)\, T_{\mathrm{cp}} + T_{\mathrm{diff}} \qquad (7)$$

where T_{diff} is the difference of the two random extra offset times. For $(H_2 - H_1)T_{\mathrm{cp}} > \ -T_{\mathrm{diff}}$, a non-zero time interval is obtained in Eq. (3) only if

$$-(H_2 - H_1)T_{\mathrm{cp}} < T_{\mathrm{diff}} < L - (H_2 - H_1)T_{\mathrm{cp}}. \qquad (8)$$

In contrast, Eq. (4) gives a non-zero time interval for any $T_{\mathrm{diff}} < -(H_2 - H_1)T_{\mathrm{cp}}$. As we assign T_{ex} at random, T_{diff} is independent of $(H_2 - H_1)$, and has the probability density function (pdf)

$$f_{\mathrm{diff}}(y) = \int_{-\infty}^{\infty} f_{\mathrm{ex}}(y + x) f_{\mathrm{ex}}(x)dx \qquad (9)$$

where $f_{\mathrm{ex}}(x)$ is the pdf of T_{ex}. For illustration convenience, we simply assume T_{ex} to be a uniform random variable defined on $(0, Z)$, i.e., $f_{\mathrm{ex}}(x) = 1 / Z$ for $0 < x < Z$ and zero otherwise. Then T_{diff} is a random variable defined on $(-Z, Z)$ with the pdf

$$f_{\mathrm{diff}}(y) = (Z - |\, y\,|) / Z^2 \quad \text{for} \quad -Z < y < Z, \qquad (10)$$

and zero otherwise. With $(H_2 - H_1)T_{\mathrm{cp}} = L$, a data burst of H_2 will not be blocked by any data burst of H_1 if the offset time is from Eq. (1), i.e., no non-zero time interval will be found from Eqs. (3) and (4). With randomized extra offset time, however, the probability of having a non-zero time interval is 0.5 if the pdf of T_{diff} is from Eq. (10) and $Z = L$. In Fig. 3, solid lines with asterisks, diamonds, and pentagrams are the H_{b} of OBS LAUC-VF with randomized extra offset time and $Z = 1$, 2, and 4, respectively, for the eight channel 8 × 8 MSN for different T_{cp}. It shows that H_{b} can be significantly reduced with large random extra offset times. An average delay of $Z / 2$ however is added to the data bursts.

IV. PERFORMANCE EVALUATION

Owing to the page length limitation, only the throughput improvement of randomized extra offset time approach is demonstrated in this paper. A detail comparison between different methods listed in Section III-A will be provided in a later paper. In this section, we use simulations to investigate the throughput of OBS LAUC-VF on an 8 × 8 Manhattan Street Network (MSN) [7] with different control overhead T_{cp} and extra random offset time T_{ex}. We focus on the system throughput performance since the average added delay is simply $Z / 2$ for the data bursts.

In the simulations, data bursts arrive as a Poisson process. When a new data burst arrives at a node, it randomly chooses a destination from the rest of the nodes in the network and uses minimum hop routing to determine the paths. The length of each data burst (transmission time) is one time unit, i.e., $L = 1$. To focus on the effect of random extra offset time, no traffic shaping (smoothing the control packet transmissions as in [10]) is applied at the source. After a new data burst is generated at a node, the control packet is sent out immediately to reserve the required wavelength channels on the path. The data burst is then transmitted after the sum of a time period from Eq. (1) (with $T_{\mathrm{sw}} = 0$) and a random extra offset time T_{ex}. In the simulations, the extra offset time is a random value in $(0, Z)$, where Z is 1, 10, and 100. If two or more wavelength channels are available, LAUC-VF scheme is used to choose a suitable one. The normalized load offered to a node is the ratio of the average data burst length to the inter-arrival time normalized by the number of channels per link. In the simulations, we assume eight channels per link and all nodes receive the same offered load. The propagation delay of a link is 10 time units. All simulations are

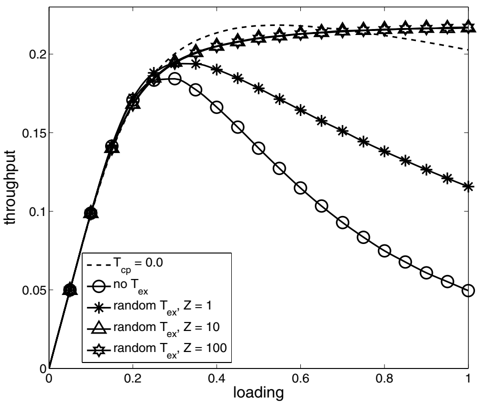

Fig. 5 The throughput of the LAUC-VF with different values of T_{ex} and Z. The control overhead T_{cp} = 0.1.

