

Received November 5, 2021, accepted November 24, 2021, date of publication December 20, 2021,  
date of current version December 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3132578

# Modified Optical Burst Switching (OBS) Based Edge Node Architecture Using Real-Time Scheduling Techniques

MEDIEN ZEGHID<sup>ID1,2</sup>, KAMAL AMASEB<sup>3</sup>, HASSAN YOUSIF AHMED<sup>ID1</sup>, AKHTAR NAWAZ KHAN<sup>ID4</sup>, ABDELLAH CHEHRI<sup>ID5</sup>, (Senior Member, IEEE), TEENA SHARMA<sup>ID5</sup>, AND KOTTAKKARAN SOOPPY NISAR<sup>ID6</sup>

<sup>1</sup>Electrical Engineering Department, College of Engineering at Wadi Aldwaseer, Prince Sattam Bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia

<sup>3</sup>Electronics and Micro-Electronics Laboratory (E. μ. E. L), Faculty of Sciences, University of Monastir, Monastir 5000, Tunisia

<sup>2</sup>Department of Electrical Engineering, Krar Universitas Sudan, Omdurman 11123, Sudan

<sup>4</sup>Department of Electrical Engineering, University of Engineering and Technology, Peshawar, Jalozai Campus, Peshawar 25120, Pakistan

<sup>5</sup>Department of Applied Sciences, University of Quebec in Chicoutimi (UQAC), Chicoutimi, QC G7H 2B1, Canada

<sup>6</sup>Department of Mathematics, College of Arts and Sciences, Wadi Aldawaser, Prince Sattam Bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia

Corresponding authors: Medien Zeghid (medien.zeghid@fsm.rnu.tn), Hassan Yousif Ahmed (hassanuofg@gmail.com), and Kottakkaran Soopy Nisar (n.soopy@psau.edu.sa)

This work was supported in part by the College of Engineering at Wadi Aldwaseer, Saudi Arabia, and in part by the Université du Québec à Chicoutimi under Grant UBR 326853.

**ABSTRACT** Contention is the main problem in optical burst switching (OBS) systems that lead to loss more burst by dropping the contended burst. To deal with this problem, we propose a new proactive method which is based on the integration of Real-Time Scheduling Techniques: Earliest-deadline-first (EDF) and round-robin (RR). Based on this consideration, this paper proposes a series of novel coherent interdependence methods to propose a novel OBS edge node Architecture: (i) An adaptive priority system to obtain QoS guarantees, (ii) Admission control and bandwidth distribution to maximize the link throughput. The proposed technique classifies the incoming data packets into two categories: granted service (GS) for high priority and best effort (BE) for low priority. The RR technique is used for scheduling different BE and GS packets when they have the same deadlines while the EDF is used to manage GS packets with different deadlines. In addition, the admission and bandwidth assignment policies vary in function of traffic characteristics. For GS traffic, only the bandwidth used is needed, while the remaining bandwidth is assigned to best effort traffic. With the EDF and RR scheduling algorithms, the proposed edge node design obtains a brand new architecture. The proposed OBS architecture has been tested for dynamic traffic in which both BE and GS traffic arrives according to a Poisson distribution for different scenarios based on traffic distribution and switch resources reservation. In scenario#1, 90% GS traffic and 10% BE traffic and reserve 7% of the switch resources for BE traffic are considered. Therefore, the proposed architecture provides a higher grade of service to the end-users with the guaranteed service level of agreements than the existing architecture that does not implement resource reservation protocols. In addition, the modified OBS-edge node is designed and implemented on FPGA Virtex-11C2V40. The hardware performance analysis shows that the network dimension is a key factor in latency measurement. A  $2 \times 2$  network dimension can serve approximately 230 Mbps throughput and approximately 400 Mbps in the case when the number of flits (segment of burst) is twice.

**INDEX TERMS** OBS, edge node, contention, QoS, BE, GS, EDF, RR, scheduling, Poisson distribution, FPGA.

## I. INTRODUCTION

Optical networks have gained significant attention since their inception due to the availability of massive bandwidth in

The associate editor coordinating the review of this manuscript and approving it for publication was Barbara Masini<sup>ID</sup>.

THz [1]. To ensure desired QoS performance, several optical techniques are utilized to provide optical switching. Optical switching is a fundamental functionality in next-generation optical networks to answer to the ever-growing bandwidth demand in modern communication networks. An optical switching network primarily employs three techniques such

as optical circuit switching (OCS), optical packet switching (OPS), and optical burst switching (OBS). OCS is a type of optical switching network that operates by establishing wavelength paths (lightpaths) between the pairs of nodes. In OPS networks, the packets are independently routed via switches through the network in the optical domain [2].

OBS technique has received considerable attention from the research community as a promising solution for optical networks [3]. In an OBS network, packets are assembled into a larger frame called data bursts (DB). A burst header packet (BHP) is created for each burst. Each DB/BHP pair is routed to their destination to the corresponding egress OBS edge node through OBS core nodes [4]. OBS does not require end-to-end light path set up, and therefore it offers more efficient bandwidth utilization when compared to an OCS system. In comparison with OPS network, very less processing operations per packet are required in an OBS network core optical router. Consequently, the energy consumption and the carbon footprint of a core optical router in an OPS network are likely to be larger than that of an OBS network router for the same amount of data [5].

The OBS network architecture is mainly based on three parts referred to as 1) ingress node 2) core node and 3) egress node as shown in Figure 1. Ingress node aggregates and assembles the packets into DB that travels optically through the core node unit.

A burst disassembly unit in the egress unit recovers IP packets from the incoming data burst frame. It initially decomposes each burst packet and then disassembles the burst packet by extracting and buffering individual IP packets [5]. The function of the OBS network is mainly divided into sub-functions such as signaling, scheduling, and contention resolution [6]. The signaling function specifies the protocol that handles the transmission and how efficiently networks can be used. The scheduling function is responsible for the assignment of the desired wavelength by a given burst in case of the higher number of available wavelengths. Contention in the OBS system occurs if multiple bursts at the same time from different input ports are destined for the same output port; the contention in traditional electronic packet switching networks is handled through buffering.

Due to the use of a few buffers in OBS architecture, some of the bursts are dropped in the core network. Dropping one burst containing hundreds of packets and retransmission of data from upper layers overloads the network. Therefore, the contention resolution is considered one of the most challenging processes in OBS networks [7]. Contentions are the perennial weakness of OBS networks that results in burst losses.

The main objective of this paper is to present an efficient contention reduction through QoS support and Real-Time Scheduling Techniques (RTS). The QoS-RTS technique is implemented at the OBS network edge in an attempt to meet the delay and loss constraints of each IP packet class. We developed a half mesh OBS network, as shown in Figure 2, where each router is connected via its OBS

edge node. The data communications between the users of different OBS edge nodes are burst and transferred through the links and routers.

In this paper, we assume that:

- The QoS requirements of an IP packet are defined by the packet's class, whereas bursts are differentiated in the core based on assigned priorities. The incoming data packets are classified into two types: granted service (GS) for high priority and best effort (BE) for low priority.
- For any user, there are two virtual channels (VCs) to separate the BE from GS. This concept reduces the number of links and avoids the phenomenon of inter-blocking data between two directions of transmission and reception of data.
- The upstream devices accurately define service characteristics of data flows so that service flows mapped to BE-VC do not exceed a given percentage of the link capacity. By doing this, BE-VC is not full and the edge-scheduler can process packets in queues with lower priorities.

To this consideration, this paper proposes a series of novel coherent interdependence methods to meet the above challenges:

- We propose a new proactive technique to avoid OBS contention.
- An adaptive priority system to obtain QoS guarantees.
- Admission control and bandwidth distribution to maximize the link throughput. For GS traffic, only the bandwidth used is needed. The remaining bandwidth is assigned to best-effort traffic.
- With EDF and RR scheduling algorithms, the proposed edge node design obtains a brand new architecture. After detailed comparison, the proposed edge-node design has superior performance, that is, the proposed design reduces dropping probability than conventional design.

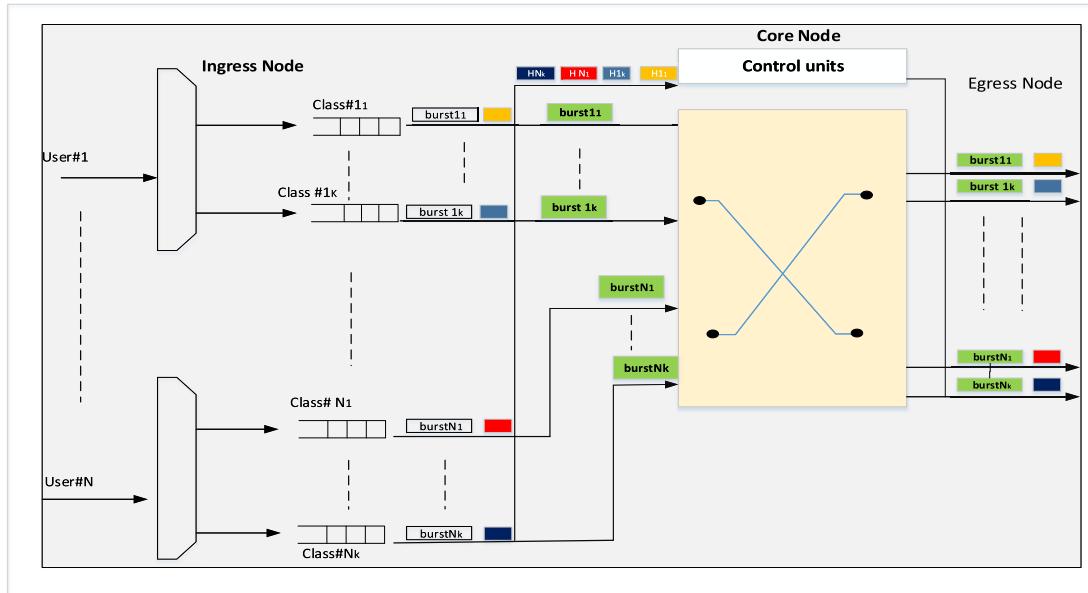
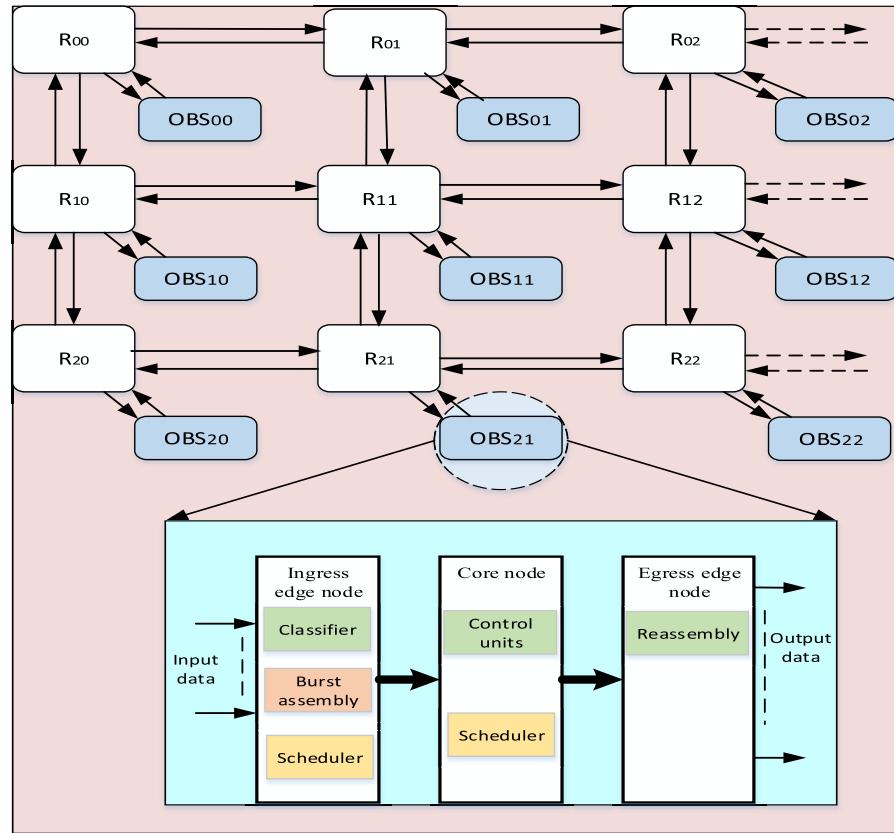
The remainder of the paper is organized as follows. We start by briefly describing the various techniques used for contention resolution in an optical burst switching network in section II. Section III discusses conventional OBS-edge node architecture. In Section IV, the modified OBS edge node design and its implementation is presented. Section V describes the Modified edge node scheduling techniques. A Poisson distribution analysis is performed in Section VI. Section VII presents hardware performance analysis. Finally, the paper is concluded in Section VIII.

