

An Efficient Real-time Service Discipline for the Control Plane to Enhance the Performance of Optical Burst Switching Networks*

Lalit Kumar Pagaria, T. Venkatesh[†] and C. Siva Ram Murthy

Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai 600036, India

Email: lalitp@cse.iitm.ac.in, murthy@iitm.ac.in

[†]Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati 781039, India
Email: t.venkat@iitg.ernet.in

Abstract—Optical burst switching (OBS) is widely believed to be the technology for the future core network in the Internet. Traditionally, the burst header packet (BHP) is served by the control processor on first-come-first-serve basis which increases the burst loss probability (BLP) at the core nodes closer to the destination due to insufficient offset time. Efficient real-time service disciplines in the control plane can reduce the BLP due to insufficient offset time by giving higher priority to the BHPs with smaller residual offset time. This in turn improves the throughput of OBS networks and reduces the BLP. This paper studies the role of real-time service disciplines used to service the BHPs in the queue in improving the performance of the OBS networks. We propose Least per-Hop Worst-case Laxity First (LHWLF) algorithm which picks the BHP with least available delay budget at that node after accounting for the waiting time in the queue. We demonstrate the effectiveness of our LHWLF algorithm by comparing with other competing ones in the literature in terms of BLP and throughput. We also demonstrate that the proposed algorithm gives fair treatment to bursts with different lengths and the bursts travelling paths of different lengths.

I. INTRODUCTION

Optical burst switching (OBS) is envisioned as one of the popular technologies to enable the deployment of all-optical networks in the core of the Internet [1], [2]. In OBS networks, the ingress node assembles IP packets from the access network into larger sized-data chunks termed as bursts. These bursts are transmitted in the optical domain along pre-computed lightpaths. At the egress node, they are disassembled into the constituent packets which are switched through the access network. Before the burst is transmitted, a control packet also known as the burst header packet (BHP), is transmitted along the path on a separate wavelength (called as control wavelength). It is processed in the electronic domain at each core node. The time gap between the BHP and the data burst, known as the offset time, is kept sufficient enough to process the BHP and to identify the wavelength to be reserved for the

data burst. The BHP carries all the required information about the data burst to enable the reservation of network resources along the path. Typically, the core node is assumed to be free of optical buffers so that the wavelength for the data burst must be reserved before it reaches the node. There are two popular protocols for the wavelength reservation categorized depending on the duration for which the wavelength is reserved [3]. In just-in-time (JIT) protocol, the wavelength is reserved from the time the BHP is processed till the end of the data burst. In just-enough-time (JET) protocol, the wavelength is reserved only for the duration of the data burst.

The offset time is usually kept sufficient to process the BHP at each node and configure the switch. It might also include some extra time required to resolve contentions for the data wavelength. The offset time decreases as the data burst gets closer to the destination due to the processing time and the queueing delay of BHP along the path. If the offset time is not kept sufficiently large to account for any unforeseen delays at the core nodes, the data burst would be dropped if it reaches the core node before the BHP is processed. On the other hand, if the offset time is kept larger than that necessary, it leads to lower utilization of the bandwidth. The offset time might also be adapted to suit the traffic conditions to avoid the loss of bursts due to insufficient offset time [4]. We do not assume adapting offset time because it adds to the complexity of network design. Therefore, offset time dimensioning has a major role in reducing the burst loss probability (BLP) in OBS networks [5], [6], [7]. In the absence of any contention resolution techniques, the residual offset time at a core node (the time between the arrival of the BHP and the corresponding data burst) might become insufficient mainly due to the increase in the queueing delay of the BHP along the path.

The OBS network can be viewed as an overlay network with an optical-electrical-optical (o-e-o) control-plane and an all-optical data-plane. Though there is rich literature on

*This work was supported by IBM Corporation through 2008 Real-Time Innovation Faculty Award.

protocols and architectures to maximize the performance of OBS network through the data-plane, there is little effort to study the impact of the design of control plane on the performance of the OBS network [2]. With high arrival rate of data bursts, especially small bursts, BHPs arrive at a core node with a very high rate. Congestion in the control-plane occurs when the arrival rate of BHPs at the core node exceeds the maximum rate at which they can be processed [8], [9]. Previous work in the literature showed that the congestion in the control plane can lead to the loss of data bursts which limits the throughput of the OBS network for large arrival rate of bursts [5], [6]. Queueing of the control packets was suggested as an effective method to reduce the congestion in the control plane and improve the performance of OBS networks [5], [9]. With queueing of the BHPs in the control plane, the selection of service discipline in the control plane also becomes an important factor that influences the performance of the OBS network [10]. Usually the BHPs are serviced only using first-in-first-out (FIFO) service discipline which was shown to reduce the throughput and lead to higher BLP in data plane. The authors in [10] proposed Largest Length/Laxity Ratio First (LLRF) Algorithm which selects the BHP with largest burst length and lowest laxity from the queue at the BHP processor. It was shown that such a selection reduces the BLP as well as increases the throughput.

