Performance Improvement Methods for Burst-Switched Networks

C. Y. Li, P. K. A. Wai, and Victor O.-K. Li

***Abstract*—In this paper, we present a performance model of optical burst switching (OBS) that can explain the degradation of OBS throughput performance when the control packet processing time increases. We then use the proposed performance model to investigate three feasible methods to improve OBS performance without significantly increasing the implementation complexity: addition of simple fiber delay lines (FDLs), random extra offset time, and window-based channel scheduling (WBS). Additional FDLs can eliminate the negative impact caused by the variation of the offset time between control packets and data bursts. The random extra offset time approach does not require any additional hardware and computational capability in the nodes. If higher computational capability is available, WBS in general can provide better throughput improvement than that of random extra offset time when FDLs are used in the nodes to compensate the processing time. Simulation results show that a combination of the proposed methods can significantly improve OBS performance.**

***Index Terms*—Control overhead; Offset time; Optical burst switching.**

I. Introduction

Recently, many in academic circles argued that, due to the lack of sophisticated optical hardware such as optical buffers, one-way reservation techniques such as optical burst switching (OBS) are likely candidates for the transmission of bursty traffic in optical networks in the near future. OBS can provide connectionless transmission services in optical networks without sophisticated optical hardware [1–5]. Data traffic discarded at immediate nodes is retransmitted by the sources. Thus the one-way resource reservation of OBS effectively reduces the hardware complexity and signal processing requirement [1–3]. In addition to implementation simplicity, for bursty traffic, OBS has better performance than the optical circuit switching (OCS) reservation scheme currently utilized in optical networks. Since the duration of transmission for bursty traffic is typically short, two-way or centralized resource reservation approaches will be inefficient if the data transmission time is not much larger than the propagation delay time between nodes. The performance of a one-way resource reservation scheme such as OBS, however, is not sensitive to the propagation delay between nodes [6]. OBS is therefore considered by some to be a long-term evolution direction of OCS.

Despite the advantages of one-way resource reservation, OBS occasionally has suboptimal channel reservation performance, and the system throughput is therefore low. Because an optical buffer is not available, an OBS node cannot resolve any output contention by temporarily storing the contending data bursts. The OBS node must also complete the routing and channel assignment computations of the transit data bursts before their arrivals. In general, only a first-come-first-served (FCFS) approach can be used to schedule the incoming data bursts. Data bursts in OBS networks with large hop number paths will commonly suffer from a larger loss rate. System throughput is reduced because more transmission bandwidth is required to retransmit the data bursts with large hop number paths. Moreover, it has been observed that further degradation of throughput performance will occur if OBS nodes require nonnegligible time (for example, 0.1 data burst transmission time) to process the channel assignment of data bursts, i.e., the offset time priority effect [7,8].

Many approaches have been proposed to improve the performance of OBS, for example, adding optical buffers (switchable fiber delay lines) to OBS nodes [9], burst segmentation [10], centralized control and two-way resource reservation [11], and dynamic routing [12]. However, many of these proposals are not practical because they inevitably require much more sophisticated implementation than the original OBS scheme. In order to improve OBS performance without significantly increasing the implementation complexity, it is necessary to delineate the relationship among control processing time, one-way resource reservation, and OBS performance such that the merits of different improvement approaches can be fully understood. With this understanding, different performance improvement methods can be combined to further improve the OBS performance.

In this paper, we study combining different methods to improve OBS throughput performance. The key requirement is that the performance improvement methods applied will not significantly increase OBS implementation complexity such that the resulting scheme is practical to implement. Our main contributions include the following:

- We show that the commonly used discarded-traffic-clear approach in OBS performance evaluations can easily overlook problems that occur only in a few paths of a network and lead to incorrect conclusions. Thus all

Manuscript received June 29, 2010; revised September 27, 2010; accepted October 4, 2010; published January 26, 2011 (Doc. ID 130879).

C. Y. Li (e-mail: enli@polyu.edu.hk) and P. K. A. Wai are with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, China.

Victor O.-K. Li is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China, and is a Visiting Professor in the Department of Computer Engineering, King Saud University, Saudi Arabia.

Digital Object Identifier 10.1364/JOCN.3.000104

performance evaluations in this paper use the discarded-traffic-retransmit approach (Section II).

- We derive what we believe to be the first performance model that can explain the phenomenon of OBS throughput degradation when the control packet processing time T_{cp} increases (Section III).
- We show that the compensation factor β in the fiber delay line (FDL) overcompensation approach is critical to OBS performance improvement, but the proper value of β is not easy to determine. We propose to use FDLs only for T_{cp} compensation. Further improvement in OBS performance is to be achieved by other methods (Subsection IV.A).
- We believe we are the first to analyze the performance improvement mechanism of the random extra offset time approach. Also, we believe we are the first to show that random extra offset time can improve system performance even if the OBS has zero T_{cp} or T_{cp} is fully compensated by FDLs (Subsection IV.B).
- We investigate the first window-based channel scheduling (WBS) that is suitable for OBS with both window time T_{wd} and T_{cp} compensated by FDLs. In traditional WBS, only T_{wd} is compensated. We present a novel channel assignment method for the proposed WBS. The proposed WBS can provide better performance improvement for OBS, with and without FDL compensation, than that of traditional WBS (Subsection IV.C).
- We show that the performance of OBS can be significantly improved by combining different methods such as FDL compensation plus random extra offset time and FDL compensation plus WBS (Section V).

In addition, we compare the performance of just-enough-time (JET), Horizon, and just-in-time (JIT) OBS using the discarded-traffic-retransmission approach in Section II. Also, we discussed the trade-off of using FDL overcompensation to improve OBS performance in Subsection IV.A.

II. Review of Throughput Degradation of Optical Burst Switching With Control Overhead

The traffic between OBS nodes is *data bursts*, each consisting of multiple data packets. When data packets arrive at an OBS node, data bursts are generated to carry the data packets to their destinations. For each newly generated data burst, a *control packet* is first sent to the destination of the data burst. The control packet reserves the resources at the intermediate 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 source node sends out the data burst following the same routing path of the control packet. The minimum offset time between the control packet and the data burst is

$$T_{\text{off}} = H \times T_{cp} + T_{sw}, \quad (1)$$

where T_{sw} is the required switch reconfiguration time at each node. T_{cp} is the processing time of a control packet in a node. H is the number of hops to the destination from the current location of the control packet [1]. Hence, H is equal to the total hop count of the path when the control packet is at the source and decreases by one for each intermediate node the control packet passes.

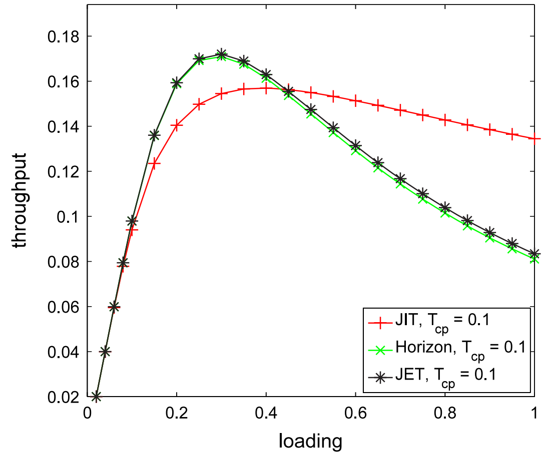

Fig. 1. (Color online) The loading–throughput curves of different OBS schemes in an 8×8 MSN [13] with $T_{cp} = 0.1$ (discarded-traffic-clear approach is used).

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 at the intermediate nodes. There are mainly three kinds of OBS. A JET OBS node can reserve the output channel only for the time period that a data burst passes [1]. Hence, the output channel is still available for other data bursts in the offset time period between the arrived control packet and the scheduled data burst. A Horizon OBS node, on the other hand, is not able to reuse the offset time period of the output channel [2]. A JIT OBS intermediate node further requires that the output channel is idle when the control packet arrives at the node [3].

A data burst will be discarded if it cannot find an appropriate output channel when it arrives at an intermediate node. Unlike common OBS performance evaluations using the discarded-traffic-clear approach [1–5], we assume that all discarded data bursts are retransmitted by their sources until they reach their destinations. It is important to note that the discarded-traffic-clear approach can easily overlook problems that may occur only in a few paths of the network. In Fig. 1, the discarded-traffic-clear approach is used for OBS in an 8×8 Manhattan Street Network (MSN) [13]. The assumptions behind the simulation can be found in Section V. Figure 1 shows that JIT has better performance than that of JET and Horizon OBS when the loading is beyond 0.44. A detailed analysis shows that the average path length of JIT received traffic is much shorter than that of JET and Horizon OBS in such cases. Thus the discarded-traffic-clear approach may lead to erroneous conclusions. Such problems can be avoided by using the discarded-traffic-retransmit approach. The performance of all data bursts will be accurately represented by the average throughput–delay curves, irrespective of the size and path length of the data bursts. For example, Fig. 2 shows the throughput–delay performance of OBS in the 8×8 MSN using the discarded-traffic-retransmit approach. From Fig. 2, it

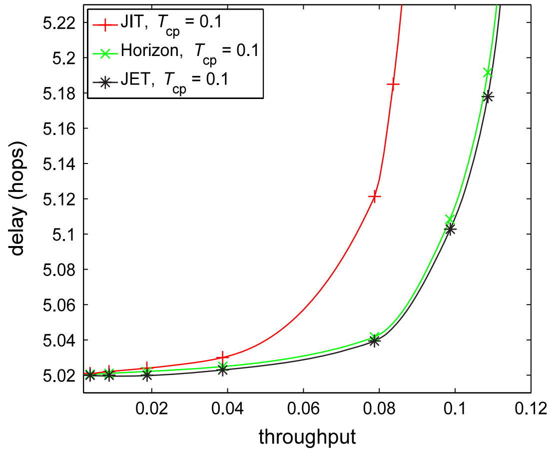

Fig. 2. (Color online) The throughtput–delay curves of different OBS schemes of an 8×8 MSN with $T_{cp} = 0.1$ (discarded-traffic-retransmit approach is used).