## II. CONTENTION RESOLUTION IN OBS

The contention resolution is one of the most challenging tasks of OBS networks. Several contention avoidance and contention resolution mechanisms have been proposed in the literature.

### A. CONTENTION RESOLUTION

Contention resolution schemes are reactive approaches to resolve the contention. The reactive approach resolves the

**FIGURE 1.** OBS network architecture.**FIGURE 2.** OBS with mesh network.

contention when it occurs. It can be categorized into time-domain (e.g., fiber delay line (FDLs)), space domain (e.g., deflected routing), and optical domain (e.g. wavelength convertor).

FDLs are proposed as a contention resolution mechanism to delay the contending bursts for fixed duration of time [8], [9]. FDLs delay the signal according to the propagation time of the signal in it. The delay achieved is proportional to

the length of FDL. In [9], DUBEY *et al.* proposed estimation of data burst in advance which enables storage of the data burst. In this process, buffering of the data burst reduces average latency as well as helps to improve the burst loss probability (BLP). However, the length of FDLs depends on the burst length, which is variable in the case of OBS, and thus the use of buffers is limited in the case of OBS. For example, a 200-km standard FDL is needed to delay a single burst for 1 ms [4].

In the optical domain, wavelength converters are used to convert the wavelength of a contending burst to another available wavelength. A lot of research is dedicated to the sharing of wavelength converters among output channels and output links [10]–[12]. In [12], authors proposed a greater number of wavelength converters and network wavelengths for contention resolution when two or more bursts contend for the same wavelength. Elevated loss at constant network design is observed due to the increase in traffic or burst arrivals. Furthermore, wavelength conversion is an expensive optical technology.

In the space domain, the contention is resolved by using deflection routing [13], in which the contending bursts are deflected to less loaded output links. For the application of deflection routing, every core node calculates the alternate path for every possible destination. In [14], the contention is resolved by deflecting the contending bursts on an alternate path. This scheme does not require any extra hardware, decreases the burst loss probability, and increases the link utilization. However, the main drawback here is over congestion and recirculation of data bursts through the network. Marc De Leenheer *et al.* [15] presented a well-known contention resolution technique referred to as deflection routing in the context of any-cast-based OBS grid network. The authors suggested that a contending job burst can be redirected to the most appropriate resources in the network. Deflection routing has the problems of the potential network loops, out-of-sequence delivery of packets, and increased end-to-end transmission delay, special algorithms are required to deal with these issues.

Burst segmentation [16, 17] is another method of contention resolution in the space domain. In burst segmentation, only the overlapping segments of the bursts are dropped instead of dropping the whole burst. The burst, at the moment of assembly, is divided into segments which are the partitioning points of the burst. Additional control information is attached with each segment. Despite its contribution to the reduction of burst loss rate, burst segmentation adds overhead to the complexity of an OBS network.

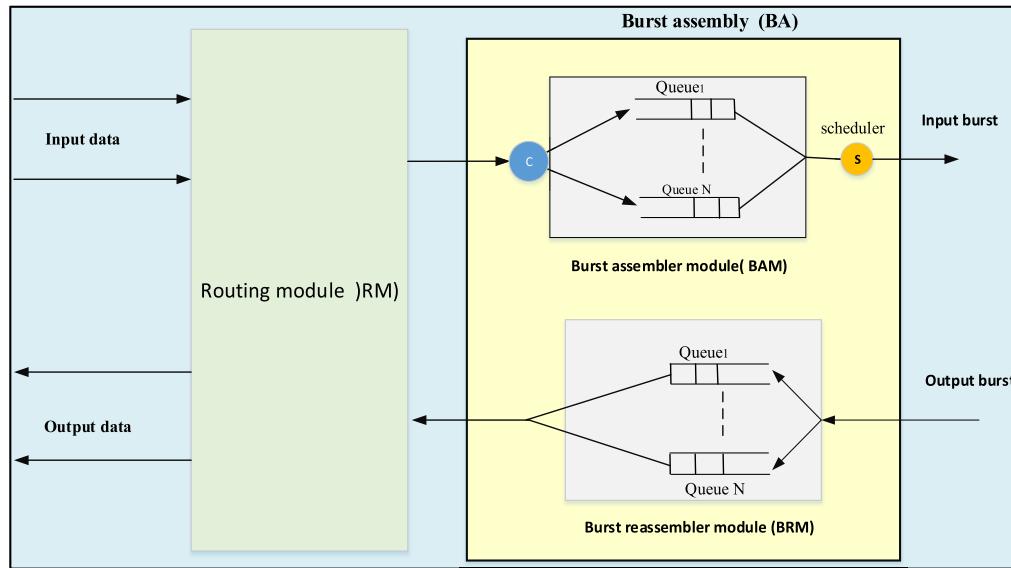
In [18], the authors reduced the packet loss by combining burst segmentation and fiber delay lines (FDLs) by resolving contentions during channel scheduling. They presented two types of scheduling algorithms which are classified based on the placement of the FDL buffers in the optical burst-switched node. These algorithms are referred to as delay-first or segment-first algorithms.

Further, in the ingress node when the packet is aggregated into DB, it is important to decide the appropriate DB length for specific network parameters to minimize the data loss probability in the OBS networks. When DB becomes too long, it will reduce the total number of bursts injected into the OBS networks. However, in the case of contention, the average number of lost packets per contention will increase. Hence, smaller bursts will increase the number of DBs in the OBS networks, leading to higher contentions with increased packet loss probability. Consequently, it is more preferable to maintain the data from loss in comparison with the data delay per contention.

### B. CONTENTION AVOIDANCE

Contention avoidance is a proactive approach that tries to minimize the contention, it avoids the contention before its occurrence. Adaptive routing is one of the proactive approaches to resolve the contention in the space domain. It balances the load on the network, better utilizes the network resources and avoids contention before its occurrence [19], [20]. A composite data-burst method in which the packets having high-priority are placed at the tail-end is presented by Vinod M. Vokkarane, *et al.* [21]. Analysis shows that the proposed scheme guarantees maximum class isolation between traffic classes with low and high priorities. In addition, the presented technique provides better QoS support in OBS by usages of prioritized contention resolution and composite burst assembly. In the prioritized contention resolution scheme, priorities are included as a field in the burst header packet (BHP). This priority field is used in the preferential segment and deflects bursts when resolving contentions is performed in the core. In [22], the data burst is assembled from several Internet protocol (IP) packets in such a manner that the number of IP packets is changed according to the traffic load. In addition, the burst is segmented because the segment located at the end of the contended burst has a greater chance of surviving contention. Another way to avoid contention is to implement a TCP-like congestion avoidance mechanism to regulate the burst transmission rate [23], [24]. In this approach, the ingress edge nodes receive TCP ACK packets from egress edge nodes, calculate the most congested links and reroute their traffic accordingly. A potential drawback of these schemes is that rerouting the data bursts to alternative paths can potentially cause link congestion elsewhere and thus result in possible network instability.

Sun *et al.* [25] presented the implementation design of an overlay mode optical burst-switched network testbed. A flexible architecture consisting of a “transceiver + forwarding” edge node segment is presented which supports optical burst forwarding as well as electronic burst assembly/disassembly. The proposed testbed design provides wavelength selection for locally generated bursts, class of service, and transparency to cut-through bursts. Moreover, the functional modules of the control plane and related key design issues are presented. Results show that the proposed scheduling mechanism can



**FIGURE 3.** OBS-Edge node conventional architecture.

efficiently combine two different contention resolutions in wavelength and space domains. Experimental results validate the effectiveness of this network testbed in reducing burst blocking probability. In [26], a novel priority-based composite assembly scheme for optical burst switching (OBS) networks is proposed. The low and high packet classes are aggregated into a single burst simultaneously, and the highest-priority packets are placed in the middle, while the low-priority packets are at the tail and head of the burst. In [27], it is shown that the arrival of very large bursts is a rare event, and for moderate burst length, the buffering of contending bursts can provide a very effective solution. However, in the case of the arrival of large bursts, deflection can be used.

### III. OBS-EDGE NODE CONVENTIONAL ARCHITECTURE

A conventional OBS edge node includes two units: routing module (RM) and burst assembler (BA) module as shown in Figure 3. Incoming packets are injected into RM at the OBS edge node. The incoming packets aggregate in frames called burst; this burst is scheduled and transmitted into the fiber channel via the output ports.

The RM sorts input packets into some classes based on the destination, class of service, or other criteria. It is assumed that the packets are uniformly distributed among several possible destinations for a span-constrained network.

In the BA module, bursts are created based on one of several assembly algorithms such as time of receiving packet, size of burst (number of packet in burst), or a hybrid of time and size together. After burst assembly, a scheduling algorithm is adopted to select the completed bursts in the burst queues which are further chosen for transmission. The completed bursts are sent to the output through the OBS edge node. Burst assembly in general is aggregated in a frame

called data-burst, which belongs to a single class and sends it to the same destination of the egress node. As per the data-burst scheduled transmission time, the generated data-burst is transmitted on the output port [16], [21].