In this paper, we show that the existing algorithms for servicing BHPs suffer from unfairness and lack of robustness which motivates the study of efficient service disciplines in the control plane. We propose a real-time service discipline for the control plane which is termed as Least per-Hop Worst-case Laxity First (LHWLF) algorithm. We use per-hop worst-case laxity to determine the tightness of the deadline which is essentially the maximum time by which the BHP can be kept in queue after which it would most likely be dropped due to insufficient offset time before reaching the destination. We provide sufficient justification for the use of this metric to service the BHPs through an example. We compare the performance of the proposed algorithm with FIFO and the LLRF algorithm which shows that our algorithm outperforms the other two and demonstrate its superiority. From this work, we demonstrate that efficient service disciplines in the control plane do improve the performance of the OBS networks.

The rest of the paper is organized as follows. In Section II, we discuss the role of control plane processor and the objectives in the design of a good service discipline in the control plane. We also demonstrate the problems with the LLRF algorithm with an example and motivate the design of the proposed algorithm. Section III, presents the proposed algorithm which is compared with the existing algorithms in terms of BLP and throughput using simulations in Section IV. Finally, we conclude the paper in Section V.

II. CONTROL PLANE SCHEDULING

The performance of the OBS network is known to be governed by several factors: the control processor architecture, complexity and the choice of burst scheduling algorithm, and

the parameters related to the data plane such as, traffic load, number of data and control channels, offset time, length of the data bursts and transmission rate in both data and control planes [11]. In the OBS network, apart from the loss of data bursts due to the unavailability of the wavelengths, there is loss of data bursts when the residual offset time at a node becomes insufficient to find a feasible schedule. In such a case, the BHP can also be dropped to avoid the reservation of the path further to the destination. Due to the lack of optical buffers in the data plane, it is difficult to control the losses in the data plane. The BHP is usually dropped due to congestion in the control plane i.e., overflow of the queue at the control processor. In this work, we assume that the buffer size at the control processor is large enough to prevent the loss of BHP due to buffer overflow. The BHP can be intentionally dropped by the control processor if the offset time is found to be insufficient or if a feasible schedule could not be found for the data burst. The loss of BHPs can be minimized to a large extent by designing efficient algorithms to serve the BHPs waiting in the queue. But, the burst loss due to infeasible schedule cannot be controlled by these algorithms.

A. Real-time Service Disciplines in the Control Plane

In JET-based reservation, the offset time can be seen as the deadline for the BHP to be processed and the wavelength to be reserved at each node. If the BHP is not processed and a schedule for the data burst found by this deadline, the data burst is lost. Further, since the data burst cannot be delayed at any intermediate node, the deadline is hard and cannot be negotiated. Therefore, serving the BHPs in the control processor queue can be seen as a real-time scheduling problem where the offset time becomes the deadline before which the BHP should be serviced by the control processor. At high arrival rate of the BHPs, the order in which the BHPs are processed becomes important to avoid the reduction in the residual offset time due to queueing delay. It was observed through analysis that the throughput of OBS network strongly depends on the variation in the burst lengths and the offset time [12].

In such a setting, design of a service discipline for the control plane becomes a challenging task. The role of the service discipline is to choose a BHP among those queued in the electronic buffer at the control processor. The primary metrics used to evaluate the service disciplines are the BLP (fraction of bursts lost in the OBS network) and the throughput which is defined as the number of bytes received at the egress node per unit time. Few of the principle factors that need to be considered in the design of a good service discipline at the control processor which also hold for the design of any real-time service discipline are [13]:

- *Only Local information:* It should use only local information at the queue which includes, current and average queueing time, number of BHPs in the queue, arrival rate, average processing time per BHP, and the information about the data burst carried by the BHP. It should not

require any additional state information to process the BHP.

- *Computational cost:* The computational cost of the service discipline also depends on the time at which it is invoked. It can be either called at the time of enqueueing or dequeueing. Further, the cost depends on the number of BHPs inspected before selecting a BHP. To speed up the processing, the algorithm must be computationally efficient.
- *Robustness:* It should always be effective even if the parameters in the network such as, topology, number of wavelengths and the load varies.
- *Fairness:* Finally, it should provide fairness in service among flows travelling different path lengths and bursts with different lengths. For example, if the scheduler picks up a BHP purely based on deadline then it is always biased towards the bursts that travel on shorter paths.

B. Motivation for Real-time Service Discipline in the Control Plane

Though FIFO is simple and easy to implement, it is not robust to changes in the network condition and does not consider the fairness issue. It also does not service the BHPs in the order of their deadlines (residual offset times). Hence, FIFO leads to a higher BLP and lower throughput as load increases [10]. To the best of our knowledge the only other service discipline available to select the BHP from the queue is the LLLRF algorithm proposed in [10]. The LLLRF algorithm selects the BHP having the largest ratio of burst length to the laxity among all those BHPs in the queue. Laxity is widely used term in real-time task scheduling which signifies the tightness in the deadline [14]. It is the remaining time for processing the task before the deadline expires. In the LLLRF algorithm, the BHP with least laxity and that corresponding to the largest burst length is given highest priority. It increases the throughput and minimizes the wastage of network resources used by the dropped bursts. On the other hand, considering the BHPs with least laxity prevents the loss of bursts due to insufficient offset time.