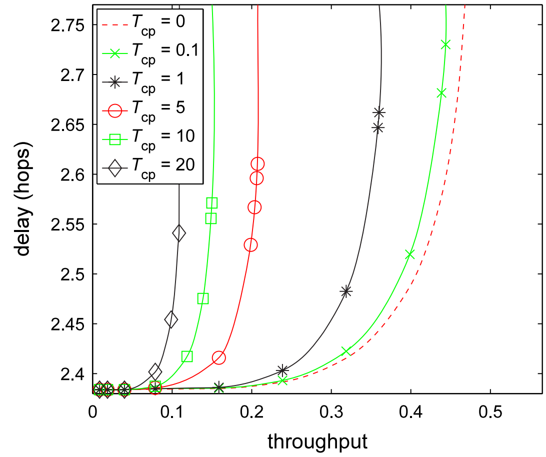

Fig. 4. (Color online) The Horizon OBS throughput–delay performance of an NSFNet with different control overheads.

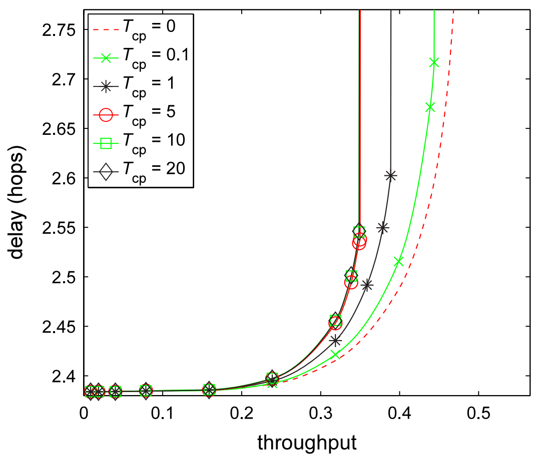

Fig. 3. (Color online) The JET OBS throughput–delay performance of an NSFNet topology network with different control overheads.

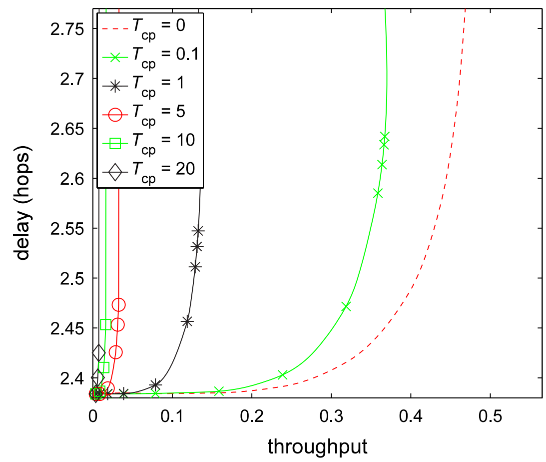

Fig. 5. (Color online) The JIT OBS throughput–delay performance of an NSFNet with different control overheads.

is obvious that JIT will never be better than JET and Horizon OBS.

Although the performance comparison of JET, Horizon, and JIT OBS has been provided in many studies [1–5], none of them were done using the discarded-traffic-retransmit approach. The accuracy of their conclusions is therefore in question. Figures 3–5 show the average throughput–delay performance of JET, Horizon, and JIT OBS in an NSFNet topology network (Fig. 6) with different control overheads. We use the total number of hops (including the retransmissions) taken by the data bursts as the delays. The solid curves with crosses, asterisks, circles, squares, and diamonds in Figs. 3–5 represent the throughput–delay performance of the OBS systems when the control packet processing times T_{cp} are 0.1, 1, 5, 10, and 20 times the data burst transmission time, respectively. We plot the cases for $T_{cp} = 0$ (dashed curve) for comparison. We observe that JET, Horizon, and JIT OBS have similar throughput–delay performance when the control overhead is negligible. Among the three OBS systems, the throughput–delay performance of JIT is the most sensitive to T_{cp} (Fig. 5). Although the throughput of Horizon OBS is also reduced with the increase of T_{cp}, it does not decrease as rapidly as that of JIT, as shown in Fig. 4. From Fig. 3, we observe that the throughput of JET also decreases with the increase of T_{cp} if T_{cp} is not large. When T_{cp} is large, e.g., larger than one time unit, however, the throughput–delay performance of JET is insensitive to the increase of T_{cp}, e.g., the curves of $T_{cp} = 5$, 10, and 20 in Fig. 3 are nearly identical despite the large differences in T_{cp}.

The priority effect of the offset time T_{off} is often used to explain the throughput degradation of JET OBS with T_{cp} [7,8]. Since T_{off} is proportional to the number of hops to the destination, 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 more 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 appears to be valid for the throughput–delay curves of $T_{cp} = 0.0$, 0.1, and 1.0, it is not sufficient to explain the curves for $T_{cp} = 5$, 10, and 20 shown in Fig. 3.

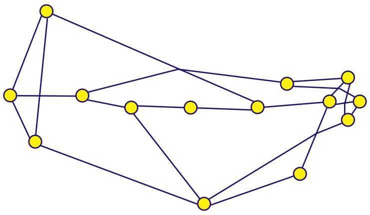

Fig. 6. (Color online) The NSFNet (1991) network topology. The original map of the network is available from the Internet (ftp://ftp.uu.net/inet/maps/nsfnet/).

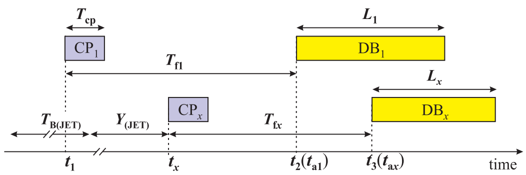

Fig. 7. (Color online) Two control packets CP_1 and CP_x and their corresponding data bursts DB_1 and DB_x.

III. The Model

Equation (1) has been used in a few performance models to determine some OBS features, for example, the OBS offset time priority effect [7,8], and the range of T_{cp} in which Horizon will have similar performance as that of JET [14]. To simplify computations, however, most OBS performance models do not use Eq. (1) for T_{off} but assume T_{off} to be independent of H and T_{cp} [15–17]. Obviously, these performance models are only valid for OBS systems with negligible T_{cp}. Thus we need to derive an OBS model that takes into account the impact of T_{cp} on system performance.

Figure 7 shows two control packets CP_1 and CP_x and their corresponding data bursts DB_1 and DB_x. 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. For simplicity, we assume that there is only one wavelength channel per output port and the data bursts DB_1 and DB_x have transmission times L_1 and L_x, respectively. In Fig. 7, we assume that the arrival times of the control packets CP_1 and CP_x are t_1 and t_x, respectively, where $t_1 < t_x$. The offset times associated with the control packets are T_{f1} and T_{fx}, respectively. Hence, data bursts DB_1 and DB_x arrive at the node at $t_{a1} = t_1 + T_{f1}$ and $t_{ax} = t_x + T_{fx}$, respectively, e.g., $t_2 = t_{a1}$ and $t_3 = t_{ax}$ in Fig. 7. Note that data burst DB_1 can arrive at the node later than that of DB_x ($t_{a1} > t_{ax}$) though Fig. 7 only shows the case of $t_{a1} < t_{ax}$. We assume FCFS. In Fig. 7, $T_{B(\text{JET})}$ is defined as the length of the time interval in which the arrival of data burst DB_1 will block the data burst DB_x. $Y_{(\text{JET})}$ is the distance between $T_{B(\text{JET})}$ and t_x. Since the control packet CP_1 arrives at the node earlier, CP_1 will block the channel request of CP_x if data burst DB_x will overlap any part of DB_1. We can therefore write the criteria of data burst DB_x being blocked by DB_1 (CP_1) using t_1, t_x, T_{f1}, T_{fx}, L_1, and L_x according to the two cases (i) $t_{a1} < t_{ax}$ and (ii) $t_{a1} > t_{ax}$ as

$$\text{(i)}: t_x + T_{fx} - L_1 < t_1 + T_{f1}, \tag{2a}$$

$$\text{(ii)}: t_x + T_{fx} + L_x > t_1 + T_{f1}. \tag{2b}$$

Equation (2a) is the criterion of the request of CP_x being blocked if data burst DB_1 arrives earlier than DB_x, whereas Eq. (2b) is for DB_1 arriving at the node later than DB_x. According to the relationship between t_1, t_x, t_{a1}, and t_{ax}, we can further derive and simplify the blocking criterion of Case (i) into

