Asian Journal of Information Technology 15 (22): 4648-4661, 2016
ISSN: 1682-3915

Priority Based Scheduling Based on Channel Utilization in OBS Networks

[1]A. Vishnu Priya and [2]P. Senthil Kumar
[1]Department of Computer Science and Engineering,
Anna university, Chennai, India
[2]Department of IT, SKR Engineering College, Chennai, India

Abstract: OBS (Optical Burst Switching) networks have more blocking probability with respect to burst movement. So it is challenging to schedule channels for OBS networks. Here we propose to develop Priority Based Scheduling architecture based On Channel Utilization in OBS Networks. Herein initially, we classify incoming packets by means of destination, loss and delay packets. We use Shaper architecture to determine the size and departure time of the bursts. Based on the time taken for a burst, we calculate expanded priority vector which considers CoS value of the job, Residual distance to the resource domain and a discrete value for the job size (burst length). We use this priority class to schedule the bursts using preemptive latest available unscheduled channel with void filling. Here, we preempt the overlapping bursts based on the priority and fill it with new resources. After scheduling the slots, we made channel utilization to choose the optimal channel with maximum defined channel utilization. Thus we determine the feasible data channel by performing bitwise AND between new BDP and register R of each data channel in parallel.

Key words: OBS, channel utilization, destination, burst length, void filling

INTRODUCTION

OBS networks: Optical Burst Switching (OBS) is one of the trusted solutions towards the next-generation optical internet with IP over WDM as the core architecture (Hu *et al*., 2004; Abe *et al*., 2005). Optical Burst Switching (OBS), a novel approach bringing together the best of optical circuit switching (wavelength routing) and optical packet switching, gained worldwide attention as a potentially bandwidth-efficient approach for future optical core networks in both academia and industries (Cankaya *et al*., 2003; Li *et al*., 2007; Xu *et al*., 2004). OBS has lower pre transmission latency comparing to OCS as it need limited or even no data delay at intermediate nodes as OCS and ensures efficient bandwidth utilization on a fiber link as OPS (Hu *et al*., 2004). OBS, a hybrid between optical circuit switching and optical packet switching, requires an optical technology level as the former whereas providing the flexibility and sub-wavelength granularity of the latter (Barakat and Sargent, 2005).

An ingress OBS node in an OBS network gathers together the incoming data for instance IP packets into (data) bursts and sends out a corresponding control packet generated for each burst. The control packet with control information like channel identification, destination node identification and the DB length and DB arrival time is delivered out-of-band and the burst is lead by an offset time. From the ingress node to egress node, control packet reserves necessary resources (for instance bandwidth on a desired output channel) at each intermediate node for the following burst, later an egress node will disassemble it into original packets (Charcranoon *et al*., 2003; Li *et al*., 2007; Xu *et al*., 2004). Several regular IP packets are buffered with same destination in the source edge routers to form a data burst. Then for each data burst, the control packet is generated by edge routers also. The control packet carries the control information of routing and switching of this data burst at intermediate nodes (Xu *et al*., 2003).

The advantages of OBS are the bandwidth usage rate optical buffer not needed. Most OBS architectures use one-way reservation, burst contention and burst loss may occur in core nodes resulting in reduced network performance (Abe *et al*., 2005; Barakat and Sargent, 2005).

Scheduling in OBS networks: OBS possess temporal and spatial separation between data (in a form of data burst) and control (including header information and routing information). In spatial separation, Data Burst (DB) and its control information, Burst Header Packet (BHP), move on different channels. Whereas in temporal separation, control information is converted to electronics and

Corresponding Author: A. Vishnu Priya, Department of Computer Science and Engineering, Anna university, Chennai, India

processed some time before its corresponding data burst reaching a core node where it is all-optically switched. A time gap namely offset time is introduced between the data burst and its control information for better realization and/or fiber delay lines are employed at the ingress ports of the optical matrix to increase a time budget for control information to be processed. Therefore, the coordination of data burst and its control information entity (a smaller packet) are to be accommodated for OBS scheduling making it a challenging task (Cankaya *et al*., 2003).

OBS networks unlike packet switched networks partition each data bursts from its control information (BHP) and sent them on different channels. Data channel is the channel having data bursts and control channel is those with BHPs. When BHP is received, resources are allocated by intermediate nodes according to its carried information. Once a BHP is transmitted to reserve network resources along the path, data burst transmission is deferred for some time known as offset time (Charcranoon *et al*., 2003).

The major issue in OBS networks is designing efficient algorithms for scheduling bursts (or more precisely their bandwidth reservation). An ideal scheduling algorithm could process a control packet fast enough before the arrival of the burst and yet could determine a suitable void interval (or a suitable combination of a FDL and an void interval) for the burst as long as its existence. Or else, it causes irrelevant discarding of burst either because of incomplete reservation before the burst arrival or simply due to lack of the scheduling algorithm's smartness in making the reservation (Xu *et al*., 2003).

The OBS scheduling algorithms may be of two types: Horizon based algorithms and void filling algorithms. Horizon based algorithms keep only the latest point in time for each wavelength used. Void filling enable assigning bursts between two scheduled bursts while void is large enough and arrival of burst in right moment. Void-filling algorithms are also called Just Enough Time (JET) algorithms since they reserve the wavelength for the accurate time of burst passing the node (Yuang *et al*., 2004).

The burst movement via OBS networks decreases its residual offset time and increases the blocking probability. This affects fairness and throughput of OBS networks. Hence more the burst moves, more likely it is to be blocked, resulting in higher blocking probability for the bursts with long route lengths. Also several bursts which already have utilized various network resources are only blocked not far from their destinations (Hu *et al*., 2004).

Literature review: Aydin *et al*. (2009) proposed an extended preemptive channel scheduling algorithm to minimize the gaps between bursts and increase the channel utilization. However, it also suffers from complexity.

Ramantas and Vlachos (2009) proposed a new TCP-aware burst assembly and burst scheduling scheme. A high priority/low latency class was utilized in the combing scheme for the sake of TCP packetsmarked as critical for TCP throughput evolution. The scheme enhances average TCP throughput and sustain fairness among all active flows. When their TCP window is increased, unnecessarily delays will be caused by short timer segmenting transmission, clipping DFL gain.

Ramantas *et al*. (2009) proposed a scheduling algorithm supporting QoS according to preemptions, controlled by a novel preemption policy. However, the frame loss ratio is increased.

Ayain *et al*. (2009) proposed novel algorithms for batch scheduling in OBS networks with different optimization criteria. Here the strong correlations among the multiple bursts were considered and the proposed interval graphs and min-cost circular flow techniques were used to achieve optimized network performance in terms of data loss rate in the network. However, the larger threshold increases the burst loss rate as well as the delay.

Wu *et al*. (2011) proposed an index-based parallel scheduler for Optical Burst Switching (OBS) networks to find feasible voids on different channels in parallel with O (1) time complexity, achieve the highest possible efficiency. The channel scheduler proposed comprises of two phases: the feasible void searching on all data channels and optimal void selecting among feasible voids found.