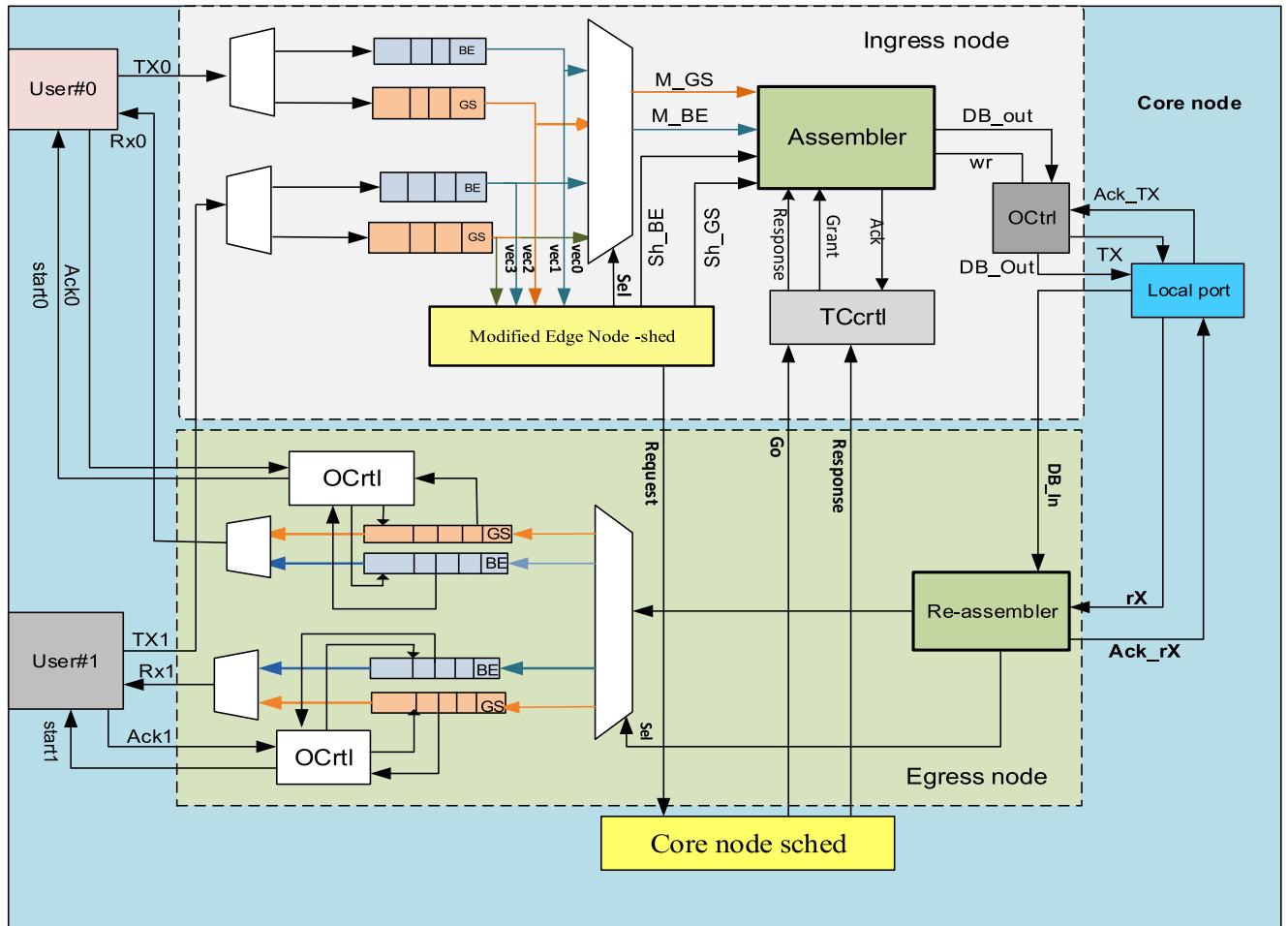
In conventional OBS edge node, the incoming packets are divided according to the arrival times and then forwarded to the OBS assembler unit directly without considering the priority of incoming, which leads to the possibility of losing the packets in the output port. In this work, we proposed a modified OBS edge node architecture that divided the incoming packets into high and low priorities which further depends on the QoS and type of output port. In the modified OBS edge node a new scheduler design is proposed, the new scheduler communicates with the main scheduler in the OBS system and uses two RTS scheduling algorithms to manage the transmission process.

### IV. MODIFIED OBS EDGE NODE: DESIGN AND IMPLEMENTATION

The front-end of the modified OBS edge architecture implements a point-to-point communication protocol as shown in Figure 4. This architecture consists of four major components to execute the transmission process: modified edge scheduler (MEN-sch), Assembler unit, time configuration controller (TC-ctrl), and output controller (O-Ctrl).

In our proposed design, the MEN-sch sends a request to the main scheduler based on the type of data burst (BE or GS). For effectiveness, prioritization is necessary at the edge node. To reduce the contention, the assembly module splits the formed burst into a fixed-length burst (packets), and each packet is then further split into flits. Each flit required to be sent at a precise time calculated by the TC-ctrl module.

During the absence of GS forwarding, a BE transmission is possible. For receiving process, there is a re-assembler



**FIGURE 4.** Modified OBS-edge node architecture.

unit connected with a de-multiplexer unit that inserts the data having high priority in virtual channels (GS-VC or BE-VC). The reassembly unit module is the main component of the back end. This module recomposes the incoming flits into a burst.

The features of the proposed modified edge node architecture are Real-time control of the system; Real-time scheduling of the services: GS and BE; Segmentation and reassembly.

#### A. QUEUES VIRTUAL CHANNELS: GS AND BE CHANNELS

The virtual channel (VC) mechanism is considered one of the most important concepts of QoS. In our proposed scheme, we have used VCs to separate the BE from GS, which reduces the number of links and further avoids the phenomenon of inter-blocking data between two directions of transmission and reception.

Each VC has its identifier as shown in Table 1 which illustrates the virtual channel number, traffic class, and rules applied for each type of traffic by tabulating the architecture of traffic on virtual channels.

**TABLE 1.** Fields of an incoming packet.

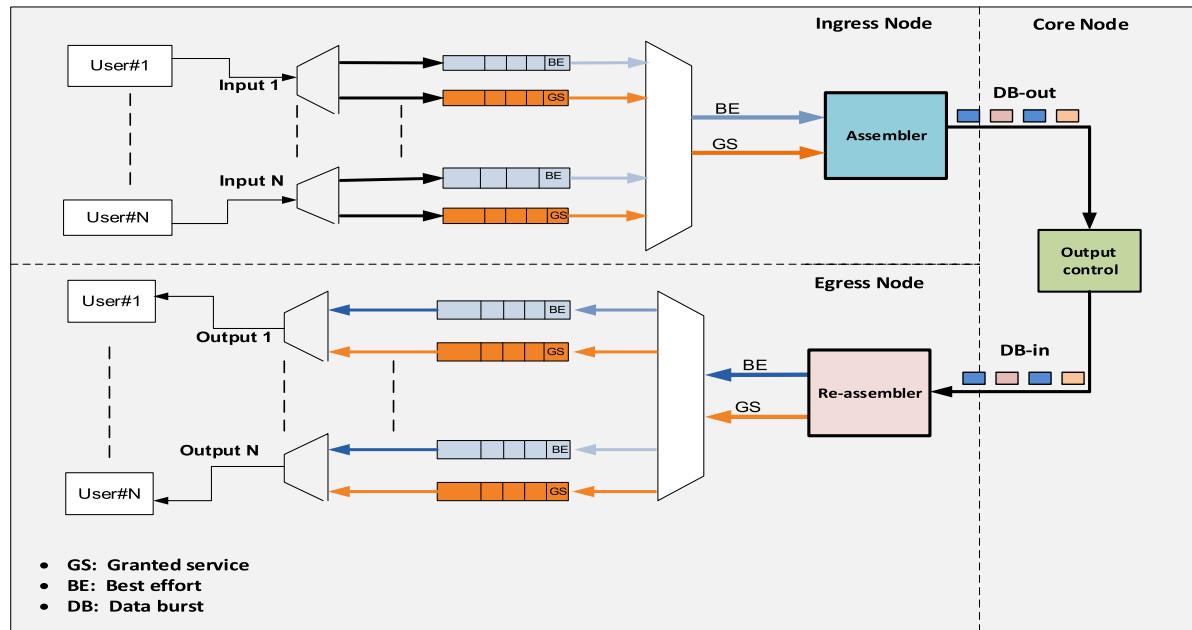
Traffic	VC Number	Class	Reservation
Best effort	0	BE	No reservation
Grantee service	1	GS	Reservation of time units along a path

In our proposed design, two VCs families are used as shown in Figure 5. VCs sources and destinations are used to store incoming input packets and then the DeMux directs these input packets to the corresponding channel (BE-VC or GS-VC). The VCs destinations stores the burst received by the “re-assembler” module. Each stored packet in BE-VC or GS-VC destinations is sent to the corresponding user.

The header of the incoming packets contains five fields: Data; st; @src; @dest; @Deadline; T. Table 2 describes the role of each field in the packet.

#### B. MODIFIED EDGE NODE SCHEDULER (MEN-SCH) MODULE

The MEN-sch prioritizes traffic classes and allocates the physical channel of the OBS to the router for the BE traffics

**FIGURE 5.** Modified OBS-edge node GS /BE VC channels.**TABLE 2.** Traffic type vs. virtual channels (VC).

Field	Description
Data	The data Load
St	Initial start time of a packet
@Src	Address of the OBS /user source
@Dest	Address of the OBS/user destination
Deadline	Deadline for transmission of a packet
T	Type of traffics: 1 for a GS and 0 for BE packets

only after processing all the GS traffics. Therefore, once a source VCs receives data from a user, it sends a data vector to the MEN-sch module, and each vector contains traffic type, deadline when the data is GS, source address, and destination address fields. In the case of two users per edge node, table 3 shows the VC identifier and data vectors for each type of traffic.

**TABLE 3.** Data vectors associated with the VCs sources.

User	VC	Data Vector (Di)
User#1	00 (BE)	Vect#0
	01 (GS)	Vect#1
User#2	10 (BE)	Vect#2
	11 (GS)	Vect#3

Once the MEN-sch receives one or more data vectors, it generates:

- A “read” signal: is used to enable the corresponding VC source.
- A global request directed to the core node-scheduler in the case of the received data vector contains a GS traffic type (i.e. vect#1 or vect#3 in table 3). This request

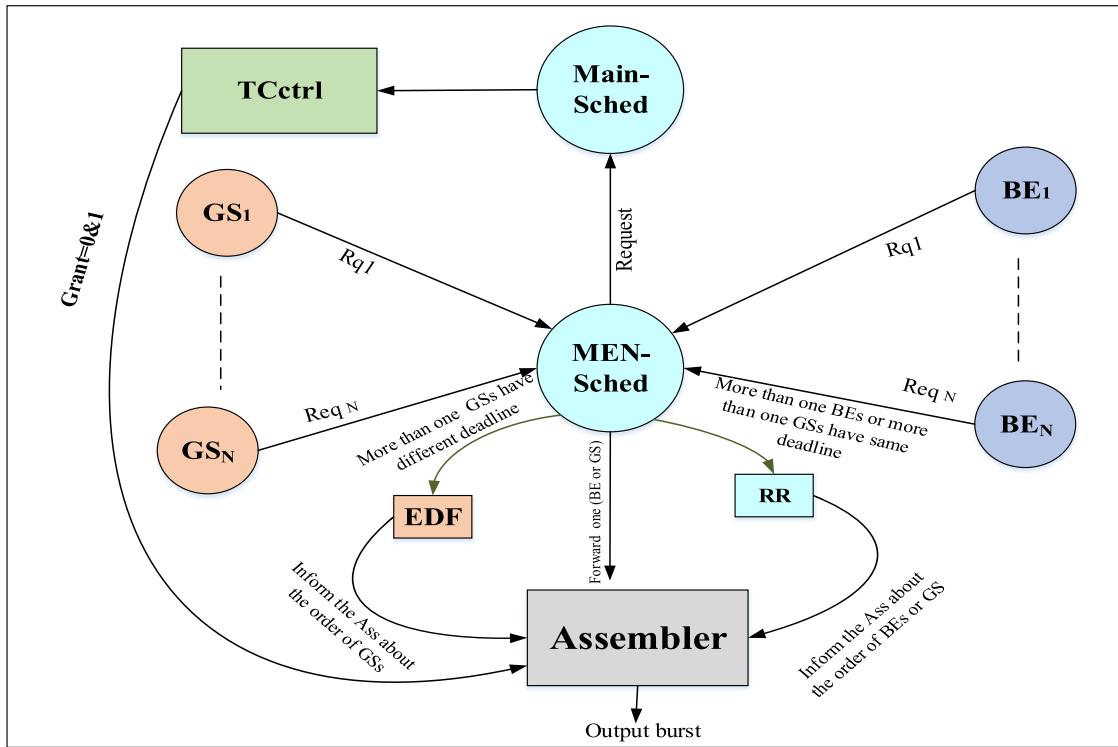
includes the address of the destination router, the start time of the first GS flit transmission, and the deadline of the GS DB. After scheduling calculation, the MEN-sch sends selection mechanism data “SMD” vector to the Mux N/1 (N is the number of users).