In the LLLRF algorithm, the i^{th} BHP in queue is assigned a priority $E(i)$ according to

$$E(i) = len/lax(i, t) \quad (1)$$

where len is the length of the corresponding burst and $lax(i, t)$ is laxity of the i^{th} BHP at time t . The laxity is calculated as

$$lax(i, t) = d - t - t_p \quad (2)$$

where t_p is the time required to process the BHP, and d is the deadline which is the time when the data burst is supposed to reach the destination before which the BHP must be processed. The deadline for the BHP is computed from the residual offset time t_o^r and the arrival time of the BHP t_a as

$$d = t_o^r + t_a \quad (3)$$

It is assumed that the processing time is independent of the load in the network which is true if a high-speed processor

is used. At any time t , if $lax(i, t) = 0$, it is set to a small value $\epsilon = 10^{-6}$ so that the BHP has the highest priority in the queue [10]. If two or more BHPs have the same priority then the processor selects the BHP corresponding to the longer burst.

The LLLRF algorithm does not guarantee fairness across different flows in the network which we illustrate now with an example. It computes the laxity using the offset time which predominantly depends on the number of hops to the destination. It always ensures that those BHPs that need to travel shorter hop lengths are favoured at the bottleneck link. We call the unfairness in the service of BHPs because of the variation in the path lengths as *path length-based unfairness*. To clarify this effect consider the example network shown in Fig. 1. Assume that the bursts are generated from both the ingress nodes $E1$ and $E2$ to the egress nodes $E3$ and $E4$ with the same length L_1 . This is the case usually if threshold-based assembly mechanism is used at the ingress node [2]. At the bottleneck link $C3 - C4$, the BHPs corresponding to the bursts destined to $E4$ suffer longer delay in the queue and might even be dropped if the offset time is insufficient compared to those destined to $E3$. This occurs because the laxity used in computing the priority of BHPs is always smaller for those destined to $E3$ than those towards $E4$.

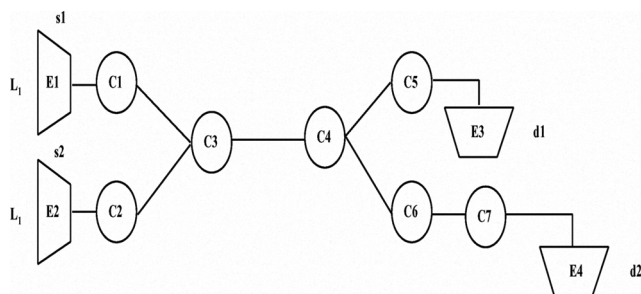


Fig. 1. Unfairness due to distance to destination

Similarly, the LLLRF algorithm is unfair to the bursts with smaller length. We call this unfairness as *burst length-based unfairness*. This is illustrated for the sample network shown in Fig. 2. In this network, the length of both the paths (from $E1$ or $E2$ to $E3$ or $E4$) are same. Assume that the bursts generated at $E1$ have a length L_1 which is larger than those generated at $E2$ with length L_2 . At the bottleneck link $C3 - C4$, BHPs corresponding to the bursts with length L_1 are serviced faster leading to a longer delay to those BHPs corresponding to the bursts with length L_2 . This might eventually lead to higher loss of bursts with shorter length due to insufficient offset time. Such unfairness is present even when offset times have large variation. Both the types of unfairness considered here are more predominant in topologies with larger variation in the path lengths and if the burst length has larger variation. We found that using the burst length to compute the priority always leads to selection of the BHPs based on the length factor. If the variation in the length of the bursts is high, the LLLRF seldom considers the laxity in the selection of the

BHP. These problems of unfairness and the lack of robustness

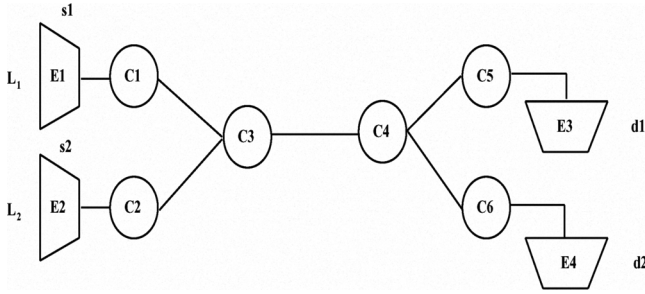


Fig. 2. Unfairness due to burst length

to different network scenarios motivate the design of efficient real-time service disciplines for the control plane to improve the performance of OBS networks. We consider all these issues in the design of the LHWLF algorithm presented in the next section and demonstrate through simulations that these problems are alleviated with our algorithm. This work also opens up new directions in the design of service disciplines for the control plane.

III. PROPOSED REAL-TIME SERVICE DISCIPLINE FOR CONTROL PLANE

In this section, we propose a real-time service discipline to process maximum number of BHPs in the queue at the control processor. All the design objectives mentioned in the previous section to maximize the performance of the scheduler have been considered while designing the algorithm. We define per-hop worst-case laxity which is used in the computation of the priorities for BHPs in the queue and determine the order of service based on the priorities.

