In: Third International Workshop on Optical Burst Switching (WOBS) Oct. 25, 2004 San José , CA, USA

A Testbed for Control Systems of Optical Burst Switching Core Nodes*

Sascha Junghans
University of Stuttgart, Institute of Communication Networks and Computer Engineering (IKR)
Pfaffenwaldring 47, 70569 Stuttgart, Germany
e-mail: junghans@ikr.uni-stuttgart.de

Abstract

On the way to bring more dynamic in optical transport networks, Optical Burst Switching (OBS) is a promising candidate for a new photonic network architecture. Beside the efforts of building fast optical cross connects, the realization of performant switch controllers is an important task. This paper presents a testbed which allows the development and realization of control systems for OBS core nodes and their performance evaluation. The testbed is working in the laboratory of the author's institute and first measurement results for a Horizon scheduling module are shown.

1. Introduction

With the large growing traffic volumes in packet based networks, the introduction of dynamic optical transport networks in contrast to static configured networks is necessary. Fast switching in the optical domain is still a sophisticated an expensive task. As optical processing is not possible, yet, switching decisions or even scheduling tasks must be performed in the electrical domain which leads to growing needs of processing power in the control part of network nodes. As optical transmission rates grow faster than electrical processing power, new mechanisms for the control part of network nodes are needed.

Optical Packet Switching (OPS) has a high complexity because of the need for very fast packet processing and short switching times. Optical Circuit Switching (OCS) with mainly static connections does not fit well to the highly dynamic traffic patterns of internet traffic. Optical Burst Switching (OBS) [1, 2] lies in between the two approaches with less complexity than OPS and a higher dynamic than OCS.

At the ingress nodes of the OBS network, data packets with the same destination node of the OBS network are assembled into larger data units, called bursts. These bursts are sent through the OBS network to the egress node without conversion into the electrical domain at the intermediate nodes. The task of fast wavelength identification and reservation in the intermediate nodes is called burst scheduling and has a big impact on the performance of the OBS network.

This paper introduces a framework which allows the realization of different scheduling modules and the evaluation of these modules by measurements. As an example scheduling mechanism a scheduling module for Horizon [3] was realized and measurements for it are shown.

The rest of the paper is organized as follows: Section 2 gives an introduction of OBS in general and of the Horizon scheduling scheme for resource reservations. In Section 3 the architecture of the control system of an OBS core node is shown and a functional description of the output port controller is given. Section 4 explains the testbed with traffic generators and traffic analyzers and shows results of measurements with the Horizon scheduling module. Finally, Section 5 concludes the paper.

2. OBS

OBS is widely discussed as a promising candidate for bringing more dynamic in optical networks. With a growing number of suggestions about the details of OBS, the picture of OBS gets less clear. Most of the proposals see three major aspects as definition for OBS systems [4]:

- Assembly of client layer data in data units with variable length, called bursts;
- Separation of control headers and payload data;
- No O-E-O conversion of the data bursts in the core network.

In OBS networks, Multi Protocol Label Switching (MPLS) [5] mechanisms can be applied easily. Packets with the same egress node as destination of the OBS network can be assigned to the same Forward Equivalent Class (FEC). Additionally, the FEC assignments can be differentiated for packets with different QoS-Classes.

*This work was partly funded within the MultiTeraNet programme by the German Bundesministerium für Bildung und Forschung.

At the ingress node of the OBS network, client layer packets are classified and assigned to the appropriate FECs [6]. All packets are stored in a packet buffer until the conditions for building a burst are fulfilled. Bursts can be sent, when the sum of packet lengths in the queue of one FEC exceed a certain value or when the waiting time of the first packet in the queue reaches a limit [7, 8, 9]. For the forwarding of the bursts, predefined labels are assigned to the bursts. These labels reduce the table lookup complexity in the core nodes and can be changed by the core nodes for reducing the label space in the network.