In this work, the purpose of the edge-node scheduler is to prioritize the transmission of packets from different users and different traffic classes with QoS support. It's similar to the traffic manager block used by network processors that typically implement per flow (or per group of flows) packet queuing and scheduling. It maximizes link throughput, at the same time it guarantees real-time requirements and avoids starvation of BE traffic. In the proposed design, we take into account the following strategies to achieve these goals:

- Admission control and bandwidth distribution to maximize the link throughput,
- An adaptive priority system to obtain QoS guarantees,
- An inhibition mechanism on QoS traffic, which avoids starvation of best-effort traffic and controls link bandwidth usage.

To parameterize bandwidth required by all sorts of traffic (BE, GS), time-space is divided in multiples of a flit frame that we denote as “slots (Tb)”.

Admission and bandwidth assignment policies vary in function of traffic characteristics. For GS traffic, only the bandwidth used is needed, which can be expressed as the maximum number of flits that can be sent in one round. Finally, the remaining bandwidth is assigned to best-effort traffic. Therefore, for GS traffic, the deadline to send a flit ( $T_{Delay}$ ) is determined as a function of its transmitting pace ( $BW_{link}/BW_i$ , where  $BW_{link}$  is the link bandwidth and  $BW_i$  is the bandwidth reserved for the  $VC_i$ ) in one round and



**FIGURE 6.** State diagram of the MEN-sch unit.

its accumulated delay. In the proposed design, the deadline for sending GS flits is defined by the main-sched after a request received from the MEN-sch indicating the presence of GS traffic. Among all QoS traffic ready to be sent – those with remaining bandwidth in the current round ( $BW_R > 0$ ), buffer space in the receiver VC, and satisfying its bandwidth constraints ( $T\_Delay \leq 0$ ) – the highest priority traffic is selected to be sent.

When there is more than one traffic with the highest priority, the EDF scheduling technique to manage the transmission process is used.

In the Modified edge node design, the BE-VCs and GS-VCs receive data from the user and send a data vector to the MEN-sch module. Each vector contains traffic type (BE or GS), deadline when the data is GS, source, and destination address fields. Hence the number of users per edge defines the number of elements in the data vector ( $\text{size}(\text{data vector}) = (\text{nbr of users})_2$ ). Furthermore in the data vector coding technique, always the  $LSB$  indicates the type of traffic whether BE ( $LSB = 0$ ) or GS ( $LSB = 1$ ) and the  $j^{th}$  bit in the data vector equals 1 where  $j = \text{index of the user}$ . To choose between queues, the Modified edge node scheduler needs to know each queue's treatment. When packets of different types arrive, the MEN-sch sends a request to the main-sched to calculate the allocated times to the GS packets. During the allotted times, the Time configuration Controller unit authorizes the transmission of GS packets while the transmission of BE packets is executed during the unallocated times.

The MEN-sch also stores the internal state like the last time a packet was transmitted from that queue and the current packet handle if any from that queue. Based on the received data vectors, MEN-sch performs a selection mechanism to choose between VC queues. The order in which traffics ( $Tr_i$ ) are served follows the cases below.

*Case 1:* when one (BE or GS) packet is received. In this case, the MEN-sch forwarded it directly to the assembler unit and  $SMD_i = '0' \& DV_i^{BE}$ .

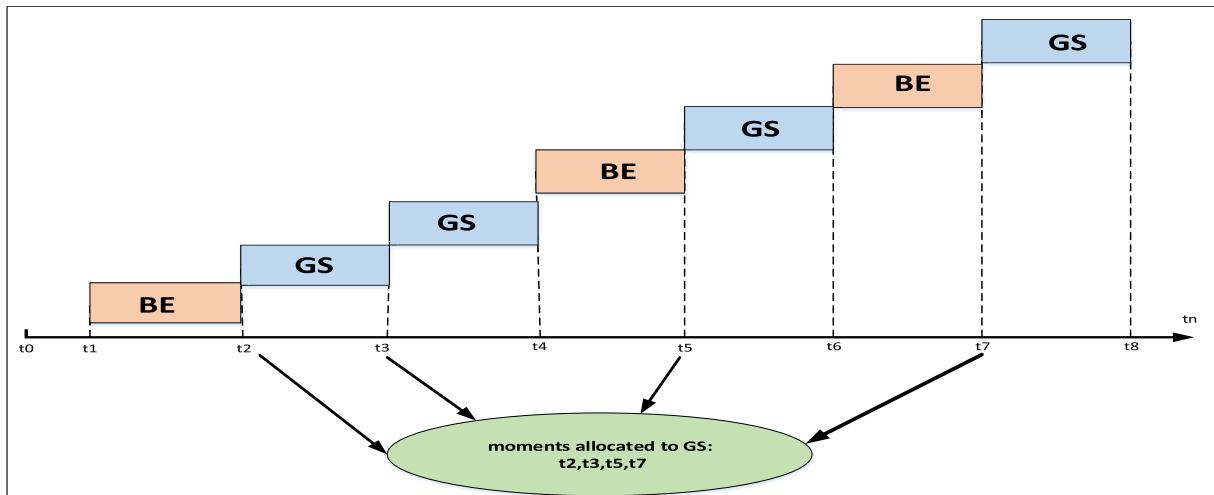
*Case 2:* multiple BE packets are received. In this case, the MEN-sch executes the RR scheduling technique to define the order of  $\{BE_i\}$  transmission. Hence  $SMD_i = '0' \& RRDV_i^{BE}$ .

*Case 3:* multiple GS packets with the same deadlines are received. The MEN-sch performs the RR scheduling technique to manage the transmission process.  $SMD_i = '0' \& RRDV_i^{GS}$ .

*Case 4:* multiple GS packets with different deadlines are received. The MEN-sch performs the EDF technique to manage the transmission process.  $SMD_i = '0' \& DFDV_i^{GS}$ .

*Case 5:* multiple GS and BE packets are received. The MEN-sch performs in parallel the EDF and RR techniques for the different GS and BE packets respectively to fix the transmission process.  $SMD_i = '0' \& (EDF \{DV_i^{GS}\} Or RRDV_i^{BE})$ .

Figure 6 shows the state diagram of MEN-sch in the MEN-OBS system. Figure 7 shows an example of interleaving between GS and BE flits as a function of transfer time to the router connected to the MEN. If there are more than two flits with different priorities (GS, BE) transmitted to the assembler



**FIGURE 7.** GS and BE flits transmission mechanism example.

unit, the scheduler allows to send the GS flits in the allocated time. This allocation takes place according to their deadline (i.e. each GS flit transmitted only at one time allocated of the set  $\langle ac \rangle$  slots in the cycles allocated by the MEN-sch module).

#### C. TIME CONFIGURATION CONTROLLER

The Time Configuration controller or the “TC-ctrl” is an intermediate module between the global scheduler “core node scheduler” and the module “assembly”. It takes care of reading the core node scheduler output and generates control signals to the assembly unit.

When the core node scheduler receives the request signal mentioned in the previous section, a centralized scheduling mechanism and a breadth-first search (BFS) routing algorithm are performed. Indeed, from the signal data request, the core node Scheduler will compute three fields necessary to conflict less GS burst and end-to-end GS-burst delivery time will be assured. When the response is ready, the Scheduler enables the TC-ctrl to communicate with the assembly unit to start the bursting process. The assembly unit starts when the grant and response are activated. The core node scheduler response contains three fields described as follows:

- $R$  (route): is the routing path for GS flits forwarded between source and destination.
- $rst$  (real start time): is the real-time transmission of the first flit of a GS burst.
- $ac$  (allocated cycles): are the units of time dedicated to the delivery of all flits of a GS burst on the path  $r$ .

#### D. ASSEMBLER UNIT

The assembler unit is a large block of our architecture and its role is as follows:

- Perform the task of assembly, i.e. dividing burst (BE or GS) into flits.

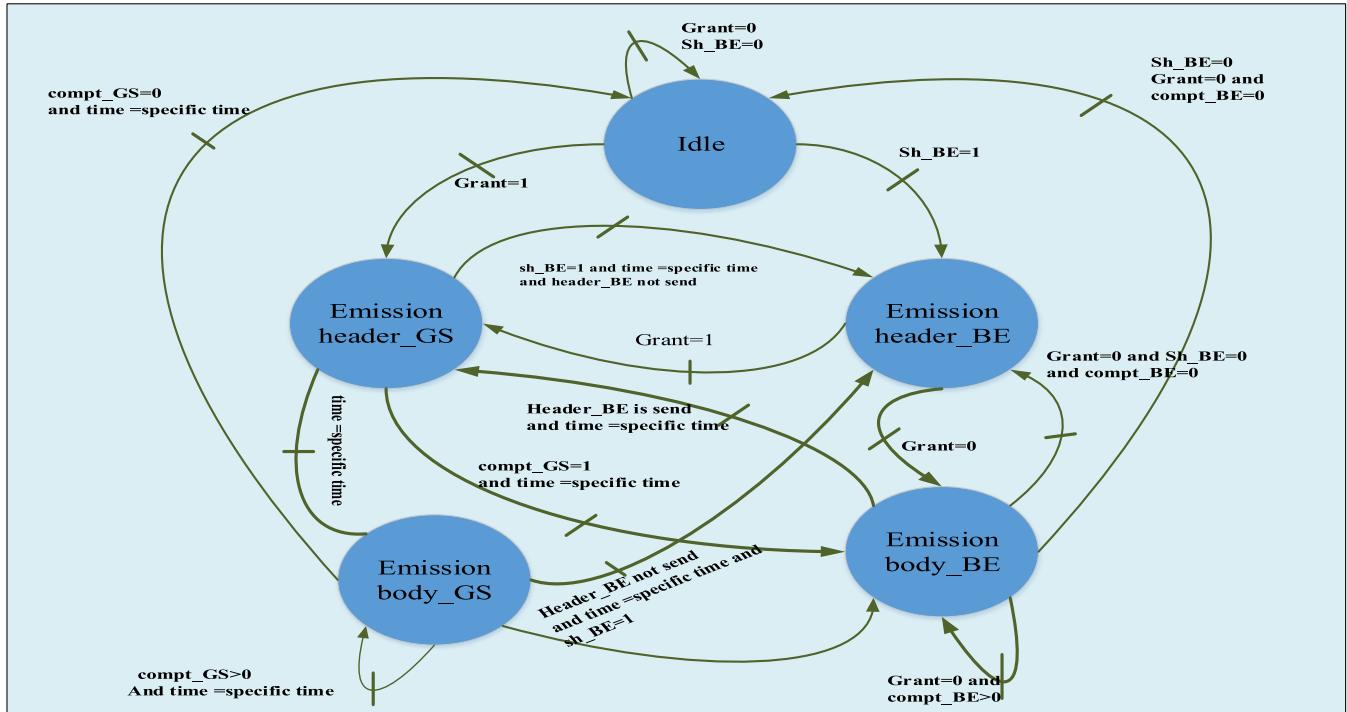
- Create header flits of GS and BE packets. An “nb” field will be added to the BE and GS headers. This field indicates the total number of flits composing the Burst. It is useful to keep track of the routing in every router in the network. The header flit of BE traffic consists of  $T$ ; @source; @dest; nb fields. For a GS, the assembly module receives the response signal sent by the configuration TC-ctrl unit. Once the response signal is received, the assembly creates the header flit structure. The GS header is slightly different from to BE header. Two new fields are involved “ $R$ ” and “ $rst$ ” instead of source and destination addresses.