Figueiredo and DaFonseca (2011) proposed a batch scheduling algorithm for OBS networks with linear computational complexity. A transformation of the problem formulation is applied to enable the problem modeling as a job scheduling with identical machine type of problem. When the load increases, the maximal clicks size of the interval graphs also increases which explains the increase in blocking.

Netak *et al*. (2011) presented a novel approach for burst scheduling namely reverse scheduling. Here normal flow of void filling algorithm is reversed for utilizing channels capacity better. However the burst dropping probability increased.

Zhang *et al*. (2013) proposed a Max-CU-VF (Maximum Channel Utilization with Void Filling) channel

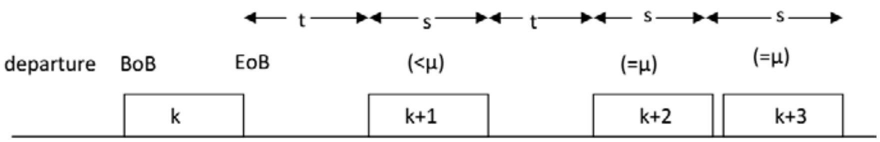

Fig. 1: Shaper departure procedure

scheduling algorithm for OBS (Optical Burst Switching) choosing the feasible data channel with the maximum channel utilization as the optimal one.

Nandi *et al.* (2009) proposed a new approach reducing burst loss and voids are efficiently utilized. The proposed algorithm, BFVF by considering the arrival data burst length and void length in scheduling, estimate the void utilization factor and then the new data burst was scheduled into a void channel with maximum void utilization factor. However the burst loss ratio is not much decreased.

Garg and Kaler (2009) proposed a novel modified Horizon scheduling algorithm with minimum reordering effects (MHS-MOE) in OBS networks. However, bit errors occurred were ignored.

Overview: Initially, incoming packets are classified by means of destination, loss and delay packets. Shaper architecture is used to determine the size and departure time of the bursts. Based on the time taken for a burst, we calculate expanded priority vector which considers CoS value of the job, Residual distance to the resource domain and a discrete value for the job size (burst length). This priority class is used to schedule the bursts using preemptive latest available unscheduled channel with void filling. Here, the overlapping bursts are preempted based on the priority and to fill it with new resources. After scheduling the slots, channel utilization is made to choose the optimal channel with maximum defined channel utilization. The feasible data channel is determined by performing bitwise AND between new BDP and register R of each data channel in parallel.

MATERIALS AND METHODS

Shaper system architecture: In any ingress node, incoming packets are classified based on their destination, loss and delay packets. Packets of same destination and loss class are assembled into a burst. Hence, a burst consists of packets of different delay classes.

A shaper can determine the sizes and departure time of bursts (Yuang *et al*., 2004). A burst is formed and departs when the burst size reaches μ (the maximum burst size) or the Burst Assembly Timer (BATr) (initially set as τ) expires.

A burst of size μ is generated and transmitted when total number of packets attains μ before the burst assembly time exceeds τ. Or else a burst of size less than μ is generated when BATr expires. BATr is initialized when it is activated or reset. It is activated when a system is changed from being idle to busy due to new packet arrivals. BATr is immediately reset when a burst departs leaving behind a nonempty queue.

In the (μ, τ) Shaper system, bursts are transported by one wavelength and forwarded via Optical Label Switched Path (OLSP). Shaper (μ, τ) is a discrete-time single-server queuing system where a time slot is equal to the transmission of a fixed-length packet. Here we assume the aggregate packet arrivals to follow a two-state Markov Modulated Bernoulli Process (MMBP) which enables batch arrivals at each state. There are two states H and L meant for high and low mean arrival rates.

MMBP consist of four parameters a, b, BA_H, BA_L. a (b) is the probability of changing from state H(L) to L(H) in a slot. BA_H (BA_L) is the probability of having a batch arrival at state H(L). The state change probability can be denoted as:

$$P_{i,j}, i, j \in \{H,L\} \text{ such that } P_{H,L} = 1 - P_{H,H} = a$$

And $P_{L,H} = 1 - P_{L,L} = b$. The batch sizes at state H and L possess distributions $b_H(m)$ and $b_L(m)$ with mean sizes $\overline{b_H}$ and $\overline{b_L}$. Let M be the mean arrival rate (packets per slot) (i.e., load) and B the burstiness of the arrival process can be:

$$B = \frac{BA_H \overline{b_H}}{M} = \frac{BA_H \overline{b_H}}{\frac{b}{a+b} \times BA_H \overline{b_H} + \frac{a}{a+b} \times BA_L \overline{b_L}} \quad (1)$$

There are five possible events occur in a slot sequentially are as follows:

- Arrival process state change
- Begin of burst departure
- Packet arrivals
- End of burst departure
- BATr activation/reset

At the beginning of slot, events and occurs. Event occurs at any time within a slot. Events and occur at the end of a slot. The departure process distribution consists of two parts:

Burst inter departure time (t): It is defined as interval from the end of a previous burst to the beginning of the following burst. It takes values which are integer multiples of a slot

Burst size (s) distributions:

Expanded Priority Vector (EPV) for contention resolution: A contention resolution scheme considering fairness between different service classes can support QoS between user jobs. We use Expanded Priority Vector (EPV) mechanism for contention resolution (Kantarci *et al.*, 2008). There are three digits in EPV as EPV_2, EPV_1, EPV_0 and it is coded into the passive bursts.

EPV_2: CoS value of the job: The jobs are classified based on their tolerate time. Less priority is given to longer tolerate time and vice versa. Here, let us take three different CoS values: 3 for Class-1 jobs, 2 for Class-2 jobs and 1 for Class-3 jobs.

EPV_1: Residual distance to the resource domain: Residual distance to the resource domain (EPV_1) is in terms of hop count. However, the bursts closer to the resource should be given higher priority than those farther from the resource domain. Hence, we set the EPV_1 field to a value of (N- hop_count) where N is the number of routers in the Grid.

EPV_0: A discrete value for the job size (burst length): Job size is also classified. Here we classify the bursts where the jobs are coded as long, medium and short size. As longer bursts have more probability to contend, to increase to ratio of handled job submissions, the shorter jobs are preferred when the jobs of the same class and the same residual distance contend. Hence we define three threshold values, namely T_s, T_m and T_l to determine whether a job is short, medium or long, respectively.

```
if (submitted job = short size)
then EPV0 = 3
else if medium size
then EPV0 = 2
otherwise
EPV0 = 1
```

A hexadecimal priority factor using these values is produced for each job at its release time and the time it is switched at the intermediate routers. The hexadecimal priority factor is as follows:

EPV_2	EPV_1	EPV_0

Fig. 2: Three digits of EPV

$$EPV_{factor} = \sum_{i=0}^{2} 16^i \times EPV_i \quad (2)$$

where, the factor 16 is determined empirically based on the topology used in the grid. At each intermediate router, this EPV Factor parameter is re-computed since the distance traveled changes.

For an incoming passive burst, an intermediate router searches for an unscheduled timeslot in each wavelength. If it could not find an available timeslot, it detects contention on the fiber and then the router discards the job with lowest EPV Factor value. The tolerate time of the discarded job is updated and dilated through another outgoing port of the router. If the dilated path violates the tolerate time, then the source node is informed that the job submission is blocked.