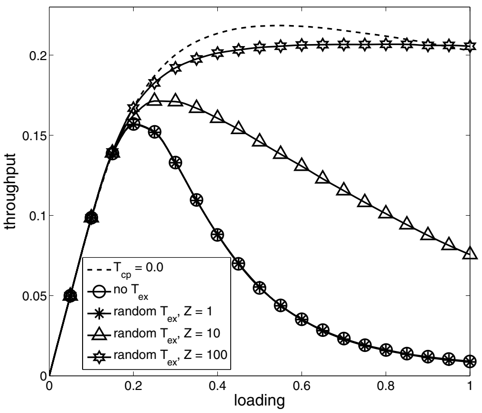

Fig. 6 The throughput of the LAUC-VF with different values of T_{ex} and Z. The control overhead T_{cp} = 1.0.

run sufficiently long such that the 95% confidence intervals are less than 1% of the average values of the results.

Figures 5 and 6 show the throughput of LAUC-VF with different values of T_{ex} and Z, and the control overhead T_{cp} = 0.1 and 1.0, respectively. In the figures, solid lines with circles, asterisks, triangles, and hexagrams are the throughput of LAUC-VF with no random T_{ex}, and that with random T_{ex} of Z = 1, 10 and 100, respectively. The dashed line is the throughput of LAUC-VF for T_{cp} = 0. From the figures, we observe that the random extra offset time approach can significantly reduce the negative impact of OBS control overhead if the added delay is not a concern. The throughput improvement however diminishes when Z increases. The throughput improves significantly when Z increases from a T_{cp} to $10T_{cp}$, but there is little additional improvement when Z increases from $10T_{cp}$ to a larger value.

With a transmission rate of 10 Gps and a data burst size of 100K bytes (L = 80 μsec), the average delays added to the data bursts are 40, 400, and 4000 μsec for Z = 1, 10 and 100, respectively. The randomized extra offset time approach is therefore appropriate for OBS networks with millisecond propagation delay between nodes such that the additional delay is negligible.

V. Conclusion

A large control packet processing time T_{cp} can severely degrade the throughput performance of optical burst switching (OBS) system. T_{cp} will become increasingly significant because the data burst transmission time will be shortened with the increase in optical fiber transmission rate. We study the relationship between control packet processing time T_{cp} and OBS throughput. We show that data bursts with large hop count path are more likely to successfully reserve the output channels at nodes in the early stage of their journey, but they are also likely to be blocked near the destination. Much transmission bandwidth will be wasted and system performance is degraded. Different methods have been discussed to reduce the impact of large T_{cp}. Among the methods, the random extra offset time is discussed in detail because of its simplicity. We show that the throughput can be significantly improved at the expense of an increase in the data burst delay.

References

[1] T. Houbavlis, et al., "All-optical signal processing and applications within the Esprit project DO_ALL," *Journal of Lightwave Technology*, Vol. 23, No. 2, pp. 781–801, 2005.

[2] C. Qiao, and M. Yoo, "Optical burst switching (OBS) - a new paradigm for an optical Internet," *Journal of High Speed Networks,* Vol. 8, pp. 69–84, 1999.

[3] J.S. Tuner, "Terabit burst switching," *Journal of High Speed Networks,* Vol. 8, pp. 3–16, 1999.

[4] J.Y. Wei, and R.I. McFarland, Jr., "Just-In-Time Signaling for WDM Optical Burst Switching Networks," *Journal of Lightwave Technology*, Vol. 18, No. 12, pp. 2019–2037, 2000.

[5] Yijun Xiong, M. Vandenhoute, and H.C. Cankaya, "Control architecture in optical burst-switched WDM networks," *IEEE Journal on Selected Areas in Communications,* Vol. 18, No. 10, pp. 1838–1851, 2000.

[6] C.Y. Li, G.M. Li, P.K.A. Wai, and V.O.K. Li, "Optical burst switching with large switching overhead," *IEEE Journal of Lightwave Technology*, Vol. 25, No. 2, pp. 451–462, February 2007.

[7] N.F. Maxemchuk, "Routing in Manhattan Street network," *IEEE Transactions on Communications*, Vol. 35, No. 5, pp. 503–512, 1987.

[8] B.C. Kim, Y.Z. Cho, and D. Montgomery, "An efficient optical burst switching technique for multi-hop networks," *IEICE Transactions on Communications*, Vol. E87-B, No. 6, pp. 1737–1740, 2004.

[9] C.Y. Li, G.M. Li, P.K.A. Wai, and V.O.K. Li, "Novel resource reservation schemes for optical burst switching," *Proceedings of IEEE International Conference on communications (ICC 2005)*, Vol. 3, pp. 1651–1655, Seoul, Korea, May 16-20, 2005.

[10] S. Verma, H. Chaskar, and R. Ravikanth, "Optical burst switching: a viable solution for terabit IP backbone," *IEEE Network*, Vol. 14, No. 6, pp. 48–53, 2000.