In the literature, the time scales for burst lengths vary from microsecond range [2] up to several milliseconds [14]. As bursts are transmitted completely in the optical domain, a burst header packet (BHP) is sent out-of-band for configuring the path just before the burst passes the node. Many OBS schemes introduce an offset-based approach: A time period called offset length before the burst is sent out, the burst header packet is sent on a separate control channel. This control channel is terminated at the next node, where all BHPs are processed for switching in the node. As the BHP processing takes some time, the BHP is delayed on its way to the egress node. The burst itself is transported without any additional delays thanks to transparent optical switching and thus, the offset shrinks. This fact shows, that the BHP processing must be performed very fast, otherwise the data burst catches up the BHP and the nodes are not configured in time. An additional need for fast BHP processing lies in the unwanted priorization of bursts with longer offsets. Several approaches for supporting quality of service (QoS) in OBS networks introduce additional QoS-offsets for priorizing bursts [10]. As investigations show, these QoS-offsets should be much greater than the offsets for compensation of the BHP processing [11]. On the other hand, all offsets contribute to the delay of the bursts at the edge node, so the total offset should be kept small for keeping buffers small, too. These two requirements can only be fulfilled by minimizing processing delays.

Depending on the implementation, BHP processing times depend on the number of wavelengths, the lengths of the bursts, the bursts arrival characteristics and the network load. A lot of scheduling algorithms need complex calculations and table lookup operations which seem to be not realizable in the necessary time limits. [12] and [13] show, that with processing times for single BHPs below 60ns an OBS networks with mean burstlengths of 10μs and 120 wavelengths can be supported with programmable logic devices available in the year 2002. A software based implementation works with mean burst lengths in the millisecond range [14].

Resource reservation algorithms with different complexity were introduced in literature. As an example algorithm the Horizon [3] scheduling scheme is used for evaluating the testbed introduced in this document. The realization complexity of this algorithm is moderate by a good performance for small offsets and without support of offset based QoS mechanisms [11].

Horizon is a reserve-a-limited-duration (RLD) reservation scheme. For each wavelength a scheduling horizon is stored. When a new BHP arrives at the reservation module, all wavelengths where identified, where the horizon is earlier than the announced start of the corresponding burst. Among the identified wavelengths, the one with the latest horizon is selected. This leads to the generation of the smallest unused voids and so to a minimized waste of bandwidth.

Fig. 1 shows an example scheduling process with Horizon for 3 wavelengths. An arriving BHP announces the arrival of a new burst. Wavelength WL2 can not be selected,

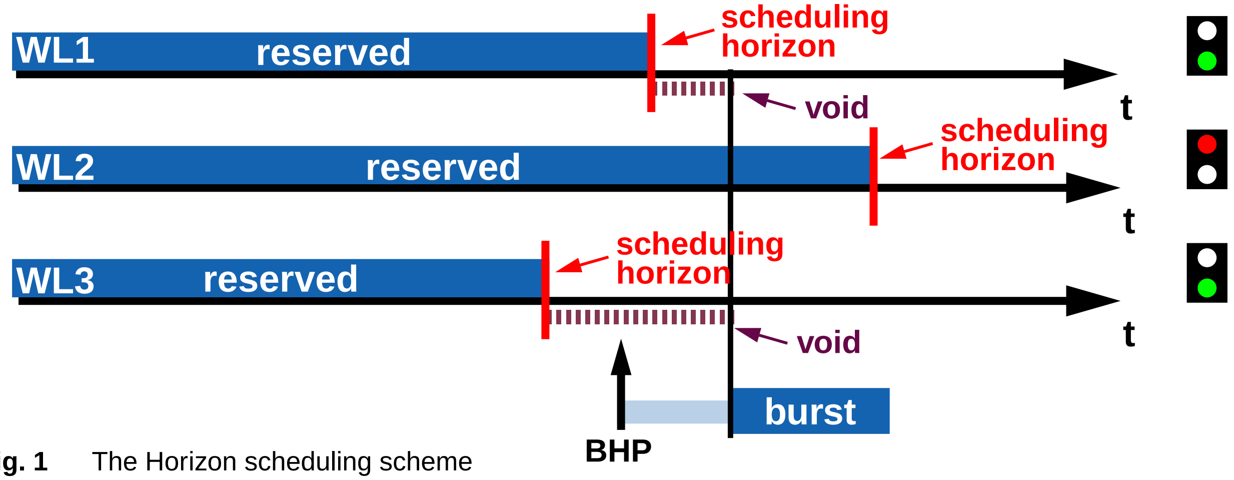

Fig. 1 The Horizon scheduling scheme

as the new burst would overlap with an already reserved one. WL1 and WL3 can be used and WL1 is selected, as the wasted bandwidth (void) is smaller than in the case of the assignment to WL3.