- After assembly and creation of burst headers, the assembly module sends the flits corresponding to each burst (BE or GS) via the O-Ctrl.

The mechanism for sending GS flits is performed in real-time. Each flit must be sent at a specific moment. These moments are determined by the “ $ac$ ” field included in the response received by the TC-ctrl unit. In the absence of a GS flit, we can transmit a flit of BE type as mentioned previously. The state machine shown in figure 8 illustrates the inner work of the assembly module.

When the configuration entity TC-ctrl sends a signal ( $Grant = 1$ ) to the assembly, feedback for the forwarding request is ready. Otherwise, if ( $Grant = 0$ ) the assembler checkout the signal “ $sh\_BE$ ” coming from the edge node scheduler. This signal indicates if a BE burst is waiting.

When the “ $sh\_BE$ ” is set, the Assembly module switched to state  $\ll Emission\ Header\_BE \gg$  to send the first flit of a BE. As long as the assembler unit doesn’t receive any signal, it keeps the normal behavior by switching its state to  $\ll Emission\ Body\_BE \gg$  and keeps forwarding the flits to the router. Once all BE flits are transmitted to the edge node the assembler sets the signal  $\ll End\_BE \gg$  to inform the MEN-sch about the success of forwarding the complete BE. If the assembler unit doesn’t receive any request ( $sh\_BE = 0$  and



**FIGURE 8.** State machine of the assembler module.

$Grant = 0$ ), then it goes back to the «Idle» state. If the assembler unit receives a signal ( $Grant = 1$ ) while it's treating a BE-flit. It interrupts immediately BE traffic transmission process and switches to the state «Emission Header\_GS». At this state, the assembler set the signal *End\_GS* to inform the MEN-shed module that it is processing a GS flit. If the current variable time is equal to the instant «rst», then the assembler sends the first flit of the GS. If the time variable is equal to one of the «ac» instants planned by the main-Scheduler, the assembler module switches to «Emission Body\_GS» to forward the GS flits.

Otherwise, the assembler returns to the point where it was interrupted. Hence, it must check the *header\_BE* is at a low level then the state is switched to «Emission Header\_BE», otherwise it switches to «Emission Body\_BE».

The OBS system sends bursts via its local port in a sequence of generic-sized flits to the modified edge node. The module reassembly unit receives these flits and reassembles them to create a burst of type GS or BE. The Re-assembler receives the first flit of the burst (header flit), after that, it starts receiving burst flits. The received flits are classified to the corresponding VCs according to their classes. If the corresponding VC is not full, the Re-assembler forwards the GS and BE to it. Otherwise, the Re-assembler unit indicates that it cannot receive new flits until it has at least one place in the corresponding VC to store the flits waiting.

#### E. OUTPUT CONTROLLER UNIT (O-CTRL)

This module is a bridge connecting the output port of the assembler module to the local OBS node's port. The output

controller performs the flow control based on the handshake synchronization protocol. The assembler set “wr” when flits are ready to move on the OBS core node. Then the O-Ctrl unit transfers these flits to the selected output port after reviewing the “ack\_tx” signal indicating the availability of at least one output port. Otherwise, the flits are stored. Hence, a deadlock type Head-of-Line is avoided.

### V. MODIFIED EDGE NODE (MEN) SCHEDULING TECHNIQUES

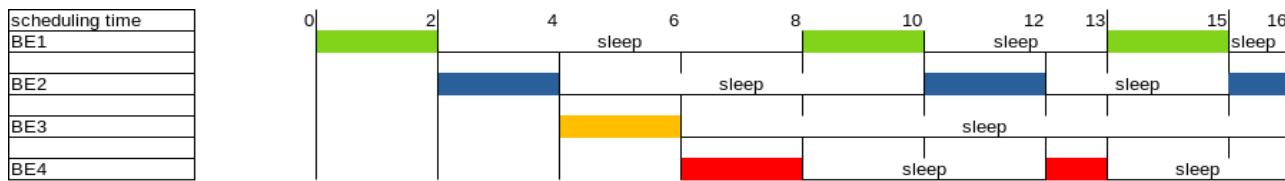
To avoid contention in the OBS system, the incoming data packets are classified into two types: granted services (GS) and best effort (BE). The characteristic of the BE and GS are listed as follows:

- BE class: data are sent via BE connection to arrive at the destination, with available bandwidth and maximum latency bound.
- GS class: data used time-division multiplexing control (TDM-Ctrl) to offer hard guarantees on maximum bandwidth and minimum latency.

In our proposed modified edge node design, two scheduling techniques are used to manage the packets. The two scheduling techniques are Round Robin (RR) and Earliest-deadline-first (EDF). RR technique is used for scheduling different BE and GS packets when they have the same deadlines while the EDF is used to manage GS packets with different deadlines.

#### A. ROUND ROBIN (RR) SCHEDULING TECHNIQUES

Round Robin (RR) Scheduling is a preemptive burst scheduling algorithm. For the RR algorithm, the time is divided into slots ( $T_b$ ) such that the length of each slot is equal to the

**FIGURE 9.** Round robin RR scheduling mechanism in modified edge node design.

transmission time of a flit. Therefore, time slots are assigned to each BE-packet in equal portions and circular order, handling all BE-packets without priority. In each time slot, the O-ctrl unit establishes up to N one-to-one connections from input ports to output ports. Each input port or output port can transfer or receive at most one flit in a one-time slot.

Every flit is assumed to arrive or depart at the beginning of a time slot. A round-robin scheduler employs time-sharing and gives each BE-packet a time and interrupts the current data if it is not completed by then. For example, if the time slot is  $2\text{ Tb}$  and the first data takes  $6\text{ Tb}$  to complete, then the RR scheduling algorithm will suspend the transmission of the active data after  $2\text{ Tb}$ , and give other data their time on the transmission. Once the other data have received their equal share ( $2\text{ Tb}$  each), data1 will get another allocation of the transmission process and the cycle will repeat. This process continues until the data is finished and needs no more time on the transmission process. Consider the following table with the arrival time and Deadline of four BE-DBs. The four BE present different arrival times and different deadlines.

**TABLE 4.** Round robin (RR) Scheduling algorithm for four different DBs.

BE <sup>i</sup>	Arrival time (Tb)	Deadline (Di)
BE <sup>1</sup>	0	6
BE <sup>2</sup>	1	5
BE <sup>3</sup>	2	2
BE <sup>4</sup>	5	3

Based on table 4, figure 9 describes the RR mechanism. From table 4 it can be stated that the modified edge node scheduling unit will suspend the transmission of the first BE after  $2\text{-Tb}$  and allow the second BE to start transmission for another  $2\text{-Tb}$  (it presents the nearest second arrival time = 1). Then BE<sup>3</sup> and BE<sup>4</sup> will get an allocation of transmission process after  $4\text{-Tb}$  and  $6\text{-Tb}$  respectively. Thus BE<sup>1</sup> will get the second and the third allocation of the transmission process after  $8\text{Tb}$  and  $13\text{Tb}$  respectively. As for BE<sup>4</sup> it needs only two allocations at  $6\text{Tb}$  and  $12\text{Tb}$  and terminates at  $13\text{Tb}$ .

### B. EDF SCHEDULING TECHNIQUE FOR GS CLASS

For GS class, worst-case delays with no packet loss can be guaranteed for the data packet. This type of QoS is performed by using earliest-deadline-first (EDF) scheduling policies on the edge node to avoid deadlocks and satisfy the need for communication requirements. EDF is a dynamic priority

scheduling algorithm for real-time embedded systems to place processes in a priority queue. Thus, the GS are sorted and therefore issued according to the value increasing time allowed. The earliest deadline first technique selects a task according to its deadline such that a task with the earliest deadline has higher priority than others. The priority of a task is inversely proportional to its absolute deadline. Since the absolute deadline of a task depends on the current instant of time therefore every GS flits is a scheduling event in EDF as a deadline of task changes with time.

For example, if there are three GS that have the same release time; different sizes; different deadlines, and different period times. Where,

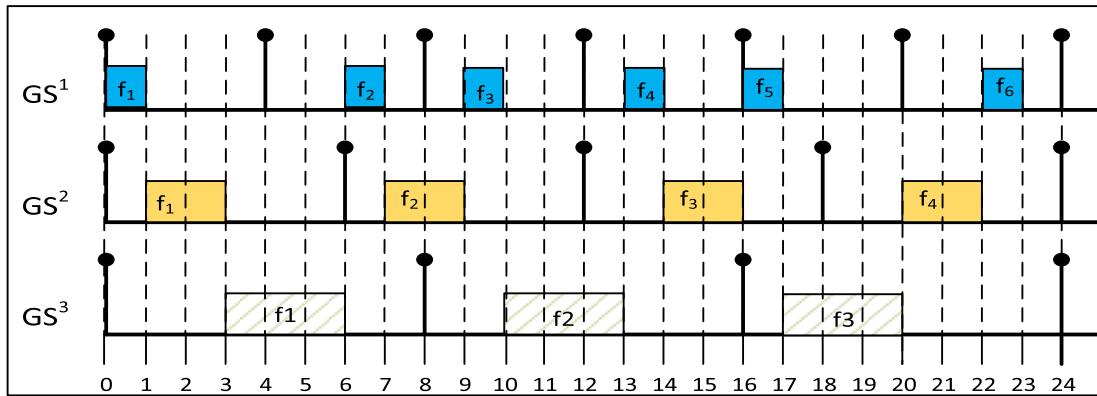
- Release time is the time of incoming GS burst.
- GS size is the number of flits per GS burst.
- Deadline is the end time of  $GS_{fj}^i$  in the  $GS^i$  where  $i$  is the number of GS,  $j$  is  $j^{\text{th}}$  flits in  $GS^i$ .
- Period time is the time of one flit in the deadline time of GS.

Figure 10 displays the earliest-deadline-first (EDF) scheduling execution process for  $GS^1$ ,  $GS^2$ , and  $GS^3$ .  $GS^1$  having number of flits = 6, period time = 1 and deadline = 4. Whereas  $GS^2$  has 4 flits, period time = 2, and deadline = 6. In the meantime,  $GS^3$  set with 3 flits, period time = 3 and deadline = 8.

At  $t = 0$ ,  $GS^1$ ,  $GS^2$ , and  $GS^3$  are released in which priorities are decided according to their absolute deadlines.  $GS^1$  has higher priority as its deadline is 4 earlier than ( $GS^2$ , deadline = 6) and ( $GS^3$ , deadline = 8), that's why it executes first and  $GS_{f1}^1$  is transmitted. Then at  $t = 1$  again absolute deadlines are compared and  $GS^2$  has a shorter deadline (the new deadlines are 8 for  $GS^1$ , 6 for  $GS^2$  and 8 for  $GS^3$ ) so it executes ( $GS_{f1}^2$  is transmitted) and after that  $GS^3$  starts execution at  $t = 3$  to transmit  $GS_{f1}^3$ . However at  $t = 4$ , which marks the end of the first deadline  $D_1$ ,  $GS^1$  is released and deadlines are compared again, the new deadlines are 8 for  $GS^1$ , 12 for  $GS^2$  and 8 for  $GS^3$ , at this instant both  $GS^1$  and  $GS^3$  has same deadlines so we continue to run  $GS^3$  as the transmission of  $GS_{f1}^3$  is in progress.