EPV_1 and EPV_0 fields support fairness guarantee among the jobs belonging to the same priority level. Therefore, this contention resolution scheme also leads to a decrease in the blocking priority of Class-2 jobs together with Class-1 jobs.

PLAUC-VF (Preemptive Latest Available Unscheduled Channel with Void Filling): PLAUC-VF is the modification of LAUC-VF in which the scheduler keeps track the free periods in each channel (Ramantas and Vlachos, 2009). LAUC-VF store only three time values per channel and makes two comparisons to decide whether a new burst can be scheduled in a channel or not. Preemption adds flexibility to the burst scheduling process enabling the re-arrangement of already scheduled bursts. PLAUC-VF stores burst length, class of service as well as unique burst identifier for all scheduled bursts. PLAUC-VF can track of and preempt the two most recent scheduled bursts.

The channel selection is same as that of LAUC-VF, assuring on average the time complexity remains low. If channel selection fails, more computationally demanding preemption phase follows. On arrival of a a new reservation request, the preemption capable scheduling unit follows these steps:

- It scans all channels for an idle period to schedule the burst

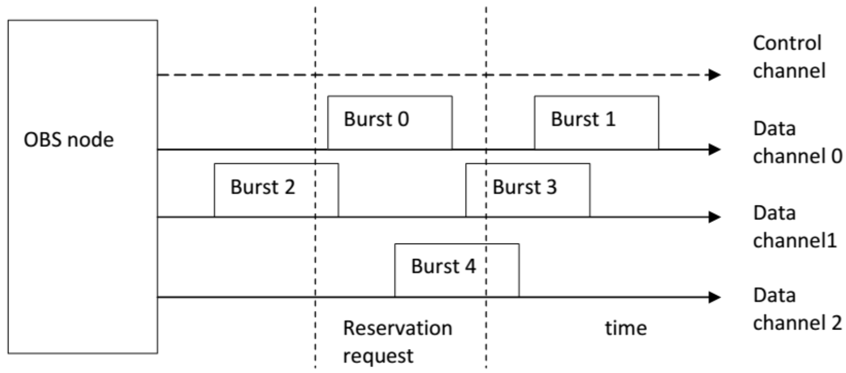

Fig. 3: Illustration of PLAUC-VF algorithm

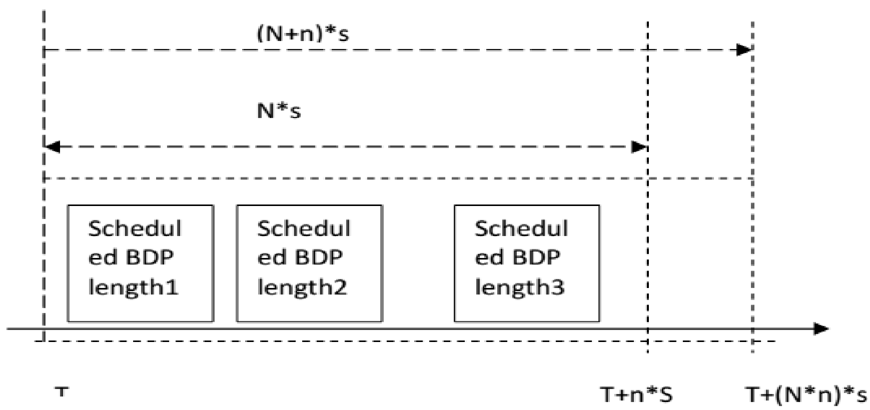

Fig. 4: Illustrating calculation of CU

- If voids occur in more than one channel, the one which minimizes the remaining idle period is chosen
- If no voids occur, there will be at least one overlapping burst in each channel. Then the scheduling channel iterates over overlapping bursts and decides whether one of them has to be preempted so as to free resources for the newly arrived burst
- The decision is based on the priority class that the burst belongs to as per Eq. 2

In Fig. 3, burst 0 belongs to normal priority class. Whereas the reservation request is of high priority class, then burst 0 will be preempted and the new burst would be scheduled in its place.

Channel utilization scheduling algorithm: The channel utilization of a channel is defined as the total length of scheduled buffer data packets in an observing time window whose window length is fixed (Zhang *et al*., 2013). It chooses the optimal channel with the maximum defined CU value from feasible data channels. Here the bitwise operation is applied. Here the time window is divided into N slots with same fixed length with certain conditions:

- Slot size $s < T_{BDP\_Min}$
- $N > (T_{off\_Max} + T_{BDP\_Max})/s$

$$> (T_{off\_Max} + T_{BDP\_Max})/T_{BDP\_Min} \quad (3)$$

Where:

T_{off_Max} = Maximum offset time

T_{BDP_Max} and T_{BDP_Max} = Time lengths of longest and shortest BDPs

Let current time T be the start of each scheduling, i.e., the beginning of time window. When Eq. 3 becomes true, every BDPs after current time are included in time window. The CU value will not vary with range of time window.

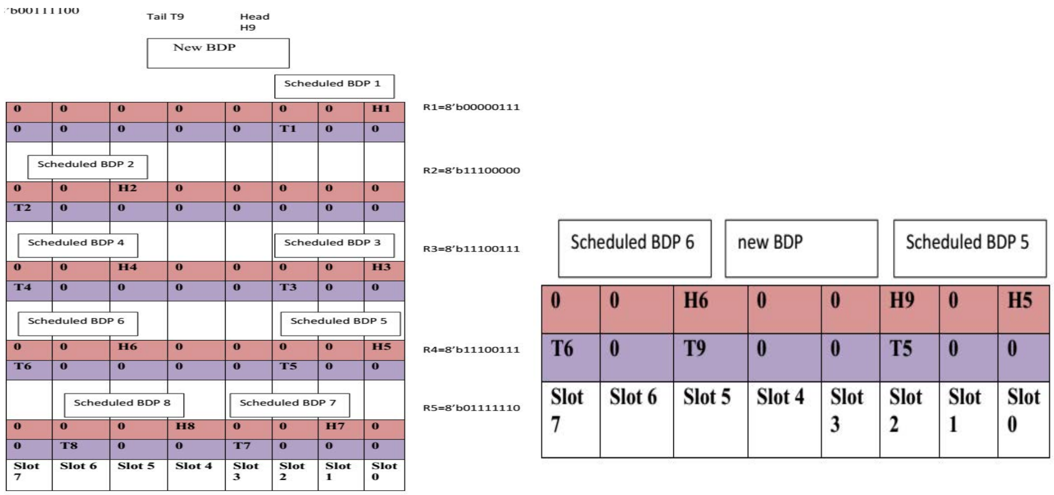

Fig 5: Data structure of scheduling scheme when a new BDP takes more than 2 slots (a) Before scheduling (B) after Scheduling

Consider N bit binary register R, an N-entry start time table and an N-entry end time table on each data channel in the scheduler with respect to N time slots. A register R records the occupation state of the slots on a data channel. If a part of scheduled BDP located in a particular slot say j, then jth bit will be set as logical 1 otherwise it is set as 0.