A. Design of the Metric

Before we describe the algorithm let us define some important notation used in our algorithm. An i^{th} BHP denoted by bhp_i is represented by the tuple (t_o^r, t_a, h_r, len) , where t_o^r , t_a , h_r and len are the residual offset time at that node, arrival time of the BHP, the number of hops remaining till the destination (including the current one), and the length of the corresponding data burst, respectively. We define the *per-hop worst-case laxity* (HWL) as the maximum time a BHP can be delayed at a core node beyond which the corresponding burst is dropped due to insufficient offset time. For i^{th} BHP, the HWL at node j at time t is computed as

$$L^j(i, t) = lax^j(i, t) - h_r * t_p \quad (4)$$

where $lax^j(i, t)$ is laxity to the destination node at time t . We subtract the term $h_r * t_p$ from the laxity to account for the time required to process the BHP at the downstream nodes (including the current one) along the path. The HWL actually indicates the deadline for the BHP at the current node. We compute the laxity as

$$lax^j(i, t) = t_o^r(i) - t_q^j(i, t) \quad (5)$$

where $t_q^j(i, t)$ is time spent by the i^{th} BHP in the queue at node j till t since its arrival at t_a i.e., $t_q^j(i, t) = t - t_a$.

This computation is similar to that in Eq. 2 except that we do not include the processing time because it is already included in the computation of HWL. Note that the offset time is initially computed at the ingress according to

$$t_o = h * t_p + t_g + t_s + t_f \quad (6)$$

where h is number of hops to the destination, t_g is guard time between two bursts (if any), t_s is the switch configuration time, and t_f is a constant time added to the offset to relax the deadline for the BHP by accounting for any unforeseen delay along the path. If $t_f = 0$, any additional time taken by the BHP will lead to the $t_o^r = t_g + t_s$ before the destination causing burst loss. We assume that no other contention resolution technique is used which delays the BHP at the core node. We assume that t_p includes the time to find a feasible schedule for the data burst and to transmit the BHP. At any hop along the path, after the BHP is processed, the offset time is reduced by the time spent in queue and the time taken for the processing. The offset time carried by the BHP is updated to the residual offset time at each node.

To see the relation between the HWL metric and the initial offset time, we give an alternate computation of HWL from the initial offset time. Assume that the nodes traversed by i^{th} BHP are represented by an ordered list $1, 2, \dots, j, j+1, \dots, h$ where h is the number of hops to the destination. So at any core node $j+1$, HWL for the i^{th} BHP at time t , can be written as

$$L^{j+1}(i, t) = t_g + t_s + t_f - \sum_{k=0}^j t_q^k(i) - t_q^{j+1}(i, t), \quad 1 \leq j < h \quad (7)$$

where the summation term accounts for the queuing time at all the previous hops. We use the HWL value to fix the priorities for BHPs in the queue and select the BHP with lowest value of HWL.

B. Least per-Hop Worst-case Laxity First (LHWLF) Algorithm

In this section we describe the proposed service discipline for the control plane. To avoid the bias towards the BHPs corresponding to longer bursts as in the LLLRF algorithm, we do not give direct importance to the burst length in the computation of priorities. Further, using HWL instead of laxity directly also avoids the bias towards BHPs travelling on shorter paths. Note that HWL gives a tighter bound on the delay that any BHP can tolerate at a core node excluding the delay at the subsequent hops. If the delay suffered by the BHP at any node is such that the HWL given by Eq. 4, becomes lower than the fixed components of the offset time ($t_g + t_s$) it is impossible that the BHP reaches the destination before the data burst (see Eq. 7). This might happen due to unforeseen delays along the path or if the queuing time is larger than the t_f added to the initial offset time. In such cases, the BHP can be dropped from the queue so that the other BHPs can have

lower waiting time in the queue. We drop the BHP if the HWL goes lower than $t_g + t_s$ unlike the LLLRF algorithm where, if the laxity becomes zero, the BHP is assigned highest priority by replacing the laxity with 10^{-6} . This reduces the wastage of resources due to the processing of BHPs corresponding to those bursts which are eventually dropped before reaching the destination. It also reduces the queueing time for the other BHPs if the number of dropped BHPs is significantly high.

When the BHP enters the service queue (queue at the control processor) the processor puts the t_a in the time-stamp field of the BHP. When the BHP processor needs to pick up a BHP for service, it considers the BHP with least value of HWL which automatically selects the BHP with tightest deadline at that node. The algorithm used to select the BHP from the service queue is presented in Algorithm 1. The algorithm uses the average processing time of the BHP as t_p and gets the values for t_a, t_o^r, h_r and len for each BHP from the BHP itself. It then computes the time spent by the BHP in the queue, t_q . After deducting t_q from the residual offset time and also the required offset time to travel all the subsequent hops to the destination, the per-hop worst-case laxity is obtained. A lower value of HWL means a higher priority for the BHP because it means that it cannot be kept waiting any longer at the node. A BHP with the highest priority is picked by the algorithm. If any BHP has HWL less than $t_s + t_g$, then it is marked to be dropped from the queue because the data burst might overtake the BHP. If two or more BHPs have the same HWL value then the algorithm picks the BHP with larger length, to improve the throughput of the network. Finally, all the BHPs which are marked by the algorithm are dropped. By avoiding direct dependence on the length of the burst, we avoid the burst length-based unfairness mentioned in Section II-B. We also avoid the path length-based unfairness by giving larger importance to the tightness in the per-hop laxity which is invariant with the path length. Note that we achieve the fairness among BHPs travelling paths with different lengths because we subtract the residual offset time for the path to be travelled from the laxity.