Thus  $GS^1$  is executed and ends its execution after one period time and  $D_1$  is updated to 8, at  $t = 1$   $GS^2$  starts to execute until  $t = 3$  (at  $t = 3$ ,  $D_2$  is updated to 12) then  $GS^3$  starts, because ' $t = 3$ ' is < to the minimum of deadlines ( $D_1$ ).

At  $t = 6$ ,  $D_3$  is updated to 16 and  $GS^2$  is released, now the deadline of  $GS^1$ (deadline = 8) is earliest than



**FIGURE 10.** Earliest deadline first scheduling technique of the GS burst.

$GS^2$  (deadline = 12) so it starts execution to transmit  $GS_{f_2}^1$ . Within 1t,  $D_1$  is updated to 12 and  $GS^2$  begins to execute at  $t = 7$ . Again at  $t = 8$ ,  $GS^1$  and  $GS^3$  are released but we continue to serve  $GS^2$  as the transmission of  $GS_{f_2}^2$  is in progress. Then at  $t = 9$ ,  $D_2$  is updated to 18 and the new deadlines become 12, 18, and 16 for  $D_1$ ,  $D_2$ , and  $D_3$  respectively. Therefore at  $t = 9$ , the transmission process of  $GS_{f_3}^1$  is executed for 1t and at  $t = 10$   $D_1$  is updated to 16, and  $GS_{f_2}^3$  is transmitted (end of transmission at  $t = 13$ ).

At  $t = 12$ ,  $GS^1$  and  $GS^2$  are released but we continue to run  $GS^3$  as the transmission of  $GS_{f_2}^3$  is in progress.

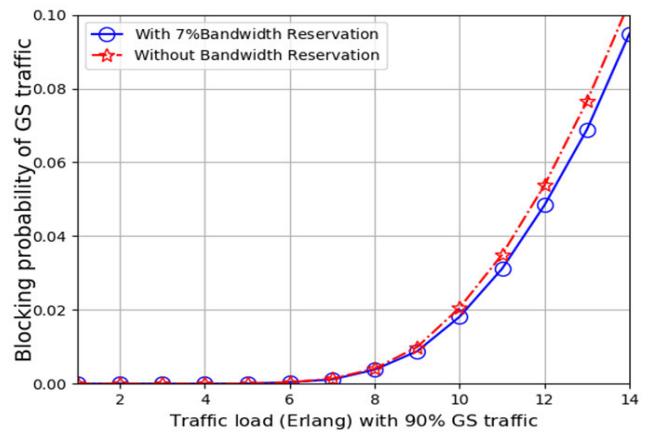
After 1t ( $t = 13$ ),  $GS^1$  starts its execution to transmit  $GS_{f_4}^1$ , and  $D_3$  is updated to 24. Within 1t,  $D_1$  is updated to 16 and  $GS^2$  begins to execute at  $t = 14$  for 2t. At  $t = 16$ ,  $D_2$  is updated to 24. Now at  $t = 16$ ,  $GS^1$  and  $GS^3$  are released together, priorities are decided according to absolute deadlines so  $GS^1$  executes first as its deadline is  $D_1 = 16$  and  $GS^3$ 's deadline is 24. After  $GS^1$  completion ( $D_1$  is updated to 20)  $GS^3$  starts and reaches at  $t = 18$  where  $GS^2$  is released in the system now by deadline comparison both have the same deadline 24 so ties are broken randomly and we continue to execute  $GS^3$ . At  $t = 20$  both  $GS^1$  and  $GS^2$  are in the system and both have the same deadline of 24 so again ties are broken randomly and  $GS^2$  executes. After that  $GS^1$  completes the execution. In the same way, the system continues to run without any problem by following the EDF algorithm.

## VI. POISSON DISTRIBUTION ANALYSIS

In this section, the proposed OBS architecture in Figure 5 has been tested for dynamic traffic in which both BE and GS traffic arrives according to a Poisson distribution and occupies switch resources for a finite duration of time and then releases the occupied resources. The service rate is considered to be exponentially distributed in simulations with the mean holding time equal to one. In all simulations, the fiber capacity has been divided into 16 wavelength channels. For all scenarios, in the case of the existing OBS architecture, both BE and GS occupy all available switch resources and mixed with each other. This makes the switch fragile for GS

traffic and contributes to high blocking of GS traffic as shown in Figures 11-14 for different scenarios.

In Figure 11, we consider 90% GS traffic and 10% BE traffic and reserve 7% of the switch resources for BE traffic. This reduces switch blocking for different load rates compared to the existing OBS architecture without bandwidth reservation. In Figure 12, we consider 75% GS traffic and 25% BE traffic and reserve 20% of the switch resources for BE traffic. In Figure 13, we consider 60% of GS traffic and 40% of BE traffic while reserving 25% of the switch resources for BE traffic. Finally, in Figure 14, we consider 50% GS traffic and 50% BE traffic and reserve 30% of the switch resources for BE traffic. In all cases, the proposed architecture is more flexible to the GS traffic and contributes to lower blocking compared to the existing architecture without resource reservations. Therefore, the proposed architecture provides a higher grade of service to the end-users with a guaranteed service level of agreements compared to the existing architectures which do not implement resource reservation protocols.



**FIGURE 11.** Traffic load (Erlang) with 90% GS traffic.

## VII. HARDWARE PERFORMANCE ANALYSIS

Our proposal employs a programmable logic device based on an FPGA to store and update connection states, and to

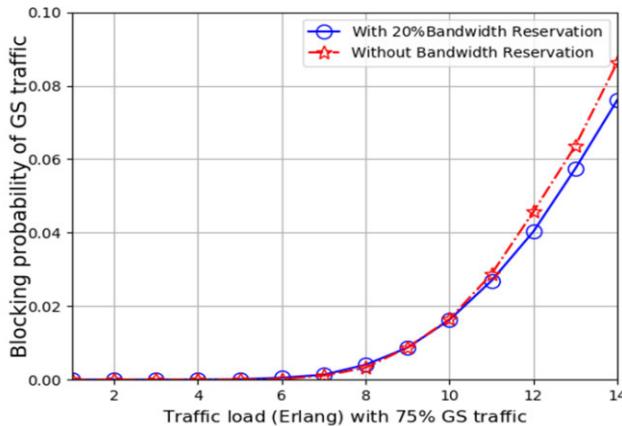


FIGURE 12. Traffic load (Erlang) with 75% GS traffic.

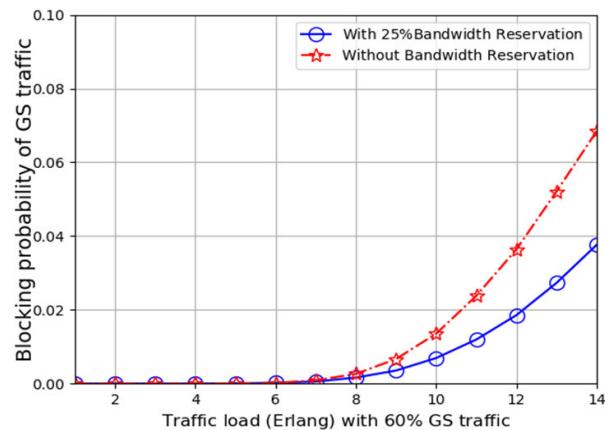


FIGURE 13. Traffic load (Erlang) with 60% GS traffic.

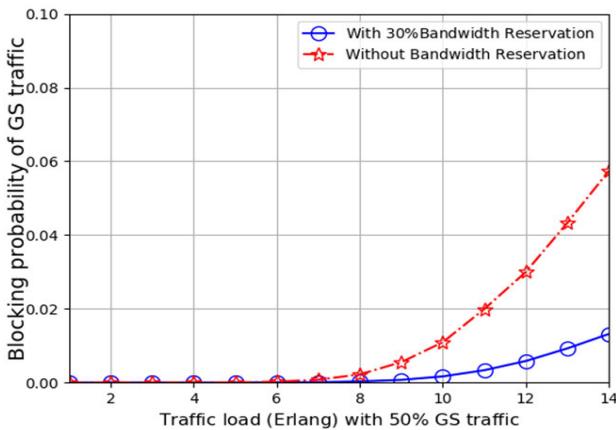


FIGURE 14. Traffic load (Erlang) with 60% GS traffic.

decide what data stream is to be sent next. In our work, the components implemented in the FPGA perform several tasks: stream scheduling and data input/output. The edge-node architecture is connection-oriented and reliable, based on credit flow control. The architecture scales from 4 to 32 streams using a Virtex-II XC2V40. It supports links with

speeds in the order of Gbps while maintaining the delay and jitter constraints for the QoS streams. The performance analysis section presents the performance evaluation of our edge-node architecture and its simulation results.

#### A. PERFORMANCE ANALYSIS: LATENCY

This section deals with the latency variations versus the number of received DB. Latency is the time frame between receiving the DB ( $t_g$ ) and sending it to the connected OBS node ( $t_a$ ). Therefore, the latency of the modified edge node design  $l_{MEN}$  is expressed as:

$$l_{MEN} = t_a - t_g \quad (1)$$

The total latency of DB is equal to the sum of the latencies of all modules composing the modified edge node architecture.

For this, we have two types of latencies: latency user-OBS node and latency OBS node-user. For the source user connected to the desired OBS node, the latency is written as follows:

$$L_{\text{user-OBS}} = L_{\text{srcq}} + L_{\text{MENsch}} + L_{\text{ASS}} + L_{\text{OutCtr}} \quad (2)$$

- In the reverse direction, the latency will be:

$$L_{\text{user-user}} = L_{\text{Dstq}} + L_{\text{ReASS}} + L_{\text{OutCtr}} \quad (3)$$

where,

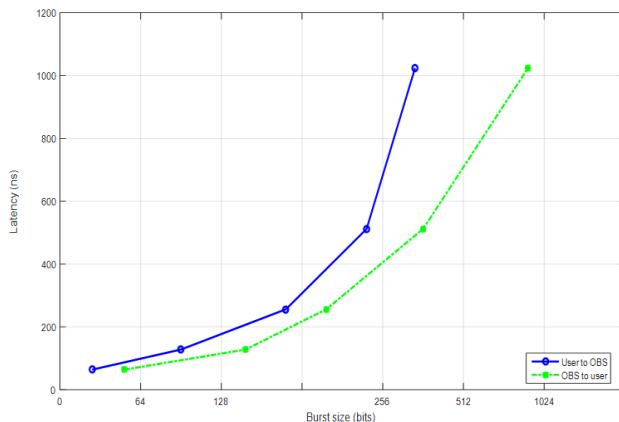
- $L_{\text{SrcQ}}$  is the latency consumed by the source VC queues.
- $L_{\text{MENsch}}$  is the latency consumed by the local scheduler.
- $L_{\text{Ass}}$  is the latency of the assembler unit.
- $L_{\text{outCtrl}}$  is the latency caused by the output controller.
- $L_{\text{ReAss}}$  is the latency of the re-assembler unit.
- $L_{\text{DstQ}}$  is the latency of the destination queues.

Figure 15 shows the variation of the modified edge node architecture latency relative to the size of a best-effort burst. For granted service burst-type, latency is deterministic regardless of BE burst load. The green-colored curve shows the (OBS -user) latency values calculated for 64, 128, 256, 512, and 1K bits. The blue-colored curve shows the latency (user-OBS) measured for the same values. Results indicate that the latency is proportional to burst size. This is more remarkable for operation reassembly (user-OBS) than (OBS -user).