The start and the end time of a scheduled BDP are recorded in the entries of the start and the end time tables corresponding to the slots where the head and the tail of the BDP are located, respectively:

$$\text{Head} = \left\lfloor \frac{T_a - T_i}{s} \right\rfloor \tag{4}$$

$$\text{Tail} = \left\lfloor \frac{T_d - T_i}{s} \right\rfloor \tag{5}$$

Where:

T_a = Arrival time of new BDP
T_d = Departure time of new BDP
T_i = Start of observing time window and
$\lfloor . \rfloor$ = A rounding down operation

Max-CU-VF consists of the following steps:

- Find the index number of slots where head and tail of new BDP located
- Generate code to show which slots would be covered by new BDP by

$$\text{New BDP} = ((8'\text{h}01 << \text{Tail}) - (8'\text{h}01 << \text{Head})) \,|\, (8'\text{h}01 << \text{Tail}) \tag{6}$$

Where:

<< = Left shift operation
| = Bitwise or manipulation

Figure 5 shows example of data structure and depicts the head, tail of a new BDP:

- Find out feasible data channels by operating bitwise AND between new BDP and register R of each data channel in parallel
- There are five cases in resulting F to determine all feasible data channels for new BDP
- If a single bit of result F = logic 1 and the head of new BDP is just located in the corresponding slot, i.e., head of the new BDP and the tail of a previously scheduled BDP are in the same slot

Here the proposed scheduler directly picks up the end time of the scheduled BDP from the end time table and compares it with the start time of new BDP to make more accurate judgment for the object of utilizing the bandwidth better.

If the end time of the scheduled BDP is less than the start time of new BDP, the channel is feasible. Otherwise, the channel is not suitable.

If only a bit of the is logic 1 and the tail of new BDP is located in the corresponding slot which implies the tail of the new BDP and the head of a scheduled BDP are in the same slot indicated by the logic 1 of (the second data channel in Fig. 5 and is 8'b00100000).

Here, the start time of scheduled BDP is read out from the start time table and compared with the end time of new BDP to judge the data channel is feasible or not.

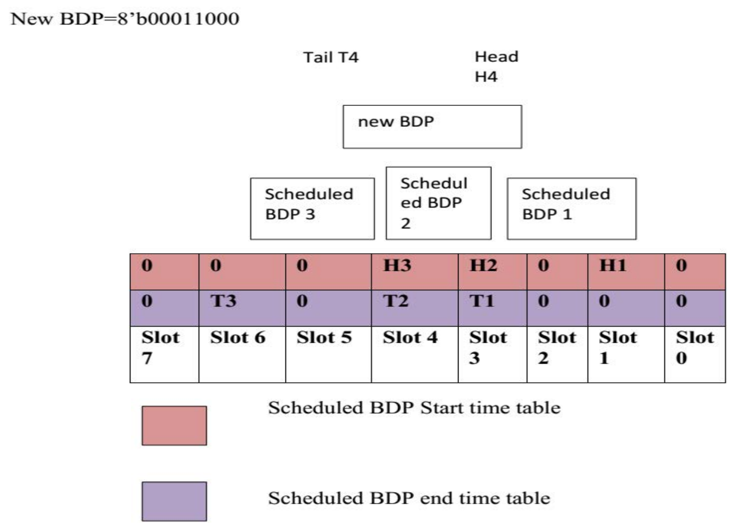

Fig. 6: Special case whennew BDP takes >2 slots

If two bits of are logic 1 and the head and tail of new BDP are located in the two corresponding slots, respectively i.e., the new BDP may collide with two previously scheduled BDPs at its head and/or tail part. The case is divided into two sub cases in terms of the number of slots occupied by the new BDP further.

Case 1: When the new BDP occupies more than two slots, extract the end time of scheduled BDP before new BDP and the start time of the scheduled BDP after new BDP from related time tables of the two slots corresponding to the two logic 1 bits in. After that, compare them with the start and the end time of the new BDP respectively to determine whether the channel is feasible or not for the new BDP.

Case 2: When new BDP takes up only two slots as in Fig. 6, the will have the same value. The channel is feasible only when the values of slot3 in the start time table and slot4 in the end time table are zero with and T1<H4 and T4<H3. Obviously, new BDP overlaps with scheduled BDP2.

- F = 0, i.e., the new BDP does not contend with any scheduled BDP on the data channel. The data channel is feasible obviously
- F is the value except for above four cases. Here, the data channel is unusable. The fifth data channel in Figure 6 shows an example of the cases, where is 8'b00111100

Thus, the scheduled data channel is utilized effectively.

Table 1: Simulation parameters

Parameters	values
No. of nodes	10
Area size	500 X 500
Mac	IEEE 802.11
Simulation time	3,5,7,9 and 11sec
Traffic source	CBR
Burst size	10000, 20000, 30000, 40000 and 50000
Load	50000, 60000, 70000, 80000, 90000 and 100000

Overall algorithm:

Step 1: initially, we classify incoming packets by means of destination, loss and delay packets.
Step 2: Shaper (μ, ρ) is a discrete-time single-server queuing system where a time slot is equal to the transmission of a fixed-length packet. We use Shaper architecture to determine the size and departure time of the bursts.
Step 3: Based on the time taken for a burst, we calculate Expanded Priority Vector which considers CoS value of the job, Residual distance to the resource domain and a discrete value for the job size (burst length).
Step 4: We then used this priority class to schedule the bursts using Preemptive Latest Available Unscheduled Channel with Void Filling. Here we preempted the overlapping bursts based on the priority and fill it with new resources.
Step 5:After scheduling the slots, we made channel utilization to choose the optimal channel with maximum defined channel utilization. Thus we determine the feasible data channel by performing bitwise AND between new BDP and register R of each data channel in parallel.

RESULTS AND DISCUSSION

Simulation model and parameters: The Network Simulator (NS2) is used to simulate the proposed architecture. In the simulation, the simulated traffic is Constant Bit Rate (CBR). OBS network is used here. Number of edge nodes is three and seven core nodes are used. The simulation settings and parameters are summarized in Table 1.

