

Just-in-Time Optical Burst Switching Implementation in the ATDnet All-Optical Networking Testbed

Iliia Baldine, Mark Cassada, Arnold Bragg[†], Gigi Karmous-Edwards, and Dan Stevenson

MCNC Research and Development Institute
Advanced Networking Research Division
PO Box 13910
Research Triangle Park NC 27709 USA

Abstract—We describe: (i) the architecture of an optical burst switched (OBS) demonstration network overlaying the ATDnet transparent all-optical testbed, and (ii) experiments underway in the testbed. The OBS overlay uses a simple hardware-based protocol embedded in OBS network controllers to manage commercial off-the-shelf DWDM switches. Data paths are all-optical and completely transparent, and can carry analog or digital traffic in any format, data rate, and modulation scheme. Experiments with latency- and jitter-sensitive HDTV transmission, petabyte file transfers, and immersive real time visualization of satellite imagery over the OBS network are ongoing. Parallel research on transport protocols, QoS-aware routing protocols, adaptors for an OBS LAN, and a network management architecture will be applied as completed. This is the first just-in-time (JIT) OBS field trial known to the authors.

I. INTRODUCTION

The infrastructure of next-generation optical internetworks will almost certainly be based upon wavelength division multiplexing (WDM) technologies. These technologies differ in the number of wavelengths per optical fiber (a function of optical channel spacing, fiber spectrum, laser and filter complexity, cost, etc.), and are often classified in that way; viz., coarse WDM (8 or fewer wavelengths per fiber), dense (tens), super-dense (hundreds), ultra-dense (thousands).

Individual wavelengths may be assigned to a single user or source, or used to carry aggregate traffic that is multiplexed in some way – by time (e.g., TDM), by traffic flow, by packet (e.g., all-optical packet switching [1,2]), by burst (e.g., optical burst switching [3,4]), etc. All-optical circuits tend to be inefficient for traffic that has not been groomed or statistically multiplexed, and all-optical packet switching will not be practical until cost-effective optical buffering and header parsing implementations are commercially available.

Optical burst switching (OBS) is a technical compromise. It does not require optical buffering or packet-level parsing, and it is more efficient than circuit switching when the sustained traffic volume does not consume a full wavelength. This paper describes the architecture of an OBS demonstration network overlaying the ATDnet transparent all-optical testbed, and summarizes experiments underway in the testbed. The OBS overlay uses a simple hardware-based

protocol embedded in OBS network controllers to manage commercial off-the-shelf dense WDM optical switches.

Optical burst switching is briefly described in Section II; the testbed configuration is summarized in Section III; network control hardware, architecture and protocols are the focus of Sections IV and V; and experimental plans and conclusions are summarized in Sections VI and VII.

II. OPTICAL BURST SWITCHING

An optical burst is usually defined as some number of (nearly) contiguous packets destined for a common egress point. Each burst is ‘announced’ by a control message that is typically sent out-of-band over a separate signaling channel; e.g., the **SETUP** message in Fig. 1. The OBS source node does not await confirmation that an end-to-end path has been established; instead, it begins transmitting the optical data burst δ seconds after receiving a **SETUP ACK** from the ingress OBS node. The **SETUP** message informs each switch along the path of the impending data burst so that switch configuration and routing decisions can be made prior to the burst’s arrival. Bursts are not buffered in most OBS architectures, so a burst may be blocked at an intermediate switch in cases of congestion or output port conflicts. Long bursts may require **KEEPALIVE** messages to maintain state. Resources may be implicitly released if the burst’s length is

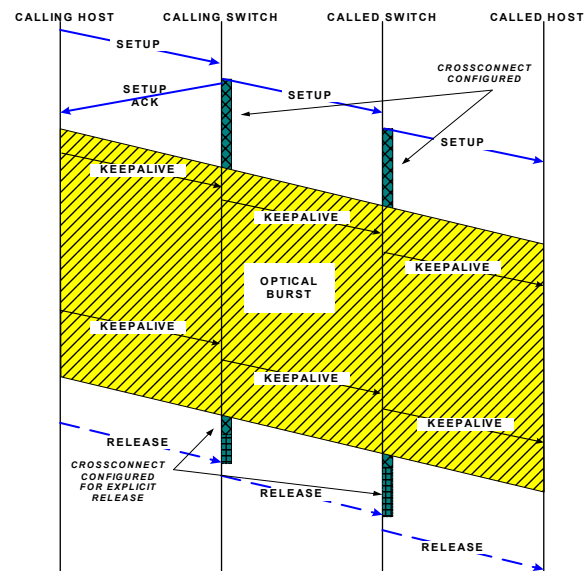


Fig. 1. Burst with **SETUP**, **KEEPALIVE**, and explicit **RELEASE** messages.

[†] Corresponding author, +1 (919) 248-1988, abragg@anr.mcnc.org.
This work was supported by the intelligence community under ARDA contracts MDA 904-00-C-2133 and MDA 904-02-C-0482.

known (and is conveyed with the **SETUP** message), or explicitly **RELEASED** as in Fig. 1.

There are several major OBS variants. They differ in a number of ways: **(i)** how they reserve resources (e.g., ‘tell-and-wait’, ‘tell-and-go’), **(ii)** how they schedule and release resources (e.g., ‘just-in-time’ [5,6], ‘just-enough-time’ [7]), **(iii)** hardware requirements (e.g., novel switch architectures optimized for OBS, commercial optical switches augmented with OBS network controllers), **(iv)** whether bursts are buffered (using optical delay lines or other technologies), **(v)** signaling architecture (in-band, out-of-band), **(vi)** performance, **(vii)** complexity, and **(viii)** cost (capital, operational, \$/Gbit, etc.). Bibliographies of OBS research – switching architectures, performance, control protocols, burst scheduling – are available from many sources; e.g., [8-10].

III. ATDNET TESTBED CONFIGURATION

The Advanced Technology Demonstration Network (ATDnet) is a high-performance optical testbed in metropolitan Washington DC [11]. ATDnet was established by DARPA as an experimental platform for network research and demonstration initiatives involving Federal agencies and research collaborators. ATDnet interconnects sites over OC-48 (2.5 Gbit/sec) dense WDM optical links. In November 2002, the research team installed proof-of-concept tell-and-go just-in-time (JIT) OBS network controllers at three ATDnet sites – the Defense Intelligence Agency (DIA), the Naval Research Laboratory (NRL), and the National Security Agency’s Laboratory for Telecommunications Science (LTS) – as part of an experimental OBS overlay. The three-site OBS testbed configuration is shown in Fig. 2.

Each testbed site has a SGI *Origin 2000* host running IRIX or Linux [12], and a Firstwave *SIOS 1000* MEMS optical crossconnect (OXC). The sites are connected by dense WDM optical fiber links. One wavelength per fiber is dedicated to the signaling channel; the remainder are used for digital and analog data transmission. JITPAC (‘Just-in-Time Protocol

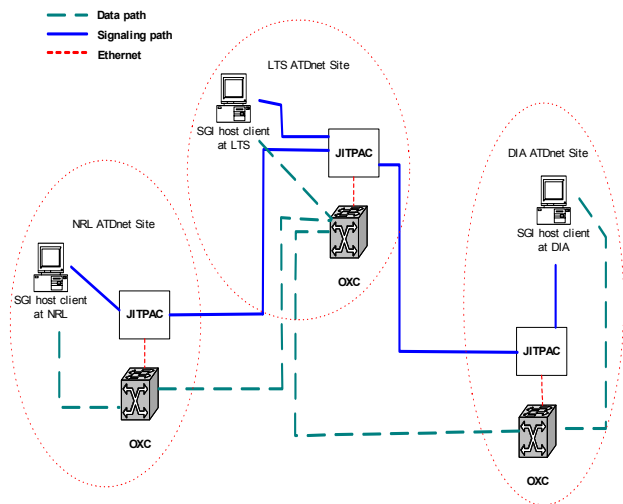


Fig. 2. ATDnet OBS testbed configuration.

Acceleration Circuit’) network controllers implement the testbed’s out-of-band signaling and control protocol.

Fig. 3 shows the NRL testbed setup – the OXC is in the background (slightly left of center), and two JITPACs – primary and a spare – are mounted in a standard 19 inch (48.3 cm) rack (foreground). Each JITPAC is connected to its site’s OXC over a dedicated Ethernet segment, and controls its OXC via remote procedure calls over that segment. Each JITPAC is also connected to its site’s SGI host, and to JITPACs at adjacent site(s) over ATM PVCs. Most of the JIT OBS protocol is implemented in hardware. There is a thin software layer to facilitate transmission of signaling messages over ATM to/from a JITPAC’s FPGA, and to issue configuration commands to its OXC.

IV. NETWORK CONTROL HARDWARE

The proof-of-concept JITPAC controllers were designed and built at MCNC-RDI, and use gate array microcode jointly developed at MCNC-RDI and NC State University [14]. Each JITPAC contains a Motorola MPC8260 PowerPC processor, an Altera EP20K400C field programmable gate array (FPGA), 4 MB SDRAM, a 64 MB SDRAM DIMM module, 16 MB of flash ROM, two serial ports (RS-232), a 155 Mbit/sec UTOPIA ATM interface for the signaling channel, an Ethernet auxiliary/craft interface, and an Ethernet interface for OXC control and configuration. A conceptual schematic diagram of a proof-of-concept JITPAC is shown in Fig. 4.

A conceptual schematic diagram of a proof-of-concept JITPAC’s Altera EP20K400C FPGA is shown in Fig. 5. The FPGA contains four modules: ingress and egress message processing engines, a register access block, and a crossbar. Signaling messages arrive at the register access

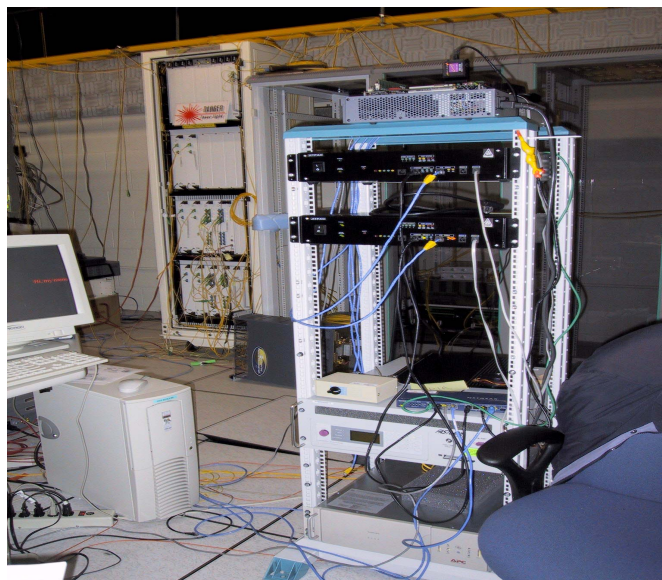


Fig. 3. JITPACs and commercial MEMS OXC at the NRL ATDnet site.

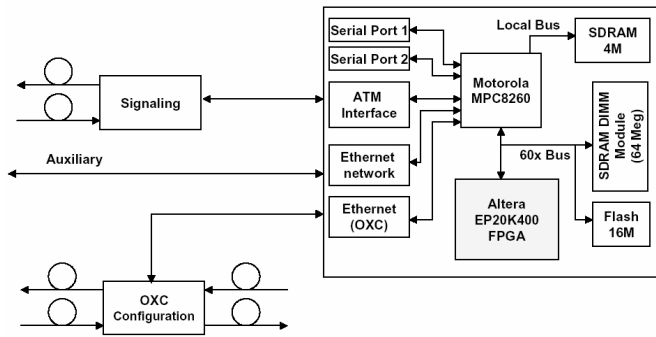


Fig. 4. Conceptual schematic diagram of a proof-of-concept JITPAC.

block, pass through the ingress message engine, the cross bar module, the egress message engine, and back through the register access block. The FPGA has a signaling throughput latency between 1.90 μ sec and 7.56 μ sec at 33 MHz.

Bill-of-materials cost is about \$4,000 per JITPAC. The JITPAC network controllers are functionally generic, and thus can be adapted for use with any commercial optical switch and host client. As noted, signaling is out-of-band; the measured signaling channel rate is about 80,000 signaling messages per second (~ 12.5 μ sec per signaling message) for the proof-of-concept JITPAC testbed controllers.

V. JIT OBS ARCHITECTURE AND PROTOCOL

The JIT OBS protocol was jointly defined by MCNC-RDI and NC State University under the ARDA-sponsored *JumpStart* project [15]. (ARDA focuses on high-performance data communications that cannot be addressed by technologies used in today's Internet [16].) JIT OBS is an open, published protocol with multicast extensions [17-18].

A. Signaling Channels

As noted, JIT OBS is an out-of-band, best-effort, packet-based signaling protocol implemented in hardware. Signaling messages undergo O/E conversion and are processed by a JITPAC at each intermediate node. If the signaling channel is congested, signaling packets are buffered and may incur queuing delay. The signaling channel is engineered so that signaling packets are unlikely to be dropped from the queue.

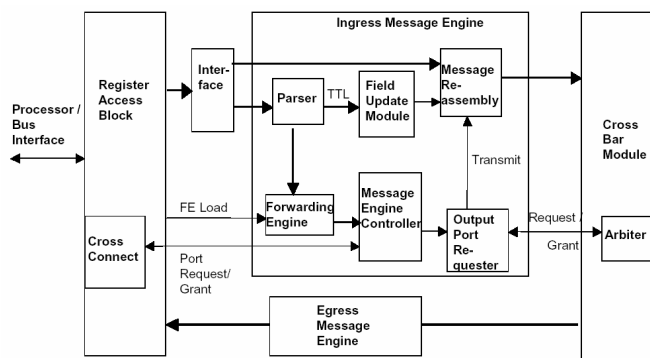


Fig. 5. Conceptual schematic diagram of a proof-of-concept FPGA.

B. Data Channels

Data transport in JIT OBS is transparent to the intermediate nodes – no O/E conversion is required on the data channels, and no global time synchronization is required among the OBS network elements. Data bursts are not buffered, so data transmissions may experience occasional blocking in the optical core. Research has focused on techniques to reduce blocking probabilities to acceptable levels for delay- and jitter-sensitive traffic [19]. Transparency means that data channels can convey analog (*e.g.*, radar) and digital traffic in any format, data rate or modulation scheme.

C. Dimensions

JIT OBS currently has six architectural ‘dimensions’:

(i) Analog vs. digital – both analog and digital bursts are supported. End-to-end transport-layer negotiation regarding a burst's format, modulation, and rate may be required.

(ii) Unicast vs. multicast – bursts are unicast by default. Multicast supports several different types of joins (source-managed, leaf-initiated, or both) and scope limits (leaves are restricted to certain domains, or no restrictions).

(iii) Explicit vs. implicit release – resources may be released in two ways. Implicit releases are timed, but require a burst's duration to be known and conveyed with the **SETUP** control message. Explicit releases free resources via a **RELEASE** control message as shown in Fig. 1. Some OBS variants require burst lengths to be estimated and announced prior to initiation to facilitate burst scheduling and resource reservation. The testbed version of JIT OBS does not support delayed resource reservations.

(iv) Burst vs. light path – burst durations are not restricted by the architecture or protocol. **KEEPALIVE** control messages may be required to maintain state as in Fig. 1. The decision depends on the burst's holding time and the network's diameter. Bursts exceeding the network diameter are ‘light paths’ and require periodic out-of-band **KEEPALIVES**.

(v) Persistent vs. on-the-fly paths – a JIT OBS session consists of one or more bursts. A persistent path (*i.e.*, a pinned route) is an option for sessions that require ordered delivery of bursts with negligible processing-induced jitter. Light paths are another type of session. Light paths retain OXC configurations for the duration of a session, while persistent paths allow bursts from two or more sessions to be interleaved and OXCs to be reconfigured during these sessions. Sessions with on-the-fly paths make per-burst routing/forwarding decisions (rather than pinning the route).

Light paths are somewhat analogous to (short-lived) provisioned circuits, persistent paths are somewhat analogous to burst-multiplexed fixed-route pipes or tunnels, and on-the-fly paths are dynamically routed bursts. Note that these OBS ‘circuits’ and ‘tunnels’ can be provisioned in tens of microseconds (depending on switch configuration times), and once provisioned, will have lifetimes ranging from a few milliseconds to many hours.

(vi) Quality of service – JIT OBS supports burst-level prioritization, burst-level preemption, and quality of service (QoS) provisioning. A combination of dimensions can be used to satisfy service-level guarantees.

D. Message Formats

JIT OBS message formats are composed of flexible information elements (IEs), with a common header and separate hardware-parsable (hop-by-hop significant) and software-parsable (end-to-end significant) components. The structure is shown in Fig. 6. IEs have a TLV (type, length, value) structure. This allows IEs to migrate from ‘softpaths’ (i.e., processed in software) to ‘hardpaths’ (i.e., processed in hardware) as JITPACs are enhanced. IEs that require processing by intermediate optical burst switches or that are closely coupled to the JIT OBS signaling protocol are usually hardpath. IEs that carry end-to-end information or that are not closely linked to JIT OBS signaling are usually softpath.

The hardpath and softpath subheaders contain information about the number of hard and soft IEs in the message. The hardpath subheader uses a bit mask to indicate its IEs. Hardpaths are limited to 64 IEs, but headers are hardware-parsable. The softpath subheader has a <type, offset> vector for each IE. The number of softpath IEs is not limited, but software must scan the entire IE vector area to discern which soft IEs are present. The common header contains the protocol type and version, the message type and length, and a pointer to the beginning of the softpath subheader. This allows hardware and software to parse their subheaders in parallel. Each message has a CRC-16 for the common header and hardpath IEs, and a CRC-16 for the softpath IEs.

The same IE format is used by all of the JIT OBS management protocols – routing, connection management, network management, etc. This greatly simplifies hardware and software, but requires flexibility to accommodate future requirements.

E. Addressing and Forwarding

JIT OBS addressing is hierarchical, addresses are variable length (up to 2,048 bits), and the address space and address

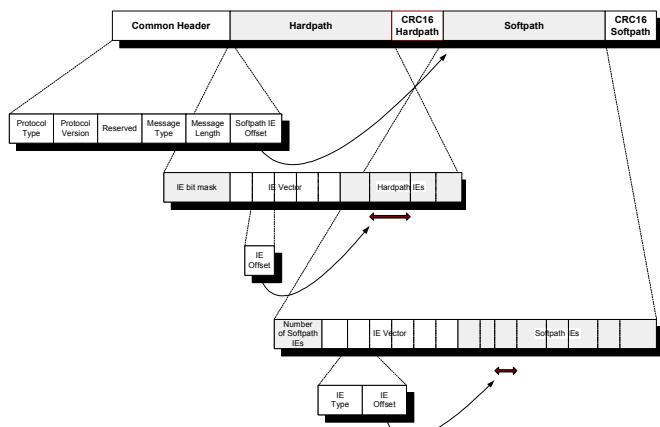


Fig. 6. JIT OBS message structure.

structure provide maximum flexibility to the assigning authority (in contrast to fixed length addressing schemes where blocks of addresses are typically allocated to different entities). The addressing hierarchy and the physical network topology are orthogonal concepts.

Fig. 7 shows an OBS realm with three levels, eight administrative domains, and six nodes. The top level contains domains 0xA and 0xB, and 4 bits are allocated to the address structure. Domain 0xA contains second-level domains 0x01, 0x02 and 0x03, and 8 address bits. Nodes represent the lowest level, and are allotted 8 bits. (Node port numbers are also shown in Fig. 7.) Domain 0xB has two second-level domains and 16 address bits. Third-level nodes are allocated 12 bits. The numbers of levels and of assigned bits are arbitrary, and are limited only by the address space. Full node addresses for Fig. 7 are shown in Table 1(a).

The JIT OBS routing protocol views each level as a set of peer-communicating entities. Logical levels are expressed as forwarding tables – one for each level in the hierarchy – and are stored in each node. A node forwards a burst by taking the longest match between its address and the destination address, and then using the appropriate domain-level table to forward the message. The forwarding table for node 1 (address = 0xA.0x01.0x1A) is shown in Table 1(b).

F. Performance

Unlike some OBS variants, JIT OBS does not use a delayed reservation burst scheduler. Resources are allocated upon receipt of a **SETUP** control message, and are released either explicitly (upon receipt of a **RELEASE** control message) or implicitly (if a burst’s duration is included with its **SETUP** message). More sophisticated delayed reservation and void-filling schedulers can improve throughput for bursts transiting long paths, but these schedulers are complex and may introduce significant processing overhead. Under heavy load, utilization in a JIT OBS network may be slightly lower than for some OBS variants for the same blocking probability. JIT OBS compensates for this via mechanisms for loss-, error-, order-, delay-, and jitter-sensitive traffic.

VI. TESTBED EXPERIMENTS