$$t_x + (T_{fx} - T_{f1}) - L_1 < t_1 < t_x + (T_{fx} - T_{f1}). \tag{3}$$

In Eq. (3), the inequality $t_1 < t_x + (T_{fx} - T_{f1})$ is derived directly from the assumption of $t_{a1} < t_{ax}$. Similarly, for Case (ii) we have

$$t_x + (T_{fx} - T_{f1}) < t_1 < t_x + (T_{fx} - T_{f1}) + L_x. \tag{4}$$

Owing to $t_1 < t_x$, Eq. (3) implies T_{fx} must be smaller than $T_{f1} + L_1$ for blocking to occur if $t_{a1} < t_{ax}$. Similarly, Eq. (4) implies T_{fx} must be smaller than T_{f1} if $t_{a1} > t_{ax}$. In Eq. (3), the length of the time interval T_{b1} for t_1 in which CP_1 will block CP_x (DB_x) is L_1 if T_{fx} is smaller than T_{f1}, and it decreases to zero when we increase T_{fx} to $T_{f1} + L_1$. Hence, $T_{b1} = L_1 - \max(T_{fx} - T_{f1}, 0)$ for $T_{fx} \le T_{f1} + L_1$, and $T_{b1} = 0$ otherwise. Similarly, the length of the time interval T_{b2} for t_1 of Eq. (4) is $\min(T_{f1} - T_{fx}, L_x)$ for $T_{fx} < T_{f1}$, and $T_{b2} = 0$ otherwise. It is because T_{b2} will be equal to $T_{f1} - T_{fx}$ if $0 \le T_{f1} - T_{fx} \le L_x$, and T_{b2} will be equal to L_x if $T_{f1} > T_{fx} + L_x$. The time intervals T_{b1} and T_{b2} do not overlap each other for a given set of T_{f1}, T_{fx}, L_1, and L_x. Since Case (i) and Case (ii) in Eqs. (2a) and (2b) are mutually exclusive events, we can simply combine the formula of T_{b1} and T_{b2} to obtain the length of the time interval $T_{B(\text{JET})}$ in which data burst DB_x is blocked by CP_1 under all circumstances if the OBS JET channel schedule scheme is used as

$$\begin{aligned} T_{B(\text{JET})} = {} & L_1 - \min(\max(T_{fx} - T_{f1}, 0), L_1) \\ & + \min(\max(T_{f1} - T_{fx}, 0), L_x), \end{aligned} \tag{5}$$

and $Y_{(\text{JET})} = \max(T_{f1} - T_{fx} - L_x, 0)$ because the CP_1 that can block DB_x should arrive no later than $t_x - \max(T_{f1} - T_{fx} - L_x, 0)$ according to Eq. (2b).

If Horizon OBS is used, we can modify the blocking condition in Eq. (2b) to $t_x + T_{fx} < t_1 + T_{f1}$ to show the fact that data burst DB_x will be blocked by any scheduled data burst DB_1 with arrival time t_{a1} later than t_{ax}. We can similarly derive the maximum length of the blocking time interval $T_{B(\text{Horizon})}$ for Horizon OBS as

$$\begin{aligned} T_{B(\text{Horizon})} = {} & L_1 - \min(\max(T_{fx} - T_{f1}, 0), L_1) \\ & + \max(T_{f1} - T_{fx}, 0), \end{aligned} \tag{6}$$

and $Y_{(\text{Horizon})} = 0$.

In JIT, any scheduled data burst DB_1 with its data burst ending time later than the control packet CP_x arrival time t_x will block the data burst DB_x. The maximum length of the blocking time interval $T_{B(\text{JIT})}$ for JIT can be written as

$$T_{B(\text{JIT})} = T_{f1} + L_1, \tag{7}$$

and $Y_{(\text{JIT})} = 0$.

Despite the simplicity of expressions (5)–(7), they explain some key properties of OBS. One can easily observe from expressions (5)–(7) that all JET, Horizon, and JIT OBS will have the same performance if the offset times (T_{f1} and T_{fx}) are zeros. When the offset times are not zeros, JET and Horizon OBS will have the same throughput performance (the values of $T_{B(\mathrm{JET})}$ and $T_{B(\mathrm{Horizon})}$ being equal) if the difference between T_{f1} and T_{fx} is always smaller than L_x. Under such circumstances, the time gap between a control packet and its data burst will always be insufficient to accommodate the data burst of a later-arriving control packet, i.e., no extra transmission bandwidth savings can be gained from JET when compared with Horizon OBS. When $T_{f1} = T_{fx}$, both JET and Horizon OBS have the same throughput performance as that of the case of zero offset time, i.e., $T_{B(\mathrm{JET})} = T_{B(\mathrm{Horizon})} = L_1$. In JET and Horizon OBS, 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 result of $T_{B(\mathrm{JET})}$ and $T_{B(\mathrm{Horizon})}$.

In order to show the role of T_{cp} in the throughput performance of OBS, we substitute the offset time T_{off} from Eq. (1) into Eq. (5) and assume $L_1 = L_x = L$. We define H_1 and H_x as the numbers of hops to the destinations of data bursts DB_1 and DB_x, respectively. We have

$$T_{B(\mathrm{JET})} = L - \min(\max((H_x - H_1) \times T_{cp}, 0), L) + \min(\max((H_1 - H_x) \times T_{cp}, 0), L), \quad (8)$$

and $Y_{(\mathrm{JET})} = \max((H_1 - H_x) \times T_{cp} - L, 0)$. Note that the optical switch reconfiguration time T_{sw} is canceled in Eq. (8) and does not play any role in $T_{B(\mathrm{JET})}$ (also $T_{B(\mathrm{Horizon})}$), i.e., T_{sw} has no impact on the channel scheduling of JET and Horizon OBS. However, large T_{sw} can also cause throughput decrease if we further consider the requirement of internal connection setup inside the optical switches [5]. From Eq. (8), one can also observe that JET will have no better throughput performance than that of Horizon if $T_{cp} < L/(H_{\max} - 1)$, where $H_{\max}$ is the maximum hop count of the paths in the network [14]. Since $H_{\max}$ of NSFNet is 5, the throughput delay curves of $T_{cp} = 0.1$ in Figs. 3 and 4 are very similar.

Figure 8 plots the length of the time interval $T_{B(\mathrm{JET})}$ in which t_1 occurs such that CP_1 will block CP_x (DB_x) for different ($H_x - H_1$) and T_{cp}. The blocking time interval length $T_{B(\mathrm{JET})}$ and T_{cp} in Fig. 8 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 for $T_{cp} \geq L$, whereas the dashed line is for $T_{cp} = 0$. From Fig. 8, $T_{B(\mathrm{JET})}$ is L when $H_x = H_1$. For $H_x > H_1$, $T_{B(\mathrm{JET})}$ decreases linearly and becomes zero when $(H_x - H_1) \geq L/T_{cp}$. As T_{cp} increases, the range of $(H_x - H_1)$ for nonzero $T_{B(\mathrm{JET})}$ decreases rapidly and becomes zero for $T_{cp} > L$. For $H_x < H_1$, the interval increases linearly and becomes $2L$ when $(H_x - H_1) \leq -L/T_{cp}$. As T_{cp} increases, the blocking time interval $T_{B(\mathrm{JET})}$ becomes $2L$ for all $(H_x - H_1) < 0$ when $T_{cp} > L$. From Eq. (8), we further note that the blocking time intervals corresponding to different values of H_1 will not overlap with each other if $T_{cp} > 2L$. Under such circumstances, the total length of the time intervals in which data burst DB_x is blocked by any data burst will be equal to $2(H_{\max} - H_x)L$ and will no longer increase with T_{cp}. This is the reason that JET OBS becomes insensitive to the increase of T_{cp} when T_{cp} is sufficiently large as shown in Fig. 3.

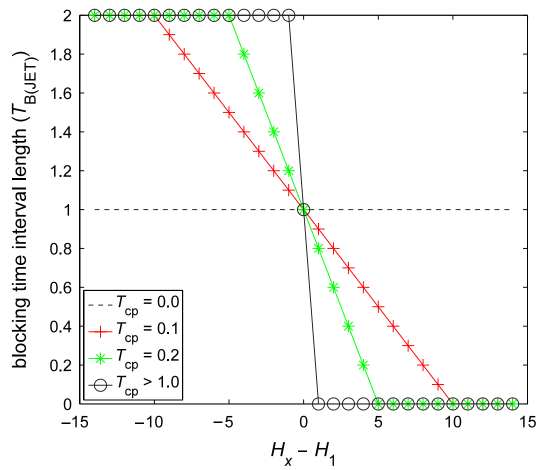

Fig. 8. (Color online) The length of the blocking time interval T_B for CP_1 to block CP_x, DB_2 versus $(H_x - H_1)$ for different T_{cp} using $T_{cp} = H \times T_{cp} + T_{sw}$.