Performance metrics: The proposed Priority Based Scheduling based on Channel Utilization (PBSCU) is

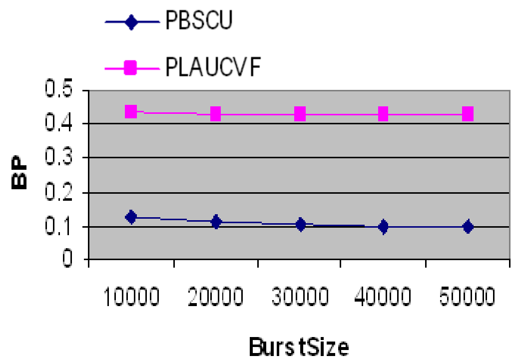

Fig. 7: Burst size vs. blocking probability

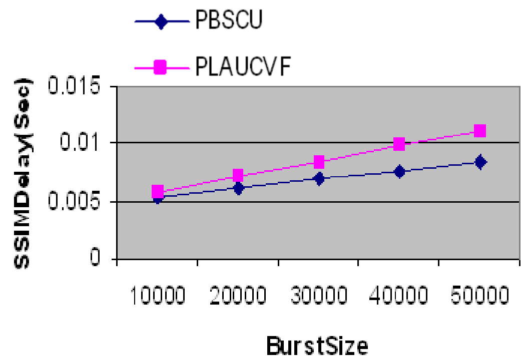

Fig. 9: Burst size vs. burst delay

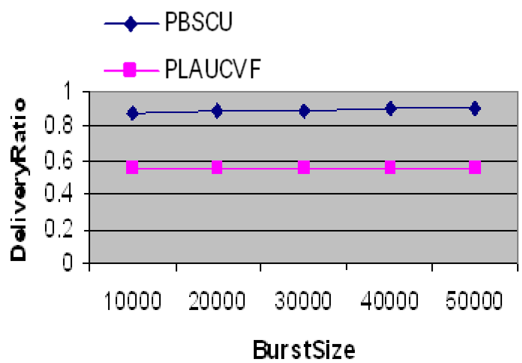

Fig. 8: Burst size vs. delivery ratio

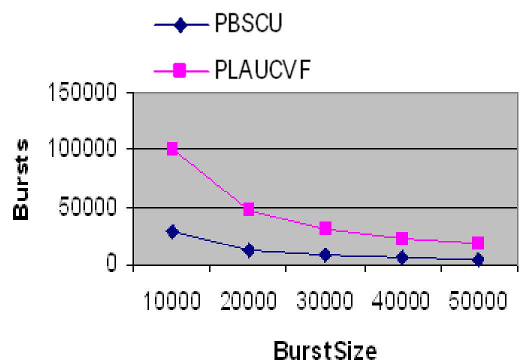

Fig. 10: Burst size vs. burst drop

compared with the PLAUCVF technique []. The performance is evaluated mainly, according to the following metrics.

Packet delivery ratio: It is the ratio between the number of packets received and the number of packets sent.

Burst drop: It refers the average number of bursts dropped during the transmission.

Burst delay: It is the amount of time taken by the nodes to transmit the data packets.

Based on burst size: In our first experiment we vary the Burst size as 10000, 20000, 30000, 40000 and 50000. Figure 7 shows the blocking probability of PBSCU and PLAUCVF techniques for different burst size scenario. We can conclude that the blocking probability of our proposed PBSCU approach has 75% of less than PLAUCVF approach. Figure 8 shows the delivery ratio of PBSCU and PLAUCVF techniques for different burst size scenario. We can conclude that the delivery ratio of our proposed PBSCU approach has 38% of higher than PLAUCVF approach.

Figure 9 shows the burst delay of PBSCU and PLAUCVF techniques for different burst size scenario. We can conclude that the burst delay of our proposed PBSCU approach has 17% of less than PLAUCVF approach.

Figure 10 shows the burst drop of PBSCU and PLAUCVF techniques for different burst size scenario. We can conclude that the burst drop of our proposed PBSCU approach has 75% of less than PLAUCVF approach.

Figure 11 shows the SSIM delay of PBSCU and PLAUCVF techniques for different burst size scenario. We can conclude that the SSIM delay of our proposed PBSCU approach has 17% of less than PLAUCVF approach. Figure 12 shows the SSIM throughput of PBSCU and PLAUCVF techniques for different burst size scenario.

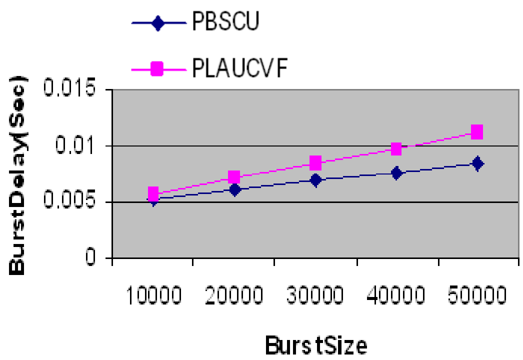

Fig. 11: Burst size vs. SSIM delay

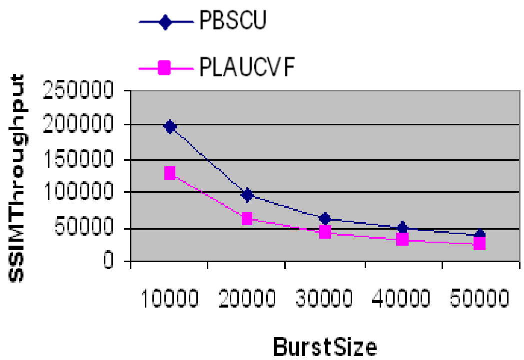

Fig. 12: Burst size vs. SSIM throughput

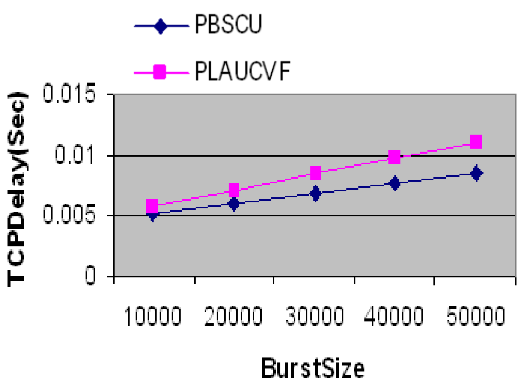

Fig. 13: Burst size vs. TCP delay

We can conclude that the SSIM throughput of our proposed PBSCU approach has 36% of higher than PLAUCVF approach.

Figure 13 shows the TCP delay of PBSCU and PLAUCVF techniques for different burst size scenario. We can conclude that the TCP delay of our proposed PBSCU approach has 17% of less than PLAUCVF approach.

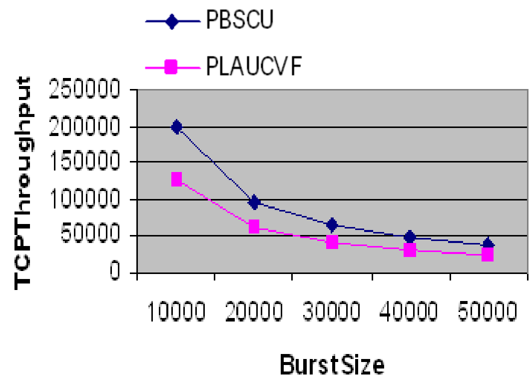

Fig. 14: Burst size vs. TCP throughput

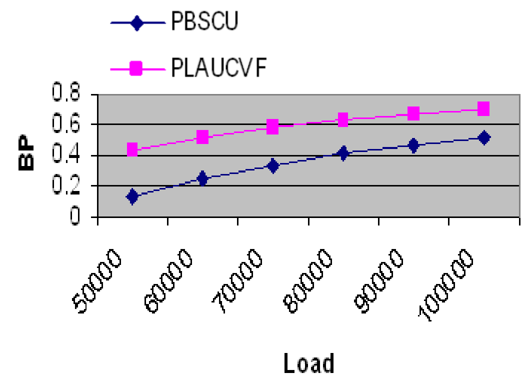

Fig 15: Load vs. blocking probability

Figure 14 shows the TCP throughput of PBSCU and PLAUCVF techniques for different burst size scenario. We can conclude that the TCPThroughput of our proposed PBSCU approach has 36% of higher than PLAUCVF approach.