3. Architecture of the control part of OBS nodes

As BHPs and data bursts are separated and transmitted on different channels, burst reservation and switching can be treated as nearly independent tasks. **Fig. 2** shows the architecture of an OBS core node. A non-blocking optical cross connect (OXC) which supports full wavelength conversion connects the input and output fibers. The OXC is controlled by a calendar which receives all connection commands and sets the OXC just for the transmission intervals of the bursts. Wavelength converters allow the conversion of a burst from one wavelength to another.

BHPs are sent to the node on a specific control channel. This control channel is terminated in the node, a switching decision is derived from the label of the BHP and the BHP is forwarded to the corresponding output port controller (OPC). The resources are reserved and the OXC calendar is informed about timing and switching of each burst. The BHP is updated and sent on a control channel to the next node.

The fact that BHPs and data bursts are transmitted without any feedback from the data bursts to the BHPs allow the investigation of the control part of OBS nodes without detailed knowledge of the properties of OXCs for the data bursts. In the following, this paper will concentrate on the control part of OBS nodes.

3.1 Optical Switch Controller

The architecture of the optical switch controller (OSC) is shown in **Fig. 3** in more detail. It consists of three types of components. The input port controller (IPC), the electrical cross connect (EXC) and the output port controller (OPC). Additionally, a node-wide system clock is provided to all modules in the system. In the most OBS scenarios, all control functions of input and output ports do not need direct interaction with other input or output ports respectively. This allows a high degree of modularization and separation of the port controllers.

The first action in the input port controller (IPC) is the assignment of a timestamp which allows the recalculation of the offset time for the adaption of the BHP at the output port controller. This timestamp is attached to the BHP during its way through the switch controller. With the label, which is contained in the BHP, the IPC takes a switching decision and forwards the BHP via the EXC to the appropriate OPC.

The OPC is the most complex part of the switch controller. It manages the traffic of all wavelength channels of the corresponding fibre and generates the BHPs to inform the next node on the link. The architecture of the OPC can also be seen in **Fig. 3**.

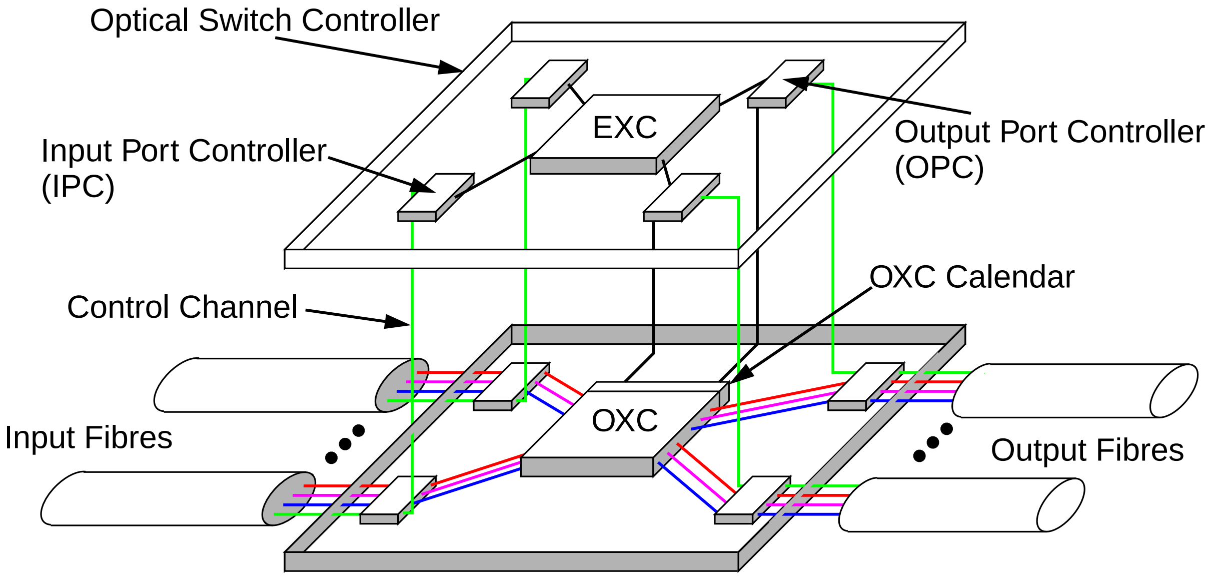
Optical Switch Controller
EXC
Output Port Controller (OPC)
Input Port Controller (IPC)
OXC Calendar
Control Channel
OXC
Input Fibres
Output Fibres