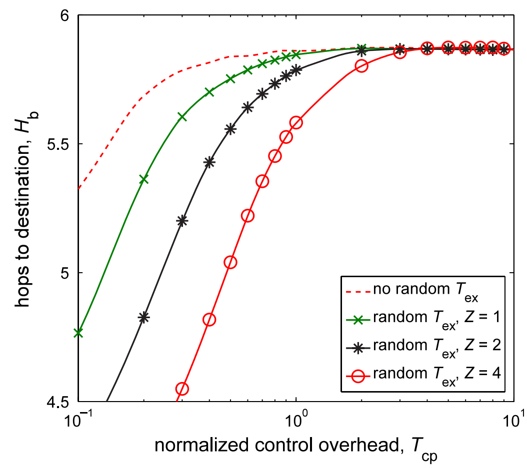

Fig. 9. (Color online) H_b of an eight channel 8×8 MSN with JET OBS using different control overhead T_{cp} when the normalized loading is one.

From Fig. 8, we observe that a larger T_{cp} not only enhances the offset time priority effect but also increases the probability of a data burst being blocked by data bursts with larger hop counts. This phenomenon can also be observed in the simulations. We define H_b as the average number of hops to the destination of the data bursts that block a data burst at a node. Figure 9 shows H_b of an eight channel 8×8 MSN with JET channel scheduling under different control overhead T_{cp} when the normalized loading is one. The paths of an 8×8 MSN have an average hop count of 5 and the maximum hop count is 9. In Fig. 9, the dashed curve is the simulated H_b of JET OBS 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 success probability in channel reservation when T_{cp} increases. However, this also increases the bandwidth wastage when these data bursts are blocked.

From Fig. 9, we also observe the JET OBS throughput being insensitive to the change of T_{cp} when T_{cp} is large. H_b of JET OBS will have a value of around 5.8 even if we increase T_{cp} from 1 to 10.

IV. Approaches to Improve the OBS Throughput Performance

In spite of the importance of higher throughput, any throughput improvement methods should not require network-wide signaling or sophisticated optical hardware. The required computations for resource reservation should also increase only moderately. With these constraints, a single method is usually insufficient to provide the required throughput improvement. By combining methods that are based on different solution strategies, it may be possible to improve the OBS throughput performance without significantly increasing the OBS implementation complexity if all these methods require low additional implementation overhead. The candidates of performance improvement methods include adding FDLs, random extra offset time, and window-based channel scheduling. We first investigate the nature of each method and how they improve the throughput performance.

A. Adding Fiber Delay Lines

From the discussion in Section III, we know that data bursts with different values of H will have the same probability to block each other if the offset time T_{off} is a constant. Under such conditions, we can eliminate the transmission bandwidth wastage caused by the offset time priority effect. Thus we should install FDLs at the node inputs to delay the incoming data bursts' T_{cp} time. Similar approaches have been discussed in early studies of OBS, for example, in tell-n-go (TAG) OBS [1]. It has been reported in [1] that JET OBS will have better performance if the same number of FDLs are used as optical bufferlike FDL delay units in JET OBS. The implementation of optical buffers to date, however, is still difficult. In contrast, only a single simple FDL (as shown in Fig. 10) is required per node input to compensate the T_{cp} of all incoming data bursts in all wavelength channels. It may not be easy to use FDLs to exactly compensate the control packet processing time T_{cp} because T_{cp} can vary with the system loading and nodes. We observe that one solution is to set the delay time T_{FDL} of FDLs to the maximum of T_{cp} and delay the forwarding of the control packet to the next node, if necessary, to keep the offset time T_{off} to be a constant.

When the offset time T_{off} is constant, data bursts with different path lengths will have the same channel reservation success probability at an intermediate node. Therefore, data bursts with larger hop count paths will suffer from larger loss rate. We observe that the solution requiring minimum extra effort is to overcompensate the control packet processing time by setting the length of the FDLs to slightly larger than that required for the compensation of T_{cp}. We define $T_{\mathrm{FDL}} = T_{cp} + \beta$, where $\beta \geq -T_{cp}$ is the compensation factor. The cases of $\beta < 0$ and $\beta > 0$ respectively represent when FDLs undercompensate and overcompensate the control packet processing time T_{cp}.

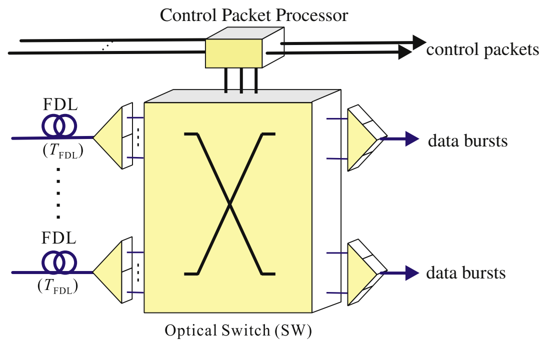

Fig. 10. (Color online) An OBS node with simple FDLs installed at each input port for control packet processing time compensation.

Undercompensation of T_{cp} will surely degrade OBS system performance, whereas overcompensation can often improve the throughput performance [7]. Overcompensation of T_{cp} will increase the offset time between the control packet and the data burst when they pass the nodes along the path. Data bursts that have passed more nodes will therefore have a greater chance to reserve an output channel at an intermediate node because of the larger offset time. Since small hop count data bursts are being penalized, however, a large value of T_{FDL} does not guarantee the increase of system throughput. Though β is critical to the system performance, the proper value of β is not easy to determine because it varies with network topology and traffic. From implementation consideration, it may be better to first use FDLs to compensate the control packet processing time T_{cp} and then use other methods to further improve the throughput performance.

B. Random Extra Offset Time

Throughput improvement has been observed with an extra random offset time [18] and this is attributed to the traffic shaping effect of the data bursts at OBS source nodes. However, we find that the random extra offset time also significantly weakens the connection between the number of hops to destination H and the offset time, and hence reduces blocking. We observe that random extra offset time can further improve the throughput performance even if the offset time priority effect of OBS is reduced by FDL compensation. Random extra offset time can reduce the loss rate of data bursts with large hop count paths and improve the throughput performance. The detail of the operation principle will be discussed later.

We first consider the cases of OBS without control packet processing time compensation. When a random extra offset time is added to Eq. (1), the offset time becomes $T_{\mathrm{off}} = H \times T_{cp} + T_{sw} + T_{ex}$, where T_{ex} is the random extra offset time. The difference of the two offset times T_{fx} and T_{f1} is

$$T_{fx} - T_{f1} = (H_x - H_1)T_{cp} + T_{\mathrm{diff}}, \tag{9}$$

where T_{diff} is the difference between the two random extra offset times. For $T_{fx} - T_{f1} \geq 0$ or $T_{\mathrm{diff}} \geq -(H_x - H_1)T_{cp}$, Eq. (3)

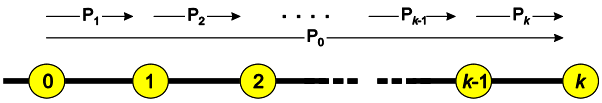

Fig. 11. (Color online) A network with one k-hop path and k one-hop paths.

will be valid (having a nonzero blocking time interval) only if

$$-(H_x - H_1)T_{cp} < T_{\text{diff}} < L_1 - (H_x - H_1)T_{cp}. \tag{10}$$

In contrast, Eq. (4) will be valid for all $T_{\text{diff}} < -(H_x - H_1)T_{cp}$. As we assign T_{ex} at random, T_{diff} is independent of $(H_x - H_1)$ and has the probability density function (pdf)

$$f_{\text{diff}}(y) = \int_{-\infty}^{\infty} f_{\text{ex}}(y+x) f_{\text{ex}}(x)\text{d}x, \tag{11}$$

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

$$f_{\text{diff}}(y) = (Z - |y|)/Z^2, \qquad \text{for } -Z < y < Z, \tag{12}$$

and zero otherwise. With $(H_x - H_1)T_{cp} = L$, a data burst of H_x will not be blocked by any data burst of H_1 if the offset time satisfies Eq. (1), i.e., no nonzero blocking time interval $T_{B(\text{JET})}$ will be found from Eq. (8). With random extra offset time, however, the probability of having a nonzero $T_{B(\text{JET})}$ will increase. For example, the probability will equal 0.5 if the pdf of T_{diff} is from Eq. (12) and $Z = L$. In Fig. 9, solid curves with crosses, asterisks, and circles are the H_b of JET with random 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 larger random extra offset times. An average delay of $Z/2$, however, is added to the data bursts.