Based on load: In our second experiment we vary the load values as 50000, 60000,70000,80000,90000 and 100000. Figure 15 shows the blocking probability of PBSCU and PLAUCVF techniques for different load scenario. We can conclude that the blocking probability of our proposed PBSCU approach has 43% of less than PLAUCVF approach.

Figure 16 shows the delivery ratio of PBSCU and PLAUCVF techniques for different load scenario. We can conclude that the delivery ratio of our proposed PBSCU approach has 38% of higher than PLAUCVF approach.

Figure 17 shows the burst delay of PBSCU and PLAUCVF techniques for different load scenario. We can conclude that the burst delay of our proposed PBSCU approach has 9% of less than PLAUCVF approach.

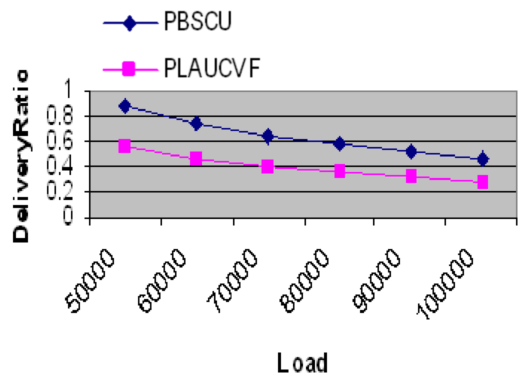

Fig. 16: Load vs. delivery ratio

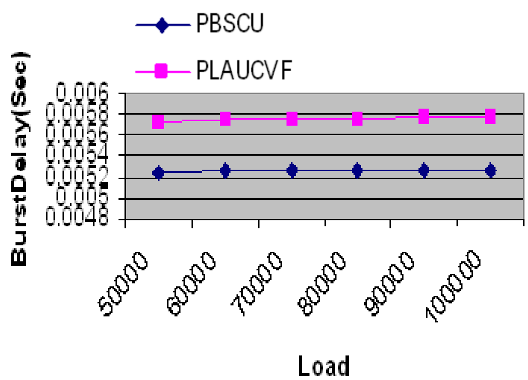

Fig. 17: Load vs. burst delay

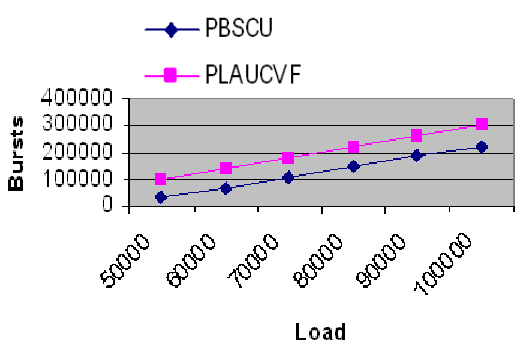

Fig. 18: Load vs. burst drop

Figure 18 shows the burst drop of PBSCU and PLAUCVF techniques for different load scenario. We can conclude that the burst drop of our proposed PBSCU approach has 43% of less than PLAUCVF approach.

Figure 19 shows the SSIM delay of PBSCU and PLAUCVF techniques for different load scenario. We can conclude that the SSIM delay of our proposed PBSCU approach has 9% of less than PLAUCVF approach.

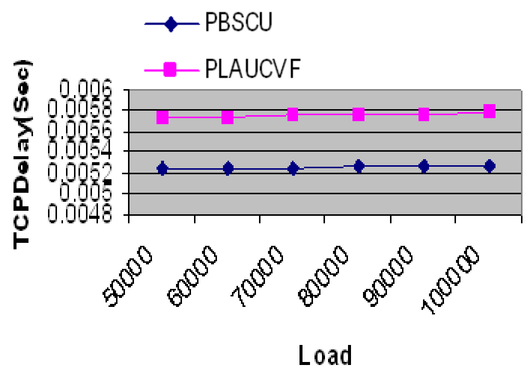

Fig. 19: Load vs. SSIM delay

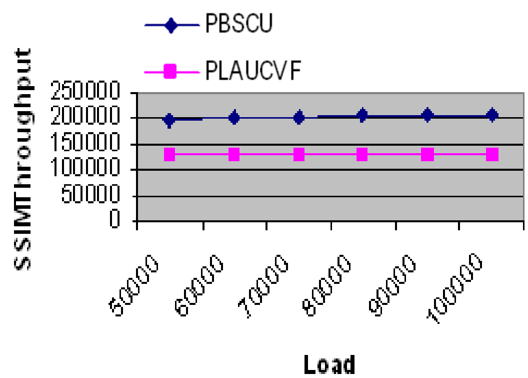

Fig. 20: Load vs. SSIM throughput

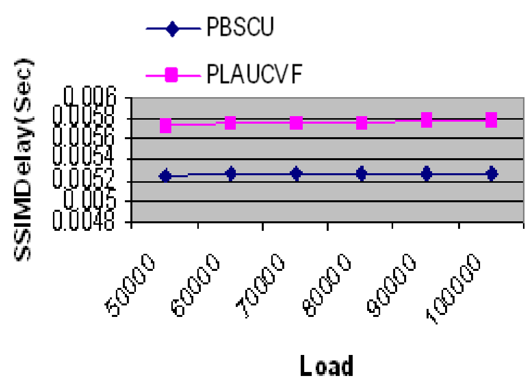

Fig. 21: Load vs. TCP delay

Figure 20 shows the SSIM throughput of PBSCU and PLAUCVF techniques for different load scenario. We can conclude that the SSIMThroughput of our proposed PBSCU approach has 37% of higher than PLAUCVF approach.

Figure 21 shows the TCP delay of PBSCU and PLAUCVF techniques for different load scenario. We can

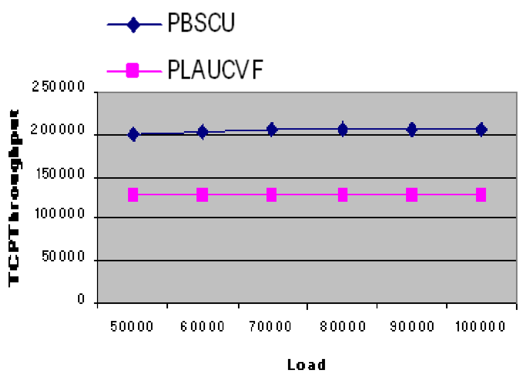

Fig. 22: Load vs. TCP throughput

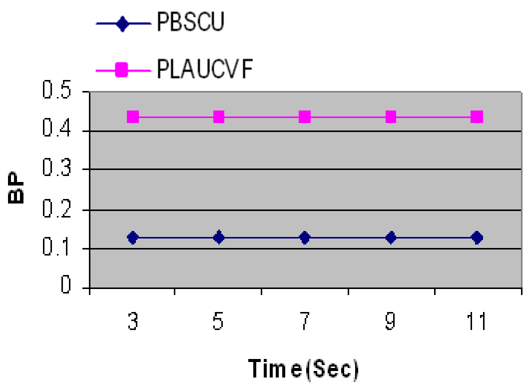

Fig. 23: Time vs. blocking probability

conclude that the TCP delay of our proposed PBSCU approach has 9% of less than PLAUCVF approach.