For a given time slot, the latency for transferring a burst through any OBS node in the network is given by:  $L_{\text{OBS}} = L_w + L_s$ . During  $L_w$  – time, queues are waiting for incoming bursts via an input port of the OBS node. This time is equal to 6 Tb, while  $L_s$  is the latency for routing burst from an input port to an output port of the same OBS node. In our design, this time is always equal to 1 Tb for the BE burst and GS traffic. The required time to transfer BE from an input port to the output port within the same OBS node can be calculated as:

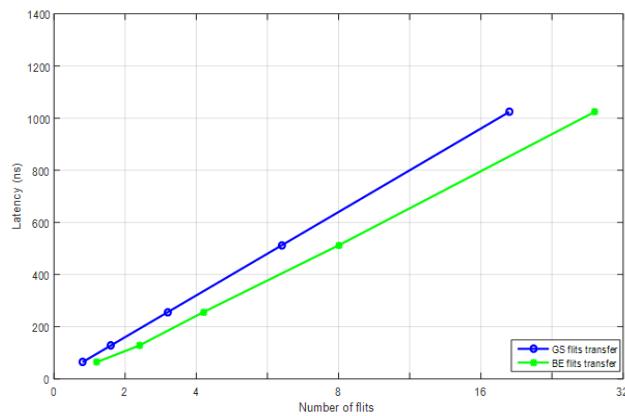
$$L_{\text{BE}} = L_w + (M \times 2\text{Tb}) \quad (4)$$

$M$  is the BE burst size measured in flits. Latency in Tb of a GS transfer from an input port to an output port for a single OBS node is given by:  $L_{\text{GS}} = M \times 2\text{Tb}$ .

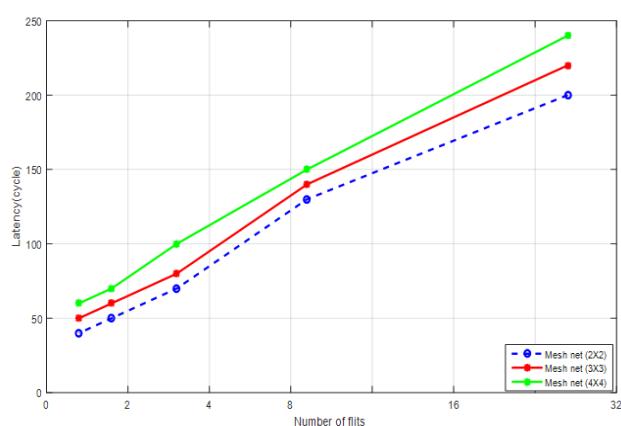


**FIGURE 15.** Variation of Latency by burst size.

Figure 16 shows the latency of arrival time ( $T_b$ ) for BE and GS flits from the input port to output port drawn vs. the number of flits. It is clear from the figure that when latency time increases, the number of flits increases as well for both BE and GS cases.



**FIGURE 16.** Variation of Latency by burst type (BE and GS).



**FIGURE 17.** Latency relative to the number of flits in a mesh net (2 x 2), (3 x 3), (4 x 4).

While the latency to transfer a flit (GS or BE) from an input port to an output port (assembly direction) is given by:

$$L_{\text{user-OBS}} = (\text{Nbr} - 1) + \text{Nbr} + 5T_b \quad (5)$$

where Nbr is the number of flits per incoming DB. In the opposite direction of assembly, the latency to transfer a flit (GS or BE) from an input port to an output port within a single modified edge node (direction disassembly) is given by:

$$L_{\text{OBSnode-user}} = T_b + 4T_b \times \text{Nbr} \quad (6)$$

Thus, the total latency when forwarding a DB from a source-user to a destination-user is written as follows:

$$L_T = L_{\text{user-OBS}} + L_{\text{net}} + L_{\text{OBS-user}} \quad (7)$$

where  $L_{\text{net}}$  is the routing latency of a DB. The minimum latency for the time slot to transfer a burst from a source OBS node to a target OBS node is given by:

$$L_{\text{net}} = \sum_{i=1}^n L_{\text{OBS}_i} = \sum_{i=1}^n L_{W_i} + (M \times 4T_b) \quad (8)$$

where

- $L_{\text{OBS}_i}$  is the number of router jumps (source and destinations OBS are included).
- $L_{W_i}$  is the time required for the routing/arbitration algorithms per OBS node.

Figure 17 shows the variation of latency plotted versus the number of flits in a mesh network (2 x 2), (3 x 3), and (4 x 4) relative to the number of flits per burst. From the figure, we can observe that the latency increases not only in terms of burst size but also in terms of network size (number of OBS jumps).

## B. PERFORMANCE ANALYSIS: THROUGHPUT

The total throughput is measured in bits per second or flits per  $T_b$ . This is the summation of traffic initiators. Query traffic is the number of flits entering the network per  $T_b$  ( $T = \Sigma T_b$ ) and per master. The total throughput is given by the below equation, where  $L$  is the number of flits per burst, and  $N$  is the number of bursts in the network.

$$\text{Th}_T = \sum_{i=\text{Nb\_maires}} N \times L \quad (9)$$

Initially, the speed of the modified edge node architecture is calculated in both directions: assembly and re-assembly. The throughput of assembly-direction is:

$$\text{Th}_{\text{user-OBS}} = \frac{N}{L_{\text{user-OBS}}} \times L \quad (10)$$

Figure 18 shows the throughput of the assembly-direction from: a- user to OBS system; b- OBS system to a user

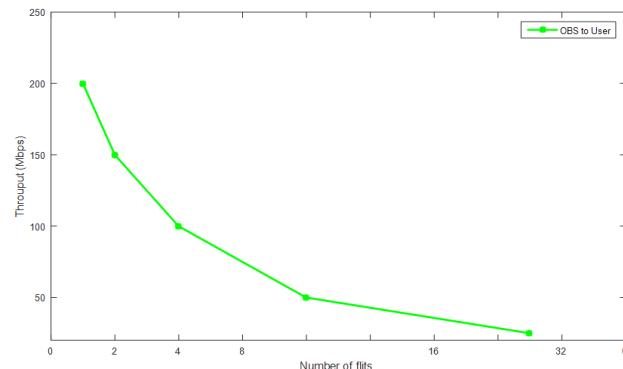
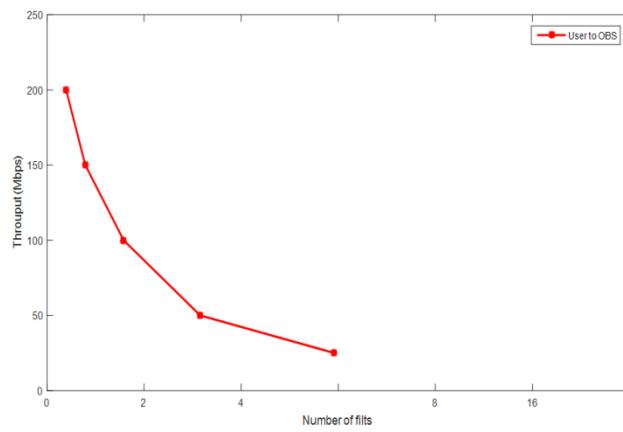
Finally, the throughput in the MEN is given by:

$$\text{Th}_T = \frac{N}{L_T} \times L \quad (11)$$

Figure 20 shows the variation of the throughput in a NET (2 x 2), (3 x 3), and (4 x 4) relative to the number of flits

**TABLE 5.** Round robin (RR) Scheduling algorithm for four different DBs.

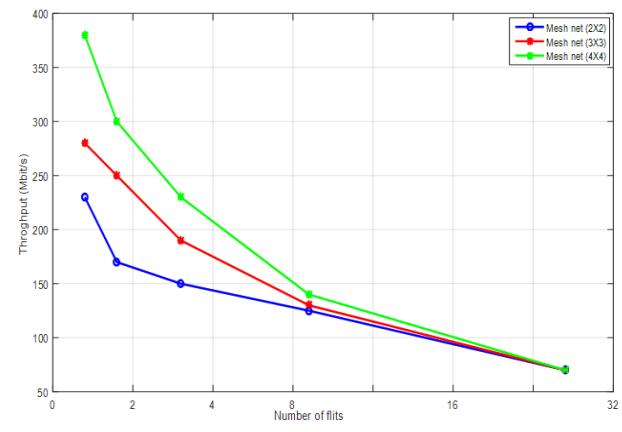
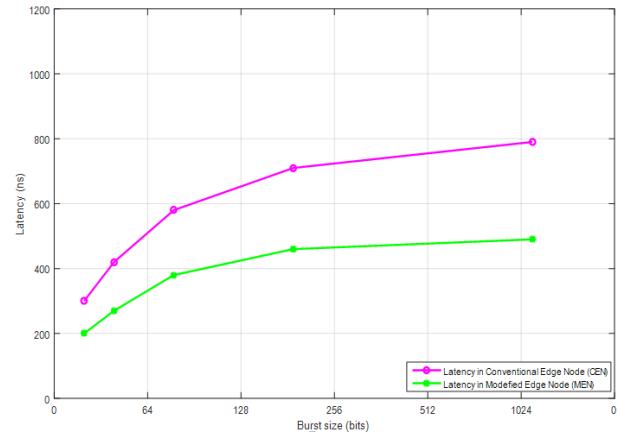
	Edge node process stages						
	Stage-1 (bursting assembly)		Stage-2 (burst transmission)				
	Aggregation of the data	Burst assembly	Route calculation	Channel scheduling	Offset time management	Burst control packet	Burst transmission
CEN	Multiple classes without priority system	data is sent without constraints	calculates the feasible path for the burst	OBS-core node scheduling	Related to the number of hops to the destination	One header for each burst	Data is sent out following its header packet with some offset time.
	Multiple classes with VCs and adaptive priority system	Admission control and bandwidth distribution	Pre-calculated feasible paths	EDF and RR scheduling algorithms	Time-space is divided into multiples of a flit frame: Tb	header for each flit of GS and BE.	“ac”: Units of time dedicated to the delivery of GS burst.

**FIGURE 18.** Throughput Vs Number of flits: user to OBS edge node; OBS edge node to user versus number of flits in a NET (2 × 2), (3 × 3), (4 × 4).

per burst. It is clear that the latency increases from topology (2 × 2) to (4 × 4). Moreover, the results show that there is a decrease in throughput when the number of flits increases since it is the inverse of latency multiplied by the number of flits per burst.

#### C. PERFORMANCE ANALYSIS: CONVENTIONAL ARCHITECTURE VERSUS MODIFIED EDGE NODE DESIGN

In the modified edge node a new scheduler module has been added which connected with the main scheduler in the OBS system and used two real-time scheduling techniques. The

**FIGURE 19.** Throughput versus number of flits in a NET (2 × 2), (3 × 3), (4 × 4).**FIGURE 20.** Burst size versus latency for CEN and MEN architectures.

additional components reduce the contention problem caused by the loss of a large number of data but at the price of latency. Therefore, a tradeoff must be made between hardware complexity and latency to reduce the contention through additional components eventually leading to classify the data into high priority-GS, and low priority-BE. Table 5 shows the differences between the conventional edge node (CEN) and

modified edge node (MEN) architectures in terms of burstification and transmission data. Figure 20 plots the burst size versus consumed latency in conventional edge node (CEN) and modified edge node (MEN) architectures. It is observed that as long as burst size increases latency increases as well. The latency of MEN is higher than CEN due to the deployment of additional components in MEN. Therefore, a tradeoff must be made between hardware complexity and latency to reduce the contention.