For the performance of OBS with FDL compensation being improved by random extra offset time, we use a simplified model to illustrate the principle. Figure 11 shows an OBS network with constant offset time. There is one k-hop path (P_0) and k one-hop paths (P_1 to P_k). Without the random extra offset time, all data bursts have the same channel reservation success probability s at any node, e.g., $s = 0.5$. The average loss rate of a P_0 data burst will therefore be $B_0 = 1 - s^k$, and that of P_j is $B_j = s^j$ for $1 \le j \le k$. After an extra offset time T_{ex} has been randomly added/subtracted to/from each pair of control packet and data burst of P_0, we assume that the channel reservation success probability of a data burst will become one of the two values $s - s_{ex}$ and $s + s_{ex}$ at random, where s_{ex} ($\le s$) is a random reservation probability caused by T_{ex}. We assume s_{ex} to be a constant for ease of illustration. Although the average channel reservation success probability of a data burst is still s, the average loss rate of P_0 data bursts becomes

$$B_0 = 1 - [(s - s_{ex})^k + (s + s_{ex})^k]/2, \tag{13}$$

and that of P_j is

$$B_j = [(s - s_{ex})^j + (s + s_{ex})^j]/2, \quad \text{for } 1 \le j \le k. \tag{14}$$

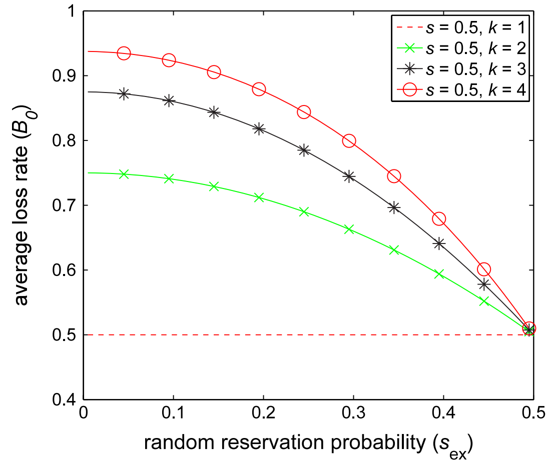

Fig. 12. (Color online) The average loss rates of data bursts with different hop count and random reservation probability. The channel reservation successful probability at a node is 0.5.

Figure 12 shows the B_0 with different hop count k and random reservation probability s_{ex}. The channel reservation success probability of a data burst at any node is $s = 0.5$. In Fig. 12, data bursts with larger hop count paths will have higher loss rate. However, all loss rates of data bursts with hop count $k > 1$ decrease rapidly when we increase the random reservation probability s_{ex}, and they finally converge to 0.5. Apart from B_1 being equal to s, all B_j are now larger than s^j and increase with s_{ex}. The reduction of loss of data bursts with larger hop count paths is achieved at the expense of the increase of loss of data bursts with smaller hop count paths.

Similar to the case of FDL overcompensation, larger values of random extra offset time T_{ex} do not guarantee increased system throughput. From the simulation results, however, JET OBS can have throughput improvement with a large range of T_{ex} and the selection of a suitable value of T_{ex} becomes easy.

C. Window-Based Channel Scheduling

Window-based channel scheduling schemes delay the channel/routing assignment an additional T_{wd} time after reading the information of a control packet. It enables us to predict the impact of a channel assignment to the channel requests (control packets) arriving in the future T_{wd} time interval. We can therefore make better channel/routing assignment decisions than the FCFS approach. To illustrate, Fig. 13 shows four control packets and their associated data bursts arriving at a node. Assuming that all data bursts are routed to the same output port O_x of the node, we may need three output channels if the channel assignment uses FCFS according to the arrival of control packets, e.g., $DB_1 \to O_{x,1}$, $DB_2 \to O_{x,1}$, $DB_3 \to O_{x,2}$, and $DB_4 \to O_{x,3}$, where $O_{x,y}$ is the y th channel of output port O_x. With the additional T_{wd} delay time; however, we need to use only two for the channel assignment, e.g., $DB_1 \to O_{x,1}$, $DB_2 \to O_{x,2}$, $DB_3 \to O_{x,1}$, and $DB_4 \to O_{x,2}$.

There are two major concerns with WBS OBS schemes. First, the additional T_{wd} time delay of the control packet in OBS will increase the equivalent control packet processing

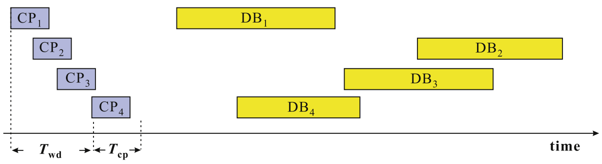

Fig. 13. (Color online) Arrival of control packets CP_1 to CP_4 and their corresponding data bursts DB_1 to DB_4.

time to $T_{cp}+T_{wd}$. Note that a normal JET system with control packet processing time $T_{cp}+T_{wd}$ will have larger blocking probability than that with control packet processing time T_{cp} unless T_{cp} is much larger than the average data burst transmission time L. Similarly, a WBS OBS scheme can have even lower throughput if T_{cp} is not much larger than L. Therefore, for the WBS OBS schemes, we can assume that T_{cp} is much larger than L [19] or the T_{wd} delay time is compensated [8,20]. WBS OBS assuming large T_{cp} does not require any additional hardware and only needs to extend the initial offset time at the source, i.e., $T_{\text{off}}=H\times(T_{cp}+T_{wd})+T_{sw}$. Since the value of T_{cp} should not be restricted, delay compensation seems to be a more attractive approach. At the moment, adding FDLs at the node inputs is the only practical way to compensate the delay of the control packet in a node. We believe that adding FDLs only for T_{wd} compensation is not reasonable. Hence, unlike traditional WBS schemes, we assume that WBS OBS with FDLs for delay compensation will always have constant offset time between control packets and data bursts, i.e., both T_{wd} and T_{cp} are compensated.

Next, it is necessary to determine the procedure for assigning the output channel to a control packet, say CP_x, after the T_{wd} time delay. A common approach is to virtually assign output channels to CP_x and other control packets that have arrived in the T_{wd} time period according to the arrival sequence of their associated data bursts [8,19,20]. The data burst DB_x of CP_x will get the channel that is assigned to DB_x in the virtual channel assignment. DB_x will be rejected if it fails to get a channel in the virtual channel assignment. This approach is effective, e.g., we will need only two output channels in Fig. 13. However, it assumes no compensation for the control packet processing time T_{cp} and only the window time T_{wd} is compensated. It will not be useful if both T_{cp} and T_{wd} are compensated, i.e., no further throughput improvement can be obtained. In such a situation, the data burst arrival sequence is the same as that of the control packets.

We need a WBS OBS channel assignment procedure for all circumstances. We propose to use a basic principle: reject a data burst if it will cause the blocking of subsequent data bursts and decrease the system throughput. Thus we assign the output channel based on the impact of the control packet on other control packets (their associated data bursts) arriving in the T_{wd} delay time interval. Consequently, we weigh the data burst DB_k of a control packet CP_k with a value w_k. To assign a channel to a control packet CP_x (data burst DB_x), we first compute two control packet sets $\boldsymbol{S}$ and $\boldsymbol{R}$, where $\boldsymbol{S}$ ($\boldsymbol{R}$) is the set of control packets that arrive in the T_{wd} delay time interval and their associated data bursts will be accepted if DB_x has (has not) been assigned a channel. We will assign a channel to CP_x only if

$$\left[w_x+\sum_{k\in \boldsymbol{S}} w_k\right]\geq \sum_{k\in \boldsymbol{R}} w_k. \tag{15}$$

The computation of sets $\boldsymbol{S}$ and $\boldsymbol{R}$ can assume any channel reservation scheme that is applicable to WBS OBS. For example, if the latest available unused channel with the void filling (LAUC-VF) scheme [4] is used in Fig. 13 assuming two output channels $O_{x,1}$ and $O_{x,2}$ only, we will have $\boldsymbol{S}=\{CP_2(DB_2\to O_{x,1}),CP_3(DB_3\to O_{x,2})\}$ and $\boldsymbol{R}=\{CP_2(DB_2\to O_{x,1}),CP_3(DB_3\to O_{x,2}),CP_4(DB_4\to O_{x,1})\}$ according to the arrival sequence of CP_2, CP_3, and CP_4. 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 [4]. Hence, we will assign a channel to DB_1 if $w_1+w_2+w_3\geq w_2+w_3+w_4$ (or $w_1\geq w_4$).

The weighing of the data burst is important for WBS OBS performance. We have tested the w_x setting of (1) a nonzero constant c; (2) the data burst length L_x; (3) the inverse of the number of hops to the destination, $1/H_x$; (4) the number of passed hops from source h_x; (5) the traveling distance from the source d_x; and (6) combinations of these parameters. Choices (1) and (2) will be equivalent if the burst length of the network is fixed. Choices (4) and (5) will be equivalent if all network links have the same length. Finally, we chose $w_x=L_x(1+h_x)$. We want a w_x setting to maximize the data bytes to be transferred in the T_{wd} time interval and also increase the priority of data bursts with large hop count. From a large number of simulations, this simple w_x setting provides slightly better system throughput performance than that from the common approach of channel assignment when WBS OBS is without FDL compensation. Note that the common channel assignment procedure approach does not improve the throughput of WBS OBS with FDL compensation. However, the proposed channel assignment procedure can further improve the system throughput in such cases. More discussion will be provided in Section V.

V. Performance Evaluation

We use simulations to verify the throughput performance improvement using the combinations of the methods discussed in Section IV on a 14-node 21-link NSFNet (Fig. 6) and a 64-node 128-link 8×8 MSN topology network [13]. The observations and conclusion, however, can also be applied to other network topologies. We assume that all links in the NSFNet of Fig. 6 are bidirectional. The links in an 8×8 MSN, however, are unidirectional and transmission directions follow the MSN practices. In the simulations, we focus on the impact of the resource reservation and assume that data bursts arriving at the nodes follow the 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 shortest-path routing to determine the path. The maximum number of paths per link for the NSFNet and 8×8 MSN are 23 and 325, respectively. Therefore, the maximum throughput per node is 13/23 or around 0.565 for the NSFNet and 63/325 or around 0.194 for the 8×8 MSN.