Figure 22 shows the TCP throughput of PBSCU and PLAUCVF techniques for different load scenario. We can conclude that the TCPThroughput of our proposed PBSCU approach has 37% of higher than PLAUCVF approach.

Based on simulation time: In our third experiment we vary the simulation time as 3,5,7,9 and 11sec. Figure 23 shows the blocking probability of PBSCU and PLAUCVF techniques for different time scenario. We can conclude that the blocking probability of our proposed PBSCU approach has 71% of less than PLAUCVF approach.

Figure 24 shows the delivery ratio of PBSCU and PLAUCVF techniques for different time scenario. We can conclude that the delivery ratio of our proposed PBSCU approach has 37% of higher than PLAUCVF approach. Figure 25 shows the burst delay of PBSCU and PLAUCVF techniques for different time scenario. We can conclude that the burst delay of our proposed PBSCU approach has 9% of less than PLAUCVF approach.

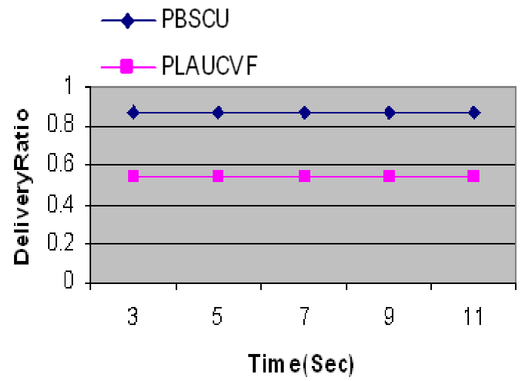

Fig. 24: Time vs. delivery ratio

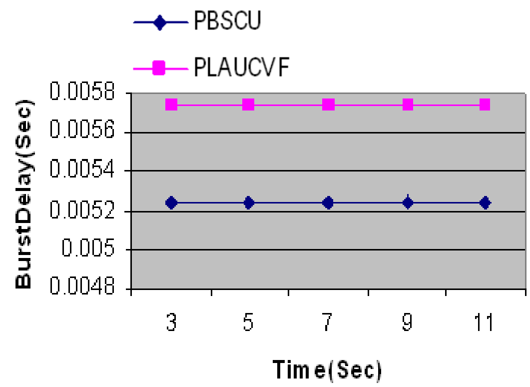

Fig. 25: Time vs. burst delay

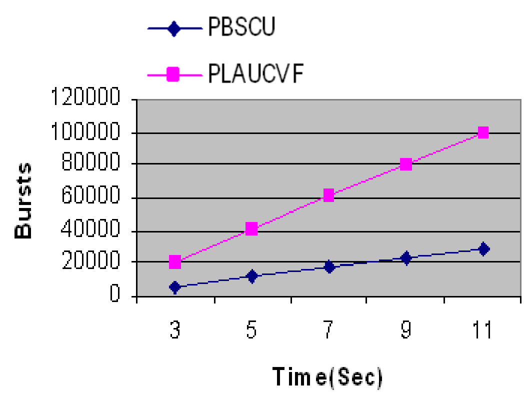

Fig. 26: Time vs. burst drop

Figure 26 shows the burst drop of PBSCU and PLAUCVF techniques for different time scenario. We can conclude that the burst drop of our proposed PBSCU approach has 71% of less than PLAUCVF approach.

Figure 27 shows the SSIMDelay of PBSCU and PLAUCVF techniques for different time scenario. We can

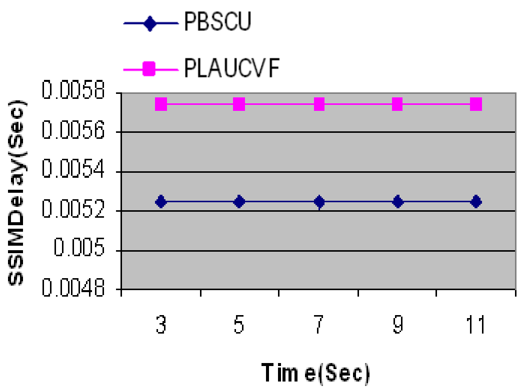

Fig. 27: Time vs. SSIM delay

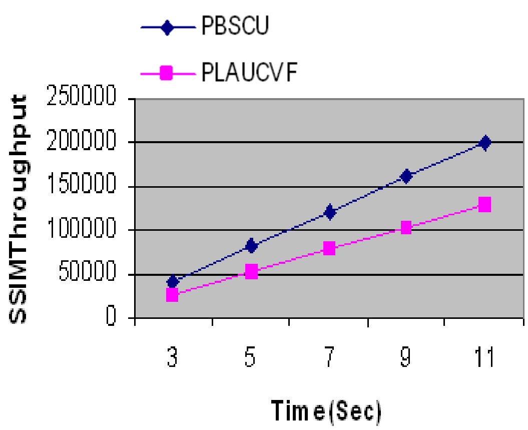

Fig. 28: Time vs. SSIM throughput

conclude that the SSIM delay of our proposed PBSCU approach has 9% of less than PLAUCVF approach.

Figure 28 shows the SSIM throughput of PBSCU and PLAUCVF techniques for different time scenario. We can conclude that the SSIM throughput of our proposed PBSCU approach has 36% of higher than PLAUCVF approach.

Figure 29 shows the TCP delay of PBSCU and PLAUCVF techniques for different time scenario. We can conclude that the TCP delay of our proposed PBSCU approach has 9% of less than PLAUCVF approach.

Figure 30 shows the TCP throughput of PBSCU and PLAUCVF techniques for different time scenario. We can conclude that the TCP throughput of our proposed PBSCU approach has 36% of higher than PLAUCVF approach.