Fig. 2 Architecture of OBS core node

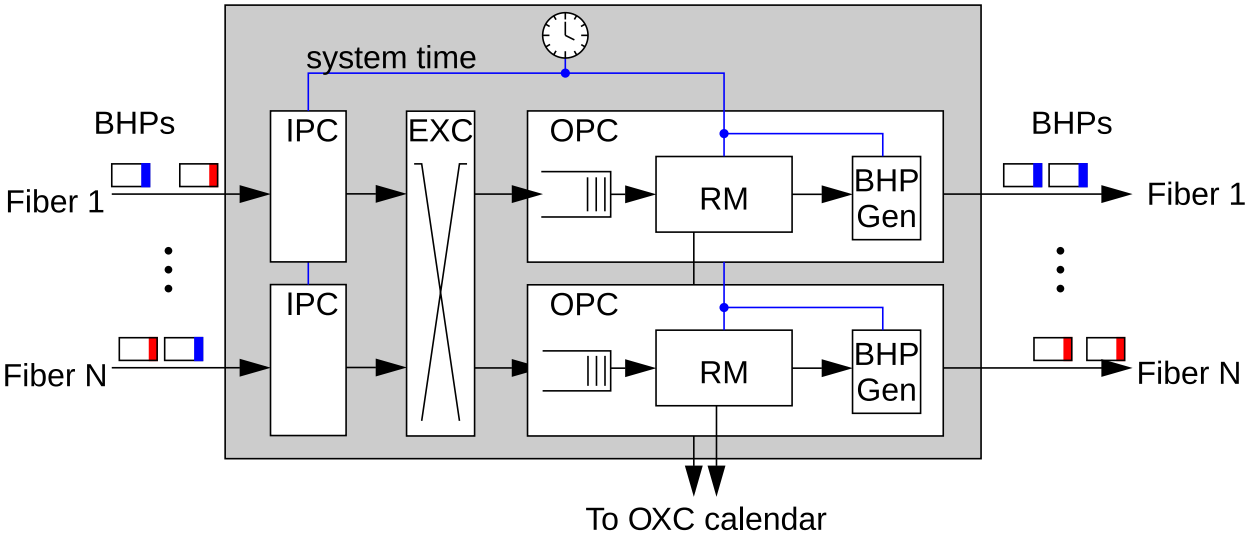

Fig. 3 Architecture Optical Switch Controller

The BHPs of all input ports announcing bursts for the corresponding output port are forwarded to the OPC. The processing of the BHPs is performed one after another for guaranteeing data consistency. Parallel processing of several BHPs is possible under special circumstances, too, but will not be discussed in this document.

The BHPs are stored in a queuing device and read out for processing. The reservation module (RM) searches for all wavelengths, which are available during the transmission time of the burst. From the available wavelengths, one is selected and reserved for this burst. This selection and reservation mechanism is called burst scheduling and can be done with different algorithms. When the burst is assigned to a wavelength, the next node on the link and the OXC must be informed about the properties of this burst. If the labelling scheme supports label changing, a new label must be inserted in the BHP, too. The propagation delay through the switch controller can be determined from the system time and the timestamp of the BHP. The offset time information in the BHP must be reduced by the value of the propagation delay and adapted in the offset field of the BHP. The BHP generator updates these information in the old BHP and sends this BHP to the next node.

3.2 Timing requirements to scheduling modules

As stated above, the control part of an OBS switch can be separated from the optical data part. As OXCs with several ports and tens of wavelengths are not available, the realization of switch controllers and OXC cannot be integrated at the moment. This problem, that no real data traffic with feedback from users and network can be used for the stimulation of the OBS system, can be solved by traffic generators, which generate BHPs without data bursts. The separation of the control part opens the possibility to investigate the behaviour of system which will be realized in future. As no data bursts traverse the link at the same time, the whole system can be operated in a kind of ’slow motion’. This operation can be used for emulating a control part with hardware components with speeds, which will be available in the future, just by extrapolating system clock frequencies and link capacities.