C. Example to Illustrate the Working of the LHwLF Algorithm

In this section we explain the advantages of our algorithm with the help of an example. Consider four BHPs $X = (30, t, 3, 20)$, $Y = (110, t, 7, 110)$, $Z = (50, t, 4, 40)$ and $W = (30, t + t_p, 2, 70)$ in the queue in the same order as mentioned. Let $t_p = 10$. Fig. 3 shows the instances of the algorithm at different times after each BHP is processed. We consider, FIFO, LLLRF, and the LHwLF algorithms in this example. FIFO always selects the BHP at the top of the queue. The BHP with the lowest metric is selected by LHwLF algorithm while the one with the largest metric is selected by the LLLRF algorithm. The selected BHPs are shown with encircled metric values. Adjacent to the selected BHP, we also indicate the probability of successful transmission of the BHP along the path by tick mark. If the BHP is sure to be lost, we indicate it with a cross mark. All those BHPs that are selected by the service discipline which are lost subsequently

Algorithm 1 Algorithm to Select the BHP to be Serviced in the Queue

```

1: Input:  $t_p, Queue_n$ , and  $l_q$  /* BHP processing time, the set
   of BHPs in the queue at the core node  $n$ , and the length
   of the queue */
2: Output:  $bhp$  /* Selected BHP to be processed or null */
3: At core node  $n$ 
4:  $HWL_{min} = \infty, len_{max} = 0, result = null$  /*
   initialization of variables */
5: for each  $bhp_i = (t_o^r, t_a, h_r, len)$  in  $Queue_n$  do
6:   /*  $1 \leq i \leq l_q$  */
7:    $t = now()$  /* get the current time */
8:    $t_q^n(t) = t - t_a$  /* time spent by the  $bhp_i$  until  $t$  in the
   service queue at node  $n$  */
9:    $lax^n(i, t) = t_o^r - t_q^n(t)$  /* laxity until the BHP reaches
   the destination */
10:   $L^n(i, t) = lax^n(i, t) - h_r * t_p$  /* HWL at the node */
11:  if  $L^n(i, t) < t_g + t_s$  then
12:    /* If there is no additional laxity remaining */
13:     $mark - as - drop(bhp_i)$  /* function to mark a BHP
   to be dropped from the queue */
14:  else
15:    if  $HWL_{min} > L^n(i, t)$  then
16:       $HWL_{min} = L^n(i, t)$ 
17:       $len_{max} = len$ 
18:       $result = bhp_i$ 
19:    else
20:      if  $HWL_{min} == L^n(i, t)$  then
21:        if  $len_{max} < len$  then
22:           $len_{max} = len$ 
23:           $result = bhp_i$ 
24:        end if
25:      end if
26:    end if
27:  end if
28: end for
29: Drop all the BHPs from the queue marked by  $mark -
   as - drop$  function
30: Return  $result$ 

```

cause wastage of network resources. On the other hand the successfully delivered BHPs correspond to the gain in the throughput. The three set of tables show the BHP selected by each of the algorithms at the time shown in the figure. The metric value is recomputed at that time. For FIFO, the metric is shown as unity because FIFO only considers the arrival time. The value “large” indicates that the laxity is zero but is replaced by a small value so that the BHP is picked. The final table in the lower part of the figure shows the payload gain for the three algorithms which is obtained by taking the ratio of the cumulative lengths of the bursts transmitted to the cumulative length of all the four bursts. This is the amount of data which might be successfully delivered by each algorithm which affects the throughput. We can observe that the LHwLF algorithm has the highest payload gain among all of those

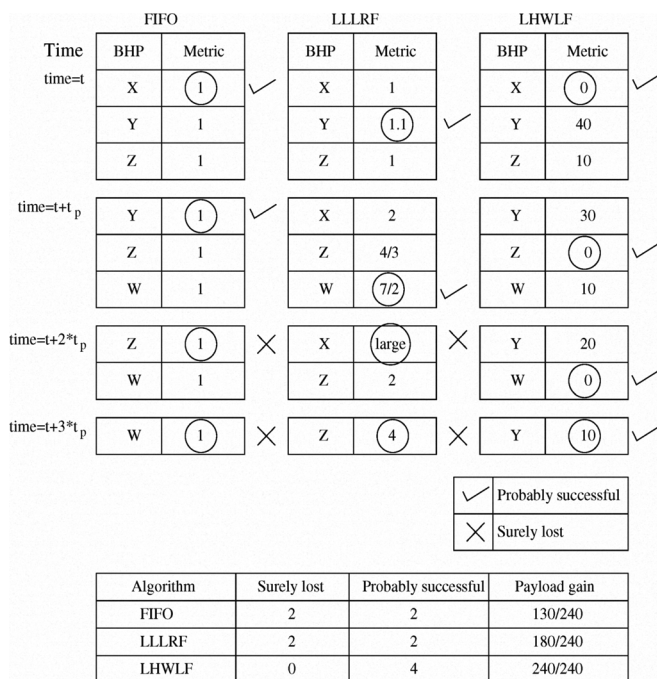


Fig. 3. Example

compared because it selects the BHP with highest probability to reach the destination as perceived at the node.