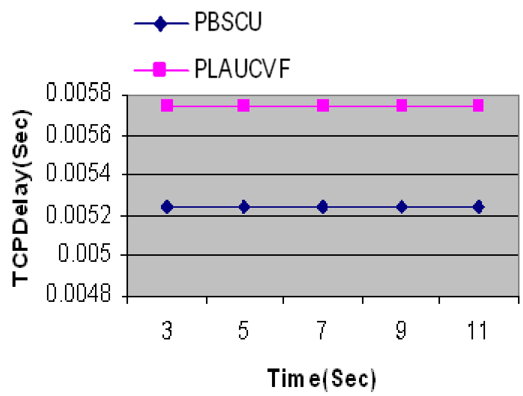

Fig. 29: Time vs. TCP delay

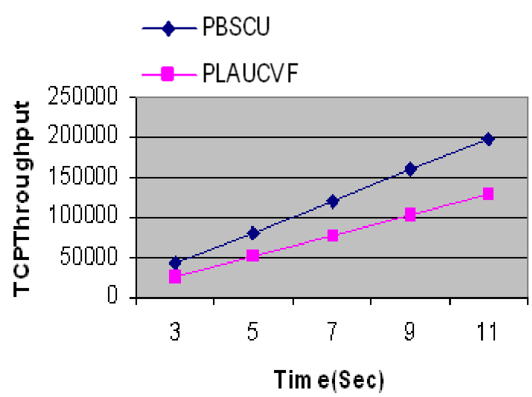

Fig. 30: Time vs. TCP throughput

CONCLUSION

OBS (Optical Burst Switching) networks have more blocking probability with respect to burst movement. So, it is challenging to schedule channels for OBS networks. Here, we proposed priority based scheduling architecture based on channel utilization in OBS networks. Here, in initially, we classified incoming packets by means of destination, loss and delay packets. Shaper (μ, σ) is a discrete-time single-server queuing system where a time slot is equal to the transmission of a fixed-length packet. We used Shaper architecture to determine the size and departure time of the bursts. Based on the time taken for a burst, we calculated Expanded priority vector which considers CoS value of the job, Residual distance to the resource domain and a discrete value for the job size (burst length). We then used this priority class to schedule the bursts using preemptive latest available unscheduled channel with void filling. Here, we preempted the overlapping bursts based on the priority and fill it with new resources. After scheduling the slots, we made channel utilization to choose the optimal channel with maximum defined channel utilization. Thus, we determined the feasible data channel by performing bitwise AND between new BDP and register R of each data channel in parallel.

REFERENCES

Abe, T., H. Pan, H. Tanida, Y.B. Choi and H. Okada, 2005. A feedback-based contention resolution mechanism for slotted optical burst switching. Proceedings of the 2nd International Conference on Broadband Networks, October 7, 2005, IEEE, New York, USA., ISBN: 0-7803-9276-0, pp: 306-309.

Aydin, M.A., T. Atmaca, O.C. Turna, H. Zaim and T. Eido, 2009. Regulative wavelength grouping approach for QoS in OBS networks. Proceedings of the Fifth Advanced International Conference on Telecommunications (AICT'09), May 24-28, 2009, IEEE, New York, USA., ISBN: 978-1-4244-3840-2, pp: 254-260.

Barakat, N. and E.H. Sargent, 2005. Achieving locally minimal burst blocking in OBS networks with delayed scheduling. Proceedings of the 2nd International Conference on Broadband Networks, October 7, 2005, IEEE, New York, USA., ISBN: 0-7803-9276-0, pp: 1093-1095.

Cankaya, H.C., S. Charcranoon and T.S. El-Bawab, 2003. A preemptive scheduling technique for OBS networks with service differentiation. Proceedings of the IEEE Conference on Global Telecommunications (GLOBECOM'03), December 1-5, 2003, IEEE, New York, USA., ISBN: 0-7803-7974-8, pp: 2704-2704.

Charcranoon, S., T.S. ElBawab, H.C. Cankaya and J.D. Shin, 2003. Group-scheduling for Optical Burst Switched (OBS) networks. Proceedings of the IEEE Conference on Global Telecommunications (GLOBECOM'03), December 1-5, 2003, IEEE, New York, USA., ISBN: 0-7803-7974-8, pp: 2745-2749.

Figueiredo, G.B. and N.L. DaFonseca, 2011. Algorithm with linear computational complexity for batch scheduling in OBS networks. Proceedings of the 2011 IEEE International Conference on Communications (ICC), June 5-9, 2011 IEEE, New York, USA., ISBN: 978-1-61284-232-5, pp: 1-6.

Garg, A.K. and R.S. Kaler, 2009. A novel scheduling algorithm for optical burst switched networks. J. Microwaves Optoelectron. Electromagn. Appl. (JMOe.), 8: 51-58.

Hu, Y., Y. Ji and Z. Gao, 2004. A new wavelength scheduling algorithm for achieving fairness in OBS networks. Proceedings of the IEEE Workshop on IP Operations and Management, October 11-13, 2004, IEEE, New York, USA., ISBN: 0-7803-8836-4, pp: 157-159.

Kantarci, B., S. Oktug and D.T. Altilar, 2008. Prioritized contention resolution scheme for grid services over OBS networks. Proceedings of the 6th Annual IEEE Conference on Communication Networks and Services Research, May 5-8, 2008, IEEE, New York, USA., ISBN: 978-0-7695-3135-9, pp: 234-241.

Li, J., C. Qiao, J. Xu and D. Xu, 2007. Maximizing throughput for optical burst switching networks. IEEE-ACM. Trans. Networking (TON.), 15: 1163-1176.

Nandi, M., A.K. Turuk, D.K. Puthal and S. Dutta, 2009. Best fit void filling algorithm in optical burst switching networks. Proceedings of the 2009 2nd International Conference on Emerging Trends in Engineering and Technology, (ICETET), December 16-18, 2009, IEEE, New York, USA., pp: 609-614.

Netak, L.D., G.V. Chowdhary, V.N. Suryawanshi and J. Borade, 2011. Reverse scheduling approach for burst loss minimization in WDM OBS based networks. Proceedings of the 2011 2nd International Conference on Computer and Communication Technology (ICCCT), September 15-17, 2011, IEEE, New York, USA., ISBN: 978-1-4577-1385-9, pp: 517-523.

Ramantas, K. and K. Vlachos, 2009. A combined TCP aware scheduling and assembly scheme for OBS networks. Proceedings of the 2009 Sixth International Conference on Broadband Communications Networks and Systems, September 14-16, 2009, IEEE, New York, USA., ISBN: 978-963-9799-49-3, pp: 1-6.

Ramantas, K., T.R. Vargas, J.C. Guerri and K. Vlachos, 2009. A preemptive scheduling scheme for flexible QoS provisioning in OBS networks. Proceedings of the 2009 Sixth International Conference on Broadband Communications Networks and Systems, September 14-16, 2009, IEEE, New York, USA., ISBN: 978-963-9799-49-3, pp: 1 6-10.4108/ICST. BROADNETS2009.7205.

Wu, G., T. Zhang, J. Chen, X. Li and C. Qiao, 2011. An index-based parallel scheduler for optical burst switching networks. J. Lightwave Technol., 29: 2766-2773.

Xu, J., C. Qiao, J. Li and G. Xu, 2003. Efficient channel scheduling algorithms in optical burst switched networks. Proceedings of the IEEE Twenty-Second Annual Joint Conference on Computer and Communications INFOCOM 2003. IEEE Societies, March 30-April 1, 2003, IEEE, New York, USA., ISBN: 0-7803-7752-4, pp: 2268-2278.

Xu, J., C. Qiao, J. Li and G. Xu, 2004. Efficient burst scheduling algorithms in optical burst-switched networks using geometric techniques. IEEE. J. Selected Areas Commun., 22: 1796-1811.

Yuang, M.C., P.L. Tien, J. Shih and A. Chen, 2004. QoS scheduler-shaper for optical coarse packet switching IP-over-WDM networks. IEEE. J. Selected Areas Commun., 22: 1766-1780.

Zhang, T., G. Wu, X. Li and J. Chen, 2013. A High speed scheduler with a novel scheduling algorithm for optical burst switching networks. J. Lightwave Technol., 31: 2844-2850.