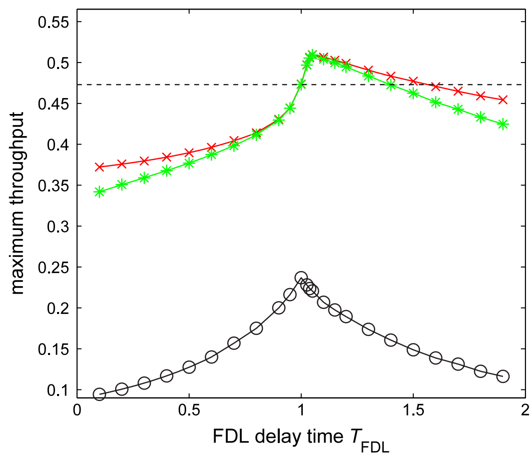

Fig. 14. (Color online) The maximum throughput of different OBS systems on an NSFnet with $T_{cp} = 1$ and using FDL compensation for different FDL delay time T_{FDL}.

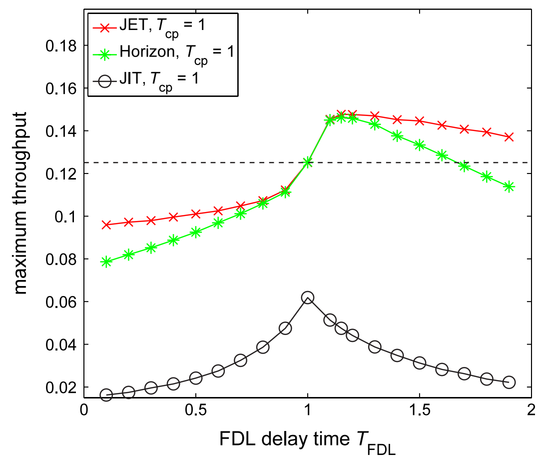

Fig. 15. (Color online) The maximum throughput of different OBS systems on an 8×8 MSN with $T_{cp} = 1$ and using FDL compensation for different FDL delay time T_{FDL}.

The transmission time L of each data burst is an exponentially distributed random number with a mean of one time unit. Once a new data burst arrives at a node, a control packet is sent out immediately to reserve the required channels and resources on the path. The data burst is then transmitted after the offset time T_{off} according to one of the six settings (a) HT_{cp}, (b) $HT_{cp} + T_{ex}$, (c) $H(T_{cp} + T_{wd})$, (d) T_{cp}, (e) $T_{cp} + T_{ex}$, and (f) $T_{cp} + T_{wd}$. Settings (a), (b), and (c) are for OBS without the control packet processing time compensation by FDLs, whereas settings (d), (e), and (f) are for those with FDL compensation. In addition, settings (a) and (e) are for OBS that also uses the random extra offset time approach, whereas settings (c) and (f) are for those that also use window-based channel scheduling. In the simulations, we assume negligible switch reconfiguration time in the OBS node ($T_{sw} = 0$). The offered loading to a node is the number of data burst arrivals to the node per unit time divided by the number of wavelength channels per link. There are eight wavelength channels per link. The link propagation time between nodes is proportional to the link length as shown in Fig. 6, and the minimum is set to $10L$. All nodes receive the same offered loading. All simulations are run sufficiently long such that the 95% confidence intervals are less than 1% of the average values of the results.

A. Performance Due to Overcompensation of T_{cp}

We plot the maximum throughput of different OBS systems on the NSFNet and 8×8 MSN in Figs. 14 and 15 with control overhead $T_{cp} = 1$ and using FDL compensation of different T_{FDL}. Hence, the β defined in Subsection IV.A is given by $\beta = T_{\mathrm{FDL}} - 1$. The curves with crosses, asterisks, and circles are the maximum throughputs of JET, Horizon, and JIT OBS, respectively. Both JET and Horizon OBS can benefit from overcompensation of T_{cp}. Their maximum throughput in Fig. 14 (Fig. 15) increases from 4.75 to 5.09 (from 0.125 to 0.148) when we increase T_{FDL} from 1 to 1.05 (from 1 to 1.15), but the throughput then decreases after that. Note that we do not need to change the actual value of T_{FDL}. As discussed in Subsection IV.A, we can have the same effect by readjusting the delay of forwarding the control packet to the next node. Since in practice the maximum throughput can change drastically with T_{FDL}, we may need an intelligent OBS node to adjust the delay according to the network status. This will significantly increase the OBS implementation complexity.

From Figs. 14 and 15, we also observe that JET OBS always enjoys the highest throughput. Horizon OBS will be a good replacement for JET OBS if we can keep T_{FDL} close enough to T_{cp}, e.g., $T_{\mathrm{FDL}} = T_{cp} \pm 0.05$ in Fig. 14 and $T_{\mathrm{FDL}} = T_{cp} \pm 0.1$ in Fig. 15. JIT OBS will have the smallest throughput even if the processing delay is perfectly compensated, i.e., $T_{\mathrm{FDL}} = T_{cp}$. Adding FDLs at node input ports (as shown in Fig. 10) will delay all incoming data bursts by a T_{FDL} time. It is equivalent to adding an extra T_{FDL} offset time between a control packet and its data burst when the control packet arrives at a node though the initial offset time of the control packet may be zero at the source. From Eq. (7), we realize that the blocking probability of a JIT OBS data burst is proportional to the average length and offset time of other data bursts in the networks. Using FDL to compensate T_{cp} can increase the JIT OBS throughput in Fig. 14 (Fig. 15) from 0.094 to 0.237 (from 0.016 to 0.062) when we increase T_{FDL} from 0.1 to 1. However, it is still much lower than that of JET or Horizon OBS. Thus, JIT OBS is in general only suitable for networks with large data bursts such that T_{cp} will be small.

B. Performance of Extra Random Offset Time

Figures 16 and 17 are the throughput–delay curves of JET OBS with random extra offset time T_{ex} when $T_{cp} = 0.1$ and 1, respectively. The curves with asterisks, circles, and squares are for JET with random extra offset time T_{ex} of $Z = 1$, 10, and 100, respectively. Note that the two curves of $Z = 100$ in Figs. 16 and 17 are similar because the large value of random extra offset time T_{ex} dominates the OBS performance. From both figures, we observe that the random extra offset time approach in general can improve the system throughput, but the suitable Z will depend on T_{cp}. For example, the best Z in Fig. 17 ($T_{cp} = 1$) is $Z = 100$ with throughput of 0.435 when

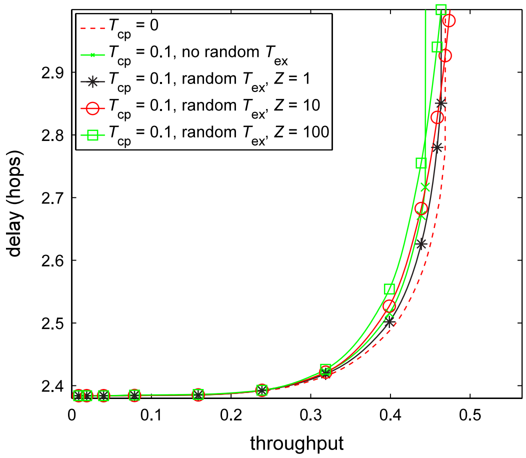

Fig. 16. (Color online) The throughput–delay performance of JET OBS on an NSFNet with $T_{cp} = 0.1$ and using random extra offset time T_{ex} of different Z.

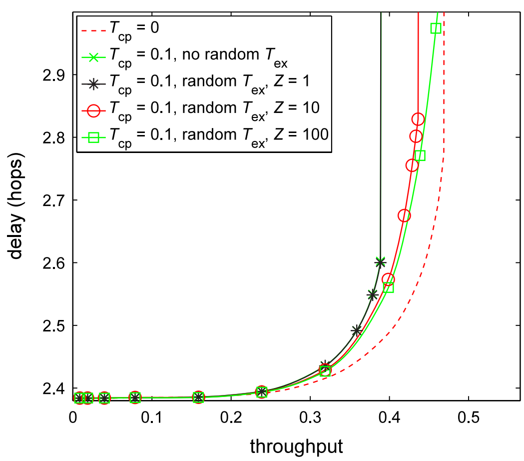

Fig. 17. (Color online) The throughput–delay performance of JET OBS on an NSFNet with $T_{cp} = 1.0$ and using random extra offset time T_{ex} of different Z.