D. Complexity and Overheads

The complexity of the LHWLF is $O(N)$, where N is the number of BHPs in the queue. If we use the algorithm when the BHPs are put in the queue, using appropriate data structures and sorted lists the complexity can be reduced to $O(\log N)$ and $O(1)$ for dequeuing. The proposed algorithm requires local queue information and additional header fields in BHP, but these overheads are negligible. To remove the effect of this cost, the service discipline can be invoked parallelly with the BHP processing and transmission.

IV. SIMULATION RESULTS

A. Simulation Environment

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Bandwidth per wavelength	1.25Gbps
Propagation delay	1ms
BHP processing time	10 μ s
Average burst duration	720 μ s
Burst assembly time out	1s
Number of control channels	1
Number of data channels	128
Switch configuration time	100ms
Burst guard time	10 μ s
Simulation interval	50s

We used OIRC OBS-ns simulator for all the simulations [15]. Since the simulations are memory-intensive we ran the simulations on Vega Super-cluster [16] which has compute nodes based on HP Proliant DL160 G5 servers with Quad-core Intel Xeon E5472, 3.0 GHz processor and 32 GB RAM. We used the simulation parameters presented in the Table I [17]. We assume that the nodes have full wavelength conversion and LAUC-VF scheduling algorithm for the data plane. The traffic arrival is assumed to be Poisson distributed and the application traffic is UDP [18]. The queueing delay at the edge node is ignored and t_f is set to $0.2t_p$. We use JET signalling protocol, assume that the edge node has infinite buffer for the data bursts and the core node has infinite buffer for the BHPs. All the results are presented with 95% confidence level. We used the USANET topology shown in Fig. 4 for all the simulations. The other parameters are mentioned along with the results. We compare the proposed algorithm with LLLRF and FIFO service disciplines.

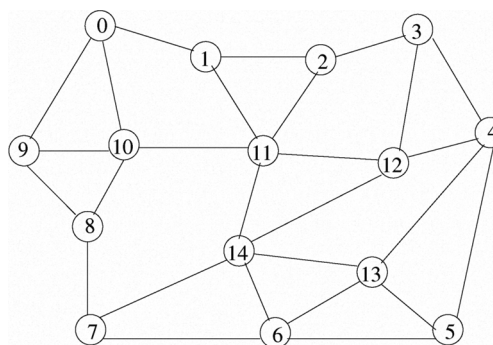


Fig. 4. USANET topology

B. Performance with Increasing Load

First, we compare the performance of the three service disciplines for increasing load values. We measure the throughput as the amount of data successfully received in the data plane at the destination in a unit time normalized with the link capacity. We also compare the BLP and the statistics of the BHPs dropped due to various reasons with increasing load. Fig. 5 shows the variation of the BLP with increasing load for all the three algorithms. We can see that the LHWLF algorithm has the lowest BLP among all the three. As the load increases, all the three algorithms have similar performance in terms of the BLP because the loss occurs due to the lack of feasible schedule in the data plane. In such a case BLP cannot be reduced by the service discipline alone. This can be seen from the composition of BHP losses shown in Fig. 6 which shows the percentage of losses from those dropped due to different reasons in the observed BLP. Here, the dropped BHPs are shown in three classes. The type of losses indicated by *drop 1* are those BHPs dropped from the service queue itself by the LHWLF algorithm. This is only applicable in case of our algorithm. The type of losses shown by *drop 2* are those that are dropped eventually because of insufficient offset time. These are prominent for FIFO and LLLRF algorithms.

These losses also lead to wastage of network resources. So at low loads the LHWLF algorithm improves the utilization by dropping the BHPs earlier along the path. At high load most of the losses occur due to the lack of a feasible schedule in the data plane which are indicated by *drop 3* in which case the performance of all the algorithms is similar. Fig. 7 shows the variation in the normalized throughput of the BHPs with increasing load. We compute the normalized throughput as the ratio of observed network throughput to the link capacity. The results show that our algorithm outperforms the others at all loads. The throughput saturates beyond a certain load due to the fixed capacity of the link.