## VIII. CONCLUSION

In this paper, a proactive technique to avoid contention in OBS networks is proposed. The conventional architecture of the OBS edge node has been modified to incorporate the proposed proactive technique. Two real-time scheduling techniques are integrated and employed in the modified design. The proposed OBS architecture has been tested for dynamic traffic in which both BE and GS traffic arrives according to a Poisson distribution for different scenarios based on traffic distribution and switch resources reservation.

Four scenarios are examined; in scenario#1, a 90% GS traffic and 10% BE traffic and reserve 7% of the switch resources for BE traffic is considered whereas a 75% GS traffic and 25% BE traffic and reserve 20% of the switch resources for BE traffic for scenario#2. In the meantime, 60% and 50% of GS traffics and 40% and 50% of BE traffics and reserves of 25% and 30% of the switch resources for BE traffics were allocated for remaining scenarios respectively. In all scenarios, the proposed architecture is more flexible to the GS traffic and contributes to lower blocking compared to the existing architecture without resource reservations. Therefore, the proposed architecture provides a higher grade of service to the end-users with the guaranteed service level of agreements compared to the existing architectures which do not implement resource reservation protocols.

## REFERENCES

- [1] F. Espina, J. Armendariz, N. García, D. Morató, M. Izal, and E. Magaña, “OBS network model for OMNeT++: A performance evaluation,” in *Proc. 3rd Int. ICST Conf. Simulation Tools Techn.*, Malaga, Spain, 2012, pp. 1–8, doi: [10.4108/ICST.SIMUTOOLS2010.8708](https://doi.org/10.4108/ICST.SIMUTOOLS2010.8708).
- [2] Z. Rosberg, H. L. Vu, M. Zukerman, and J. White, “Performance analyses of optical burst-switching networks,” *IEEE J. Sel. Areas Commun.*, vol. 21, no. 7, pp. 1187–1197, Sep. 2003.
- [3] L. Shuo, “Analysis and synthesis of optical burst switched networks,” Ph.D. dissertation, Dept. Electron. Eng., Univ. Honkong, Hong Kong, Feb. 2014.
- [4] P. J. Jue and V. Vokkarane, *Optical Burst Switching Networks*. USA: Springer, 2005.
- [5] T. S. El-Bawab and J.-D. Shin, “Optical packet switching in core networks: Between vision and reality,” *IEEE Commun. Mag.*, vol. 40, no. 9, pp. 60–65, Sep. 2002.
- [6] A. Lazzez, “Optical burst-switched networks: Design, modeling and performances analysis,” Doctoral dissertation, Ph.D. dissertation, Univ. Carthage, Tunis, Tunisia, 2008.
- [7] T. Zhang, W. Dai, G. Wu, X. Li, J. Chen, and C. Qiao, “A dual price-based congestion control mechanism for optical burst switching networks,” *J. Lightw. Technol.*, vol. 32, no. 14, pp. 2492–2501, Jul. 1, 2014.
- [8] J. Pedro, P. Monteiro, and J. Pires, “Efficient optical burst-switched networks using only Fiber Delay Line buffers for contention resolution,” in *Proc. 4th Int. Conf. Broadband Commun., Netw. Syst.*, 2007, pp. 2–11.
- [9] S. Dubey, S. Debnath, and R. Srivastava, “Contention resolution in optical burst switches using fiber delay line buffers,” *J. Eng. Sci. Technol.*, vol. 1, no. 2, pp. 530–547, 2017.
- [10] V. Eramo, M. Listanti, and P. Pacifici, “A comparison study on the number of wavelength converters needed in synchronous and asynchronous all-optical switching architectures,” *J. Lightw. Technol.*, vol. 21, no. 2, pp. 340–355, Feb. 1, 2003.
- [11] M. F. Hayat, F. Z. Khan, and H. R. van As, “Performance model for an OBS node with a shared wavelength converter pool and an FDL buffer per link,” in *Proc. 15th Int. Conf. Opt. Netw. Design Modeling (ONDM)*, Feb. 2011, pp. 1–6.
- [12] M. S. Reza, M. M. Hossain, and S. P. Majumder, “Contention problem in optical burst switching network,” in *Proc. Int. Conf. Comput. Intell. Commun. Netw.*, Bhopal, India, Nov. 2010, pp. 239–242.
- [13] X. Wang, H. Morikawa, and T. Aoyama, “Deflection routing protocol for burst-switching WDM mesh networks,” *Proc. SPIE Terabit Opt. Networking: Archit., Control, Manage.*, vol. 4213, pp. 242–252, Oct. 2000.
- [14] S. Pallavi and M. Lakshmi, “A simple node architecture for optical burst switching,” *Int. J. Eng. Technol.*, vol. 6, no. 2, pp. 1072–1078, 2014.
- [15] M. De Leenheer, J. Buysse, C. Develder, B. Dhoedt, and P. Demeester, “Deflection routing in anycast-based OBS Grids,” in *Proc. 6th Int. Conf. Broadband Commun., Netw., Syst. (ICST)*, 2009, pp. 1–4.
- [16] V. M. Vokkarane, “Burst segmentation: An approach for reducing packet loss in optical burst switched networks,” in *Proc. Int. Conf. Commun.*, 2002, pp. 2673–2677.
- [17] S. Sarwar, S. Aleksic, and K. Aziz, “Optical burst switched (OBS) systems with segmentation-based dropping,” *Elektrotech. Infotech.*, vol. 125, pp. 296–300, Dec. 2008.
- [18] V. M. Vokkarane and J. P. Jue, “Segmentation-based nonpreemptive channel scheduling algorithms for optical burst-switched networks,” *J. Lightw. Technol.*, vol. 23, no. 10, pp. 3125–3137, Oct. 1, 2005.
- [19] V. M. Vokkarane, K. Haridoss, and J. P. Jue, “Threshold-based burst assembly policies for QoS support in optical burst-switched networks,” *J. Opt. Commun. Netw.*, vol. 4874, pp. 125–136, Mar. 2002.
- [20] S. Singh, “Priority based burst assembly for optical burst switched network,” *Indian J. Sci. Technol.*, vol. 9, p. 48, Dec. 2016.
- [21] 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, no. 7, pp. 1198–1209, Sep. 2003.
- [22] A. K. Garg, “Contention reduction and service differentiation in OBS networks,” *Optik*, vol. 123, no. 12, pp. 1108–1111, Jun. 2012.
- [23] X. Cao, J. Li, Y. Chen, and C. Qiao, “Assembling TCP/IP packets in optical burst switched networks,” in *Proc. GLOBECOM*, Taipei, Taiwan, Nov. 2002, pp. 2808–2812.
- [24] S. Y. Wang, “Using TCP congestion control to improve the performances of optical burst switched networks,” in *Proc. IEEE Int. Conf. Commun.*, Oct. 2003, pp. 1438–1442.
- [25] Y. Sun, T. Hashiguchi, V. Quang Minh, X. Wang, H. Morikawa, and T. Aoyama, “Design and implementation of an optical burst-switched network testbed,” *IEEE Commun. Mag.*, vol. 43, no. 11, pp. S48–S55, Nov. 2005.
- [26] A.-H. Guan, F. Hu, and W.-C. Li, “A new composite assembly mechanism for supporting QoS in OBS networks,” *Optoelectron. Lett.*, vol. 10, no. 1, pp. 55–58, Jan. 2014.
- [27] R. Awasthi, L. Singh, and A. U. Khan, “Estimation of bursts length and design of a fiber delay line based OBS router,” *Int. J. Sc. Tech.*, vol. 12, no. 3, pp. 834–846, 2017.



**MEDIEN ZEGHID** received the Ph.D. degree in information and communication, sciences and technologies from the University of South Brittany, Lorient, France, in 2011. He is currently an Assistant Professor with the Department of Computer Engineering and Networks, Prince Sattam Bin Abdulaziz University. His research interests include information security, architectural synthesis for the crypto-systems, image & video coding, and optical communication.



**KAMAL AMASEB** received the B.Eng. degree in computer engineering and the M.Sc. degree in computer engineering & network from Gezira University, Sudan, in 2000 and 2008, respectively. He is currently a Lecturer with the Electrical Engineering Department, College of Engineering, Sattam Bin Abdulaziz University, Saudi Arabia. His research interests include computer networks, wireless communications networks, and optical communications.



**ABDELLAH CHEHRI** (Senior Member, IEEE) is currently a Professor with the University of Quebec (UQAC), Canada. He worked as an Adjunct Assistant Professor with the School of Electrical Engineering and Computer Science, University of Ottawa, and as the Manager of Next-Generation Networks (NGN)/VoIP & Emerging Technologies at Bell Canada. He received many prestigious awards, including, Dean's Scholarship Award, Postdoctoral Studies (University Ottawa),

Scholarship Fund to Support Success (Laval University), Japan Society for the Promotion of Science, MITACS, NSERC Postdoctoral Fellowship. He served as the Program Chair and workshops chair of the 2014 IEEE International Humanitarian Technology Conference (IEEE IHTC 2014). He served as a guest editor and served as a technical reviewer for several international conferences and journals (IEEE, Elsevier, and Springer). Additionally, he is a member of the Order of Engineers of Quebec (OIQ) and IEEE Canadian Humanitarian Initiatives Committee (HIC).



**HASSAN YOUSIF AHMED** received the B.Eng. degree in computer engineering (network systems) and the M.Sc. degree in computer science and information from Gezira University, Sudan, in 2002 and 2007, respectively, and the Ph.D. degree in electrical and electronic engineering from Universiti Tecknologi PETRONAS, Malaysia, 2010. He is currently an Associate Professor with the Electrical Engineering Department, College of Engineering, Prince Sattam Bin Abdulaziz University. He is also the Dean's Assistant for Quality and Development. His research interests include computer networks, wireless communications networks, optical communications, and cryptography systems.



**TEENA SHARMA** received the B.Tech. degree in electronics and communication engineering from SBCET Jaipur, India, in 2008, and the M.Tech. degree from the Department of Electronics and Communication Engineering, Malviya National Institute of Technology Jaipur, India, in 2013. She is currently carrying out her research with the University of Quebec at Chicoutimi, QC, Canada, and a Research Assistant with Thales Communication, Canada. Her research interests include optical fiber technology, machine learning, the IoT, and wireless backhaul networks.



**AKHTAR NAWAZ KHAN** received the B.Sc. degree (Hons.) in electrical engineering from the University of Engineering and Technology (UET), Peshawar, Peshawar, Pakistan, in 2006, and the M.Eng. and D.Eng. degrees in telecommunications from the Asian Institute of Technology, Thailand, in 2009 and 2013, respectively. He is currently working as an Assistant Professor with the Department of Electrical Engineering, UET, Peshawar, Jalozi campus, and has been held this position, since 2015. Before Jalozi campus, he has worked as an Assistant Professor with the Department of Telecommunications, UET, Peshawar, Mardan campus, from May 2013 to April 2015. He has also worked as an International Internship Scholar with the Fukuda Laboratory, National Institute of Informatics, Tokyo, Japan, on the topic of large-scale simulation for internet topology analysis. His research interests include the design and analysis of optical WDM networks.



**KOTTAKKARAN SOOPPY NISAR** is currently a Full Professor in applied mathematics with Prince Sattam Bin Abdulaziz University, Saudi Arabia. His current research interests include special functions, fractional calculus, fluid dynamics, bio mathematics, machine learning, SAC-OCDMA, and multidisciplinary applications of mathematics. He is also an editorial board member of many reputed journals and has published more than 400 research articles in Scopus/WoS indexed journals. He is the winner of a number of awards, such as PSAU research excellent awards (Best researcher of the University), in 2019 and 2020, the Riemann Young Researcher Award 2020, and Indo Asian-Niels Henrik Distinguished Innovative Scientist Award 2020.

• • •