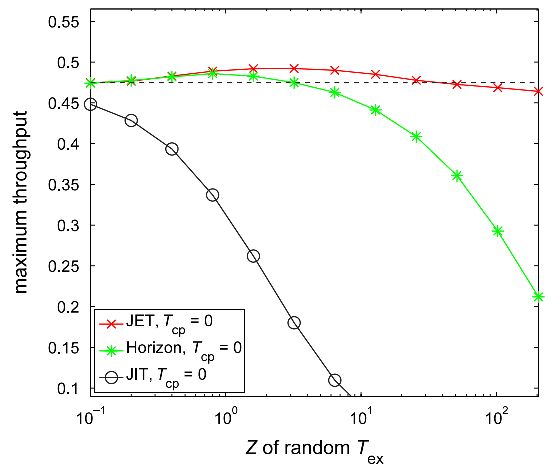

Fig. 18. (Color online) The maximum throughput of different OBS systems on an NSFNet with $T_{cp} = 0$ and using random extra offset time T_{ex} of different Z.

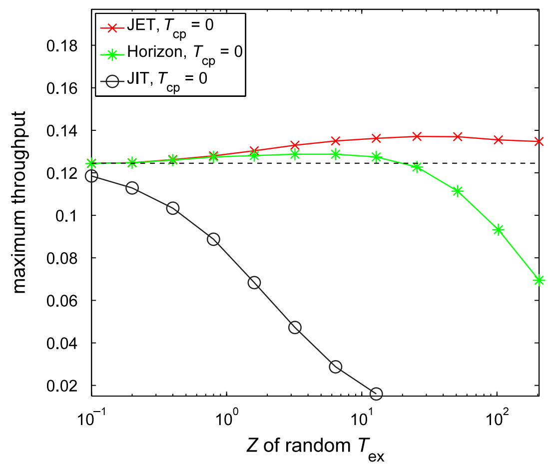

Fig. 19. (Color online) The maximum throughput of different OBS systems on an 8×8 MSN with $T_{cp} = 0$ and using random extra offset time T_{ex} of different Z.

the delay is 2.75 hops, whereas that of the normal JET has a value of 0.39. Also, the throughput increases with the value of Z. However, in Fig. 16 ($T_{cp} = 0.1$), the best Z is $Z = 1$ with throughput of 0.456, which is larger than the throughput 0.439 of the case of $Z = 100$.

Figures 18 and 19 show the maximum throughput of OBS systems using the extra random T_{ex} approach and the T_{cp} having been compensated by FDLs. The network topologies used in Figs. 18 and 19 are the NSFNet (Fig. 6) and the 8×8 MSN. In the figures, the curves with crosses, asterisks, and circles are the maximum throughputs of JET, Horizon, and JIT OBS, respectively, with $T_{cp} = 0$ and with random T_{ex} of different Z. Note that the actual value of T_{cp} will not be important for JET and Horizon OBS if it is compensated by FDLs. The figures show that the random T_{ex} approach is good for both JET and Horizon OBS. In Fig. 18 (Fig. 19), the best maximum throughput of JET and Horizon OBS are 0.490 and 0.484 (0.137 and 0.128), respectively, when $Z = 2$ and $Z = 1$ ($Z = 30$ and $Z = 5$). JIT OBS reserves the output channel once the control packet arrives at a node. Applying the random T_{ex} approach will increase the offset time and will reduce the throughput of JIT OBS as shown in the figures. Horizon OBS does not reuse the transmission bandwidth between the control packet and the data burst. If T_{ex} is small, the random T_{ex} approach can improve the throughput of Horizon OBS by reducing the loss rate of data bursts with large hop count paths. For large T_{ex}, however, the transmission bandwidth wastage in the offset time dominates and decreases the throughput of Horizon OBS. JET OBS better utilizes the transmission bandwidth and has a large tolerance range of T_{ex}, e.g., the maximum throughput changes from 0.492 to 0.471 in Fig. 18 (0.131 to 0.136 in Fig. 19) when we increase Z from 2 to 100. From the simulation results, we observe that a rule of thumb is to set T_{ex} to a value that ranges from the average to maximum hop count of the paths.

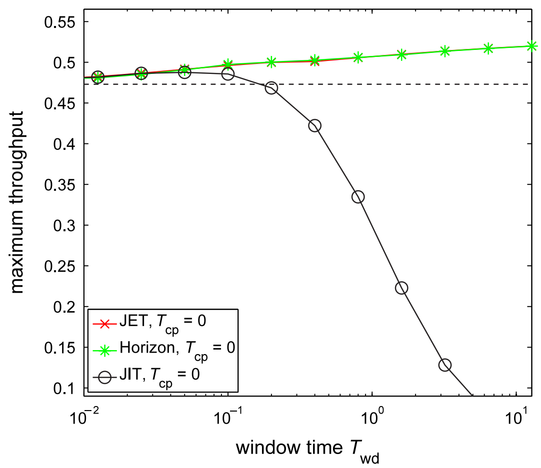

Fig. 20. (Color online) The maximum throughput of different OBS systems on an 8×8 MSN with $T_{cp} = 0$ and window time T_{wd}.

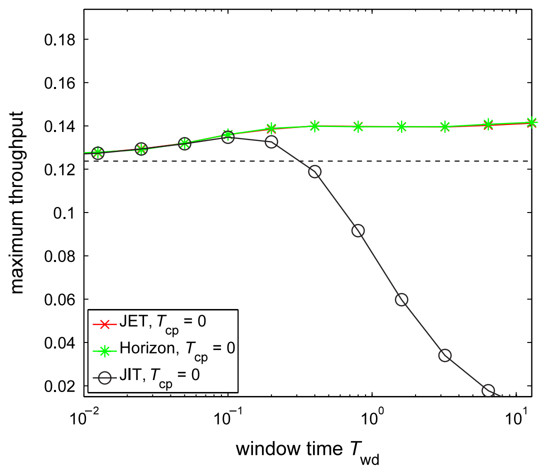

Fig. 21. (Color online) The maximum throughput of different OBS systems on an 8×8 MSN with $T_{cp} = 0$ and window time T_{wd}.

C. Performance of Window-Based Channel Scheduling

Figures 20 and 21 show the maximum throughput of OBS systems using the proposed WBS with different window time T_{wd}. Similarly, we set $T_{cp} = 0$. In the proposed WBS with FDLs for processing time compensation, both JET and Horizon will have the same performance, i.e., the throughputs of JET and Horizon OBS are nearly identical. Also, from Figs. 20 and 21, WBS will reduce the JIT OBS throughput except when T_{wd} is small. In contrast, WBS can provide significant throughput performance improvement for JET (Horizon). We will have higher throughput with larger T_{wd}, though the increase in throughput diminishes with the increase of T_{wd}. To further improve the throughput performance, we may combine WBS with FDL overcompensation. Figure 22 shows the throughput–delay curves of WBS OBS on an NSFNet with $T_{cp} = 1.0$. The curves with asterisks and circles are throughput–delay curves for WBS OBS with $T_{wd} = 1$ and $T_{\mathrm{FDL}} = 2.0$ and 2.1, respectively. The curves with squares and diamonds are those for WBS OBS $T_{wd} = 8$ and $T_{\mathrm{FDL}} =$ 9.0 and 9.1, respectively. The curve with crosses is that of normal JET OBS with $T_{cp} = 1$. The dashed curve is the reference throughput–delay curve of JET OBS with $T_{cp} = 0$. The WBS OBS maximum throughput increases from 0.5 to 0.51 when we change T_{wd} from 1 to 8. To increase throughput without further increasing T_{wd}, we slightly overcompensate the processing delay by changing T_{FDL} to 9.1, and the maximum throughput then increases to 0.52 (the curve with diamonds). Note that FDL overcompensation does not guarantee throughput improvement. In Fig. 22, the maximum throughput of the WBS OBS with $T_{\mathrm{FDL}} = 2.1$ is slightly smaller than that of $T_{\mathrm{FDL}} = 2.0$ (the curves with circles and asterisks).

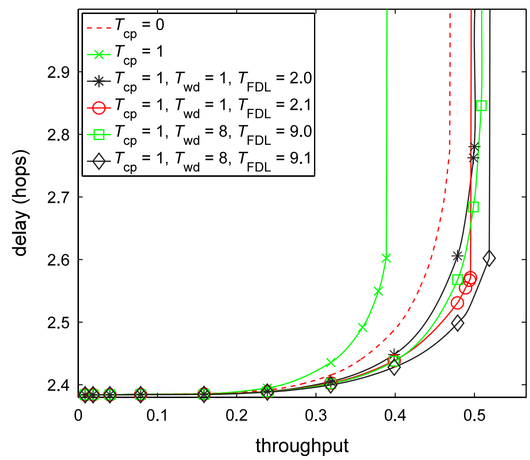

Fig. 22. (Color online) The throughput–delay performance of WBS OBS on an NSFNet with $T_{cp} = 1.0$ and using FDL compensation of different FDL delay T_{FDL}.

Figures 23 and 24 show the throughput–delay curves for WBS OBS without FDLs for processing time compensation when $T_{cp} = 1$ and 5, respectively. The curves with crosses are for normal JET OBS (no WBS). The curves with asterisks and circles are for WBS OBS with $T_{wd} = 8$ using the proposed channel assignment procedure and the common approach, respectively. The simulation results show that the WBS should only be used as a supplementary approach to further improve the throughput performance of OBS with FDL processing time compensation. Without FDL processing time compensation, the common approach cannot provide any throughput performance improvement. A small throughput improvement can be obtained with the proposed channel assignment procedure if T_{cp} is large, e.g., $T_{cp} = 5$. As discussed in Subsection IV.C, however, the T_{cp} of WBS OBS in general should not be restricted.