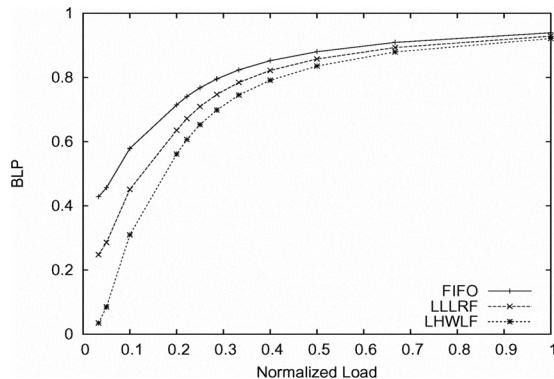


Fig. 5. Variation of BLP with the load

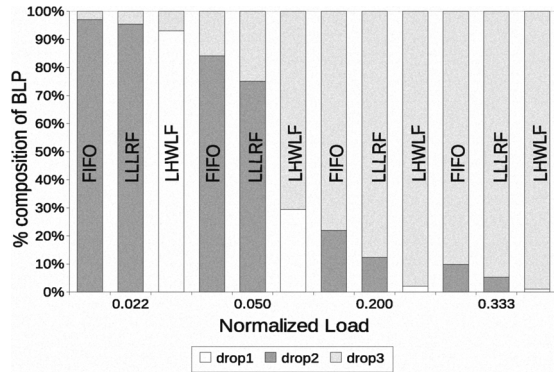


Fig. 6. Percentage of BHPs lost due to different reasons with load

C. Performance with Increasing Number of Data Wavelengths

Fig. 8 and Fig. 9 show the performance of all the algorithms in terms of the BLP and the throughput with increasing number of data wavelengths, respectively. We fixed the normalized load to 0.4 in this case. It can be seen that as expected the BLP decreases with the number of data channels but the LHWLF algorithm has lowest BLP among the three. The reduction in the BLP with the proposed algorithm is predominant with increasing number of data wavelengths because, the loss due to infeasible schedule is lower with larger wavelengths and the BLP occurs only due to design of service discipline. Similarly, the throughput with LHWLF algorithm is higher compared to

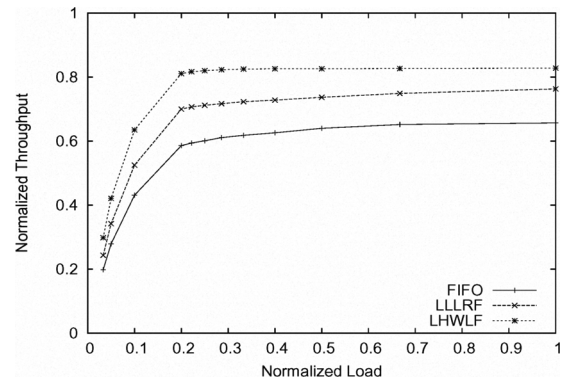


Fig. 7. Variation in the throughput with load

the others but is steady due to the effect of load saturation. Because the proposed algorithm schedules the BHPs based on the worst-case bound on the residual delay budget at the core node, it outperforms the others.

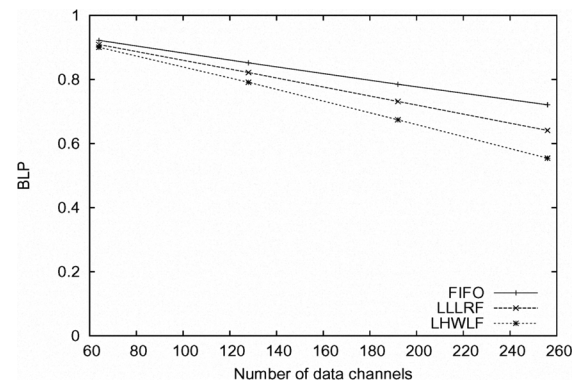


Fig. 8. Variation of BLP with the number of data channels

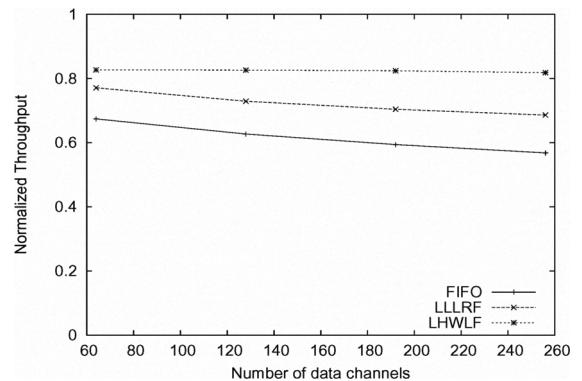


Fig. 9. Variation in the throughput with number of data channels

D. Fairness

As discussed in Section II-B, the proposed algorithm avoids the unfairness due to variability in the path length as well as the burst length. Fig. 10 and Fig. 11 demonstrate that the proposed algorithm is fair across different path lengths and burst lengths. The average path length travelled by the bursts in

the USANET topology in our simulations is about 3.8. We plot the average number of hops travelled by bursts for different loads when different service disciplines are used in the control plane. Results show that the average path length traveled by the bursts with LHWLF service discipline is close to the expected value of 3.8 especially for low loads. This demonstrates that the proposed algorithm gives fair treatment to all the bursts travelling along the paths with different lengths. On the other hand the average path length traveled by the bursts is smaller for FIFO and LLLRF algorithms. This demonstrates that the proposed algorithm provides fairness to flows traversing different path lengths. Similarly, Fig. 11 demonstrates that the service discipline is fair across bursts of different lengths. The average burst length used in the simulations is close to 900KB which is shown as the theoretical value in the plot. It can be seen that the LHWLF algorithm serves bursts such that the average burst length is closer to the expected value. The other algorithms always favour larger bursts which results in the average burst length to be large.