The system frequencies of logic devices depend on the complexity of combinatorial logic, the propagation delays on the chip and on the used technology. Full custom designed ICs are the fastest option for realizing logic devices, but the production costs per device are only moderate for a high number of produced ICs. For smaller device counts or for higher flexibility of devices already working in the field, programmable logic devices like field programmable gate arrays (FPGA) are used in many applications. Especially in the area of networking devices, where new data services are introduced in short time intervals, the possibility for reconfiguring the system at the customer, leads to a wide usage of FPGAs.

In the last years, speed and complexity of FPGAs raised very fast. Todays FPGAs perform well for several OBS scheduling mechanisms. The next generations of FPGAs will allow higher operation frequencies and higher logic complexity and can support more wavelengths or more sophisticated scheduling algorithms.

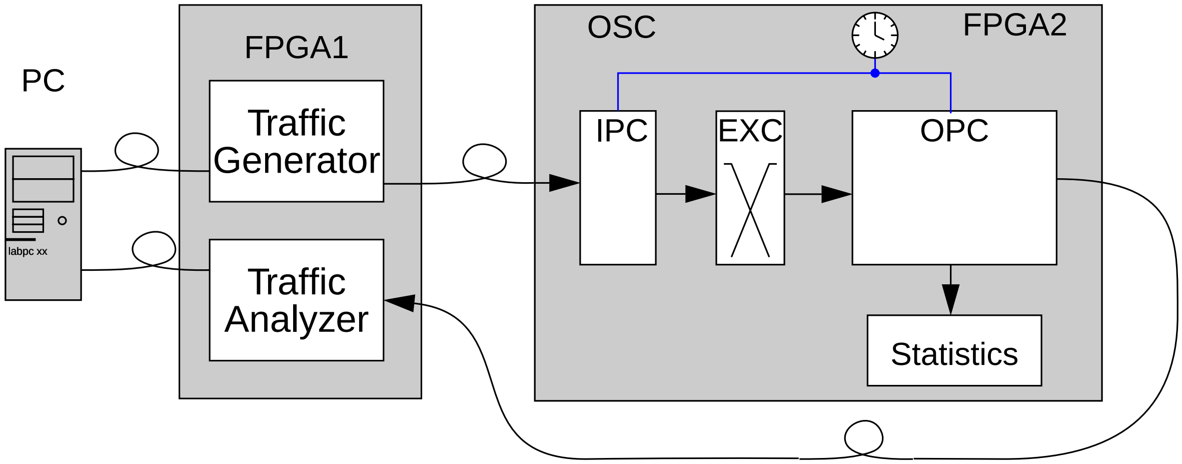

Fig. 4 Structure of the Testbed

The transmission rates in dense wave division multiplexing (DWDM) systems grew in the last years very much. The reservation module treats the length indications as time intervals. This approach makes the system independent from the real transmission rates on the links. If the link speed of the actual used control channel is not sufficient for transporting all BHPs or if the performance of the used FPGA is not fast enough, all timings can be scaled down for emulating the system in slow motion.

4. Testbed and measurement

In the systems design lab of the Institute of Communication Networks and Computer Engineering (IKR) a testbed is developed which allows the realization of all main components for the control part of an OBS network. It consists of computer systems for realizing traffic sources with different statistical behavior as well as for analyzing the output data of the emulated control network. With an interface to the IKR simulation library [15] a large set of traffic generators can be used in the realized system. Time critical functions of data sources and analyzing elements are supported by hardware elements; the more complex generation of random data and the analysis are processed with software. The Optical Switch Controller (OSC) itself is realized with the IKR hardware platform as well.

The testbed for realizing and evaluating an OSC is shown in **Fig. 4**. The PC derives traffic patterns with burst length distributions and inter arrival time distributions from the IKR simulation library. A burst of BHPs with traffic pattern information is packed in an ethernet frame and sent to the traffic generator in FPGA1. The traffic generator extracts the BHPs with their timestamps a sends them according to the traffic patterns to the input port controller (IPC) of the OSC. The electrical cross connect (EXC) is only integrated for future use when nodes with several input and output ports will be investigated. Actually, the traffic pattern for multiple input ports are generated by the software on the PC. The output port controller (OPC) performs the scheduling task and generates new BHPs for the next node. These BHPs are sent to the traffic analyzer which adds timestamps to the BHPs and sends the BHPs with the timestamps to the PC for tracing the system behaviour.

As the whole system is developed from the IKR, it is possible to intervene in all stages of the system for getting debugging information. Statistic modules where placed at interesting points in the hardware design for getting detailed informations about the processing of the BHPs.

These statistic modules count all processed BHPs, all dropped bursts and the distribution of the processing delays for the BHPs. Further statistics like wavelength conversion ratios and queue lengths can be added easily. All these statistical informations can be read out via a dedicated control bus of the UHP and are sent back to PC for analysis.

Fig. 6 shows the format of the BHP defined by us for the system. It contains 16 bytes of header information. It consists of a 2 byte label field, 1 byte for wavelength numbers and 1 byte reserved for extensions. The offset and burst lengths are both coded in 4 bytes with an accuracy of $1ns$. An additional 4 byte Burst ID field is included for tracking the packets in the testbed. For transmission of the BHPs, layer1 of optical gigabit ethernet links is used. With an modified framing structure, the transmission of a maximum of about $5 \cdot 10^6$ BHPs per second is possible.

31		0
Label	Wavelength	Reserved
Offset		
Burst Length		
Burst ID		

Fig. 6 BHP format

The described system is realized and working in the laboratory. **Fig. 5** shows the results of measurements with a Horizon reservation module. The bursts lengths are uniformly distributed in the range from $60\mu s$ to $140\mu s$. The inter arrival times are negative exponential distributed with varying link load. The fixed offset is $32\mu s$ but has no influence on the burst losses, as long as all offsets have the same length [11]. This behavior was confirmed by our experiments.

Fig. 5 shows the burst loss probability depending on the load. The error bars show the standard deviation which is derived from multiple sets of measurements. The results show the behavior which was analytically predicted by J. Turner in [3]. The burst loss probabilities raise with growing load. For higher numbers of wavelengths, the losses decrease significantly.

The testbed works for the used parameters. But the experiments show, that the time accuracy of the used computer equipment is an important factor. For higher numbers of wavelengths, the number of BHPs per second leads to shorter intervals for sending BHPs with timestamps to the traffic generator. As the input queue in the traffic generator is limited by the available memory blocks in the used FPGA to about 500 BHPs, the jitter of the intervals for sending the BHPs from the PC must be very low. For allowing higher jitter, the input queue in the generator should be enlarged by additional external memory or larger FPGAs.

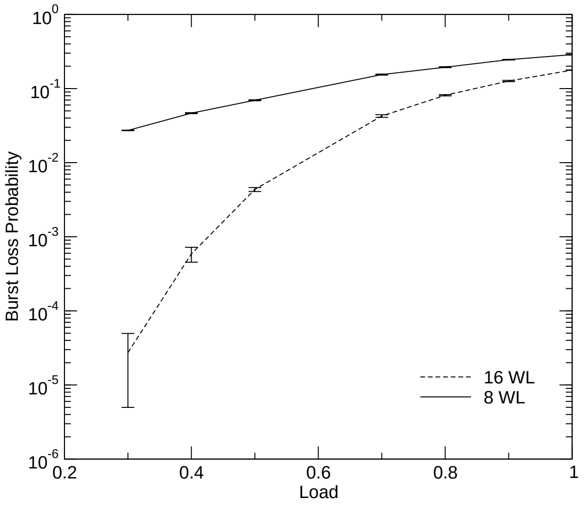

Fig. 5 Burst Losses for 8 and 16 wavelength depending on load

5. Conclusion and Outlook

This paper introduces a testbed for testing and evaluating the performance of implemented scheduling modules. For the Horizon scheduling scheme, a scheduling module was realized and its performance was evaluated by mesurements. The results show a good correspondence to the behavior predicted by simulations.

The implementation of the control part of an OBS node helps to identify realization specific problems and consequences on the system performance. In this work, the feasibility of OBS control system is proven for the Horizon scheduling scheme. The testbed allows the investigation of realization specific aspects for offset based scheduling algorithms. Even for scenarios with constant or no offsets, the system can be adapted easily. The actual realization is based on logic designs synthesized on programmable logic devices. The universal hardware platform provides interfaces for integrating microcontrollers in the system which allow software based realizations, too. The behavior of the Horizon module shows a good correspondence between the simulation of the Horizon module and its implementation in our testbed.

The research activities are still in progress, additional investigations on realizing other scheduling algorithms, timing uncernities introduced by synchronization processes and accelerating the system by parallel processing of BHPs will be done in future. The author designed and implemented a scheduling module for the Just-Enough-Time (JET) reservation algorithm [13]. This and other new modules will be integrated in the testbed and evaluated by measurements.

The system is designed extensible, which allows the integration of additional input and output ports. With additional ports the investigations of a single node can be improved by using multiple traffic generators. Multi-hop paths can be studied with usage of several nodes, too.

This work should contribute to the identification of constraints and possibilities of the realization of OBS control systems and add a piece to the picture how OBS implementations could work finally.

Acknowledgements

The author would like to thank Christoph Gauger for many fruitful discussions in the context of OBS and all his assistance for connecting the SimLib to the system. He also thanks Simon Hauger for his work on the hardware platform with the traffic generators and analyzers. He also thanks Martin Köhn for his helpful comments on the previous versions of this paper.

Bibliography

[1] M. Yoo, C. Qiao: “A new optical burst switching protocol for supporting QoS.” *Proceedings of SPIE Photonics East ‘98 Conference on All-Optical Networking*, Boston, Nov. 1998, pp. 396-405.

[2] C. Qiao, M. Yoo: “Optical burst switching (OBS) – a new paradigm for an optical Internet.” *Journal of High Speed Networks*, Vol. 8, No. 1, Jan. 1999, pp. 69-84.

[3] J. S. Turner: “Terabit burst switching.” *Journal of High Speed Networks*, Vol. 8, No. 1, Jan. 1999, pp. 3-16.I.

[4] C. M. Gauger: „Trends in optical burst switching.“ *Proceedings of SPIE ITCOM 2003*, Orlando, USA, Sept. 2003.

[5] E. Rosen, A. Viswanathan, R. Callon: “Multiprotocol Label Switching Architecture”, RFC 3031, IETF, January 2001

[6] Dolzer, K.; Payer, W.: „On aggregation strategies for multimedia traffic“ *Proceedings of the 1st Polish-German Teletraffic Symposium (PGTS 2000)*, Dresden, 2000.

[7] A. Ge, F. Callegati, L.S. Tamil: „On optical burst switching and self-similar traffic“ *IEEE Communications Letters, Vol. 4, No. 3*, March 2000, pp. 98-100

[8] K. Dolzer: “Assured Horizon – A new combined framework for burst assembly and reservation in optical burst switched networks.” *Proceedings of the European Conference on Networks and Optical Communications (NOC 2002)*, Darmstadt, June 2002.

[9] G. Hu, K. Dolzer, C.M. Gauger: “Does burst assembly really reduce the self-similarity?” *Proceedings of the Optical Fiber Communication Conference (OFC 2003)*, Atlanta, March 2003

[10] M. Yoo, C. Qiao, S. Dixit: “QoS Performance of Optical Burst Switching in IP-over-WDM networks.” *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 10, Oct. 2000, pp. 2062-2071.

[11] K. Dolzer, C. M. Gauger, J. Späth, S. Bodamer: “Evaluation of reservation mechanisms for optical burst switching.” *AEÜ International Journal of Electronics and Communications*, Vol. 55, No. 1, Jan. 2001.

[12] S. Junghans, C. M. Gauger: “Architectures for Resource Reservation Modules for Optical Burst Switching Core Nodes.”, *Beiträge zur 4. ITG-Fachtagung Photonische Netze*, Leipzig, Germany, May 2003, pp. 109 - 117.

[13] S. Junghans, C. M. Gauger: “Resource reservation in optical burst switching: architectures and realizations for reservation modules”*Proceedings of SPIE OptiComm*, Dallas, Texas, USA, Oct. 2003, pp. 409 - 413.

[14] J. Xu, C. Qiao, J. Li, G. Xu: “Efficient Channel Scheduling Algorithms in Optical Burst Switched Networks.” *Proceedings of INFOCOMM*, San Francisco, USA, Apr. 2003, pp. 2268-2278.

[15] Institute of Communication Networks and Computer Engineering, University of Stuttgart: “The IKR simulation library.”, `www.ikr.uni-stuttgart.de/IKRSimLib/`