In overcompensation of T_{cp}, it is difficult to determine the required FDL length (T_{FDL}) and to have the precise control. Therefore, we need methods to further improve the throughput performance after the processing time is compensated by FDLs. From Figs. 18 to 22, one can observe that WBS OBS provides better throughput improvement than that of the random extra offset time approach. Moreover, in some cases, the random extra offset time approach may only provide very slight performance improvement to OBS, for example, as shown in Fig. 18. In spite of this, the random extra

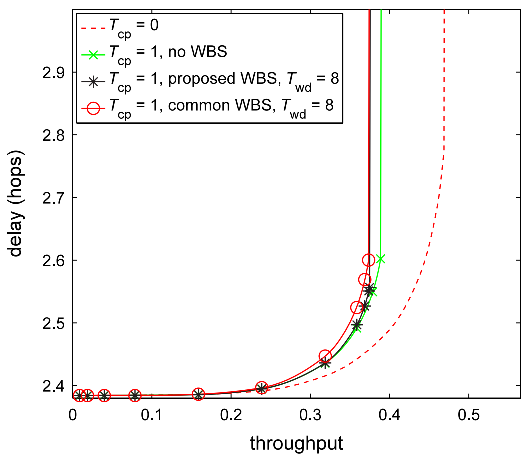

Fig. 23. (Color online) The throughput–delay performance of WBS OBS on an NSFNet with $T_{cp} = 1.0$ and without FDL processing time compensation.

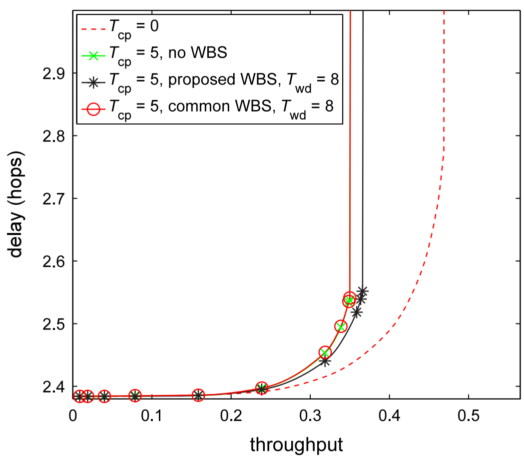

Fig. 24. (Color online) The throughput–delay performance of WBS OBS on an NSFNet with $T_{cp} = 5.0$ and without FDL processing time compensation.

offset time approach should be applied first to improve the throughput performance of OBS because it requires no additional OBS node hardware and computational capability. The WBS approach will be the next step if higher throughput is desired.

VI. Summary and Conclusion

Optical burst switching is a feasible way to implement all-optical data transmission between networks nodes. Owing to the unavailability of practical optical buffers, OBS is unable to store the income data bursts before making the routing/forwarding decision. Therefore, suboptimal channel assignment often results. JET OBS can better utilize the transmission bandwidth, but its system throughput still suffers from the large control packet processing time (T_{cp}) and the loss of data bursts with large hop count paths. We present an OBS performance model to explain the phenomenon that the throughput performance degrades when T_{cp} increases.

By combining methods that are based on different principles, we can improve OBS performance without significantly increasing the OBS implementation complexity. We have considered three different methods for the combined approach: adding simple FDLs, random extra offset time, and WBS.

Adding a single FDL to an input port of an OBS node can compensate the T_{cp} of incoming data bursts in all wavelength channels of the input port. This improves the system throughput performance by eliminating the offset time priority effect. Although overcompensating the T_{cp} is a way to further improve the system throughput performance, it needs intelligent nodes to adjust the FDL delay time T_{FDL} according to the traffic fluctuations. As this will largely increase the implementation complexity of OBS, we propose to use random extra offset time and WBS approaches for further improving the OBS throughput performance.

The random extra offset time approach can increase the throughput, with and without the FDL to compensate the processing time. Although there is an optimal value of the average extra offset time T_{ex}, the simulation results show that we can have throughput improvement with a large range of T_{ex}. We consider WBS OBS in two cases (i) both the window time T_{wd} and the control packet processing time T_{cp} are compensated by FDLs, and (ii) no FDL is used for T_{wd} and T_{cp} compensation, but T_{cp} is much larger than the data burst transmission time L. We have investigated and derived a suitable channel assignment method for WBS OBS. From the simulation results, WBS OBS can provide further throughput improvement and the throughput increases with T_{wd}. Compared with the approach to overcompensate the T_{cp}, this simplifies the implementation of WBS OBS because we no longer need to consider the optimal value of T_{wd}. From the simulation results, WBS in general can provide better throughput improvement than the random extra offset time approach. Owing to the low requirements of hardware and computational capability, however, the random extra offset time approach should be applied first to OBS systems for improving the throughput performance.

Acknowledgments

This research was supported by a grant from The Hong Kong Polytechnic University (project number J-BB9M) and by the University of Hong Kong Strategic Research Theme of Information Technology.

References

[1] C. Qiao and M. Yoo, "Optical burst switching (OBS)—a new paradigm for an optical Internet," *J. High Speed Netw.*, vol. 8, pp. 69–84, 1999.

[2] J. S. Tuner, "Terabit burst switching," *J. High Speed Netw.*, vol. 8, pp. 3–16, 1999.

[3] J. Y. Wei and R. I. McFarland Jr., "Just-in-time signaling for WDM optical burst switching networks," *J. Lightwave Technol.*, vol. 18, pp. 2019–2037, 2000.

[4] Y. Xiong, M. Vandenhoute, and H. C. Cankaya, "Control architecture in optical burst-switched WDM networks," *IEEE J. Sel. Areas Commun.*, vol. 18, pp. 1838–1851, 2000.

[5] C. Y. Li, G. M. Li, P. K. A. Wai, and V. O. K. Li, “Optical burst switching with large switching overhead,” *IEEE J. Lightwave Technol.*, vol. 25, pp. 451–462, 2007.

[6] I. Widjaja, “Performance analysis of burst admission-control protocols,” *IEE Proc. Commun.*, vol. 142, pp. 7–14, 1995.

[7] B. C. Kim, Y. Z. Cho, and D. Montgomery, “An efficient optical burst switching technique for multi-hop networks,” *IEICE Trans. Commun.*, vol. E87-B, pp. 1737–1740, 2004.

[8] J. Li, C. Qiao, J. Xu, and D. Xu, “Maximizing throughput for optical burst switching networks,” *IEEE/ACM Trans. Netw.*, vol. 15, pp. 1163–1176, 2007.

[9] X. Lu and B. L. Mark, “Performance modeling of optical-burst switching with fiber delay lines,” *IEEE Trans. Commun.*, vol. 52, pp. 2175–2183, 2004.

[10] V. M. Vokkarane and J. P. Jue, “Prioritized burst segmentation and composite burst-assembly techniques for QoS support in optical burst-switched networks,” *IEEE J. Sel. Areas Commun.*, vol. 21, pp. 1198–1209, 2003.

[11] M. Duser and P. Bayvel, “Analysis of a dynamically wavelength-routed optical burst switched network architecture,” *J. Lightwave Technol.*, vol. 20, pp. 574–585, 2002.

[12] J. Pedro, P. Monteiro, and J. Pires, “Traffic engineering in the wavelength domain for optical burst switched networks,” *J. Lightwave Technol.*, vol. 27, pp. 3075–3091, 2009.

[13] N. F. Maxemchuk, “Routing in Manhattan Street network,” *IEEE Trans. Commun.*, vol. 35, pp. 503–512, 1987.

[14] F. Vázquez-Abad, J. White, L. Andrew, and R. Tucker, “Does header length affect performance in optical burst switched networks?,” *J. Opt. Netw.*, vol. 3, pp. 342–353, 2004.

[15] N. Barakat and E. H. Sargent, “Analytical modeling of offset-induced priority in multiclass OBS networks,” *IEEE Trans. Commun.*, vol. 53, pp. 1343–1352, 2005.

[16] H. M. H. Shalaby, “A simplified performance analysis of optical burst-switched networks,” *J. Lightwave Technol.*, vol. 25, pp. 986–995, 2007.

[17] J. A. Hernandez, J. Aracil, L. Pedro, and P. Reviriego, “Analysis of blocking probability of data bursts with continuous-time variable offsets in single-wavelength OBS switches,” *J. Lightwave Technol.*, vol. 26, pp. 1559–1568, 2008.

[18] S. Verma, H. Chaskar, and R. Ravikanth, “Optical burst switching: a viable solution for terabit IP backbone,” *IEEE Network*, vol. 14, pp. 48–53, 2000.

[19] C. Y. Li, G. M. Li, P. K. A. Wai, and V. O. K. Li, “Novel resource reservation schemes for optical burst switching,” in *Proc. IEEE Int. Conf. Communications*, 2005, pp. 1651–1655.

[20] H. Li, H. Neo, and T. L. J. Ian, “Performance of the implementation of a pipeline buffering system in optical burst switching networks,” in *Proc. Global Communications Conf.*, 2003, pp. 2503–2507.