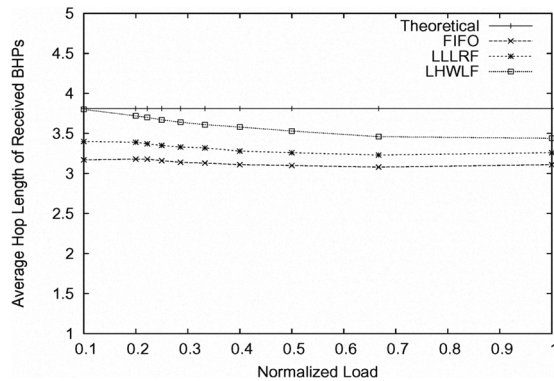


Fig. 10. Variation in the average number of hops travelled by successful BHPs with load

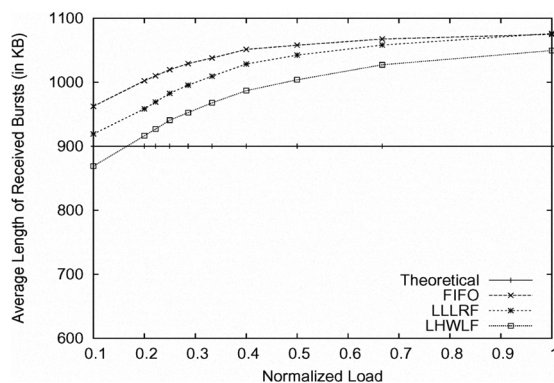


Fig. 11. Variation in the average burst length of bursts received with normalized load

We also varied the t_f from 0 to $0.4t_p$ to study the effect of relaxing the deadline or reducing the tightness in the laxity. It was observed that the FIFO and LLLRF algorithms perform badly with tight deadlines leading to higher BLP and lower throughput. However, the LHWLF algorithm has a constant

performance with any tightness in the deadline. With greater laxity, almost all the algorithms have the same performance in terms of BLP and the throughput. Due to space constraints we do not present these results here.

V. CONCLUSIONS

In this paper we studied the role of service discipline in the control plane to improve the performance of OBS networks. We proposed an efficient service discipline (LHWLF) that selects the BHP corresponding to the burst with tightest deadline at that hop. We demonstrated that the proposed service discipline improves the throughput and reduces the BLP in OBS networks. We also showed that the algorithm is fair in serving the BHPs travelling across different path lengths and the bursts with different lengths.

REFERENCES

- [1] Y. Chen, C. Qiao, and X. Yu, "Optical burst switching (OBS): A new area in optical networking research," *IEEE Network*, vol. 18, no. 3, pp. 16–23, May-Jun 2004.
- [2] J. P. Jue and V. M. Vokkarane, *Optical Burst Switched Networks*. Springer, 2005.
- [3] C. Qiao and M. Yoo, "Optical burst switching (OBS)- A new switching paradigm for optical Internet," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 69–84, March 1999.
- [4] A. Martinez, J. Aracil, and J.E. Lopez de Vergara, "Optimizing offset times in optical burst switching networks with variable burst control packets sojourn times," *Optical Switching and Networking*, vol. 4, no. 3-4, pp. 189–199, Nov 2007.
- [5] J. White, M. Zukerman, and H. L. Vu, "A framework for optical burst switching network design," *IEEE Communications Letters*, vol. 6, no. 6, pp. 268–270, June 2002.
- [6] J. Hernandez, J. Aracil, L. de Pedro, and P. Reviriego, "Analysis of blocking probability of data bursts with continuous-time variable offsets in single-wavelength OBS switches," *IEEE Journal of Lightwave Technology*, vol. 26, no. 12, pp. 1559–1568, June 2008.
- [7] L. de Pedro, J. Aracil, J. Hernandez, and J. Garcia-Dorado, "Analysis of the processing and sojourn times of burst control packets in optical burst switches," in *Proceedings of ONDM*, 2008, pp. 1–3.
- [8] Y. Xiong, M. Vandenhoute, and C. Cankaya, "Control architecture in optical burst-switched WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1838–1851, October 2000.
- [9] N. Barakat and T. E. Darcie, "Control-plane congestion in OBS networks," in *Proceedings of BROADNETS*, 2006, pp. 1–3.
- [10] J. Lin, Y. Chen, and A. M. K. Cheng, "On-line burst header scheduling in optical burst switching networks," in *Proceedings of AINA*, 2008, pp. 363–369.
- [11] M. Klinkowski, D. Careglio, and J. Sole-Pareta, "Modelling of control plane in OBS networks," in *Proceedings of ICTON*, 2007, pp. 109–112.
- [12] J. Li, C. Qiao, J. Xu, and D. Xu, "Maximizing throughput for optical burst switching networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1163–1176, Oct 2007.
- [13] H. Schulzrinne, J. Kurose, and D. Towsley, "An evaluation of scheduling mechanisms for providing best-effort real-time communications in wide-area networks," in *Proceedings of IEEE INFOCOM*, 1994, pp. 1352–1361.
- [14] C. S. R. Murthy and G. Manimaran, *Resource Management in Real-Time Systems and Networks*. MIT Press, 2001.
- [15] OIRC OBS-ns Simulator, <http://wine.icu.ac.kr/obsns>, (Last accessed in May 2009).
- [16] Vega Supercluster, <http://hpce.iit.ac.in/Vega.html>, (Last accessed in May 2009).
- [17] H. Guo, Z. Lan, J. Wu, Z. Gao, X. Li, J. Lin, and Y. Ji, "A testbed for optical burst switching network," in *Proceedings of OFC/NFOEC*, 2005, pp. 1–3.
- [18] M. Izal and J. Aracil, "On the influence of self-similarity on optical burst switching," in *Proceedings of GLOBECOM*, 2002, pp. 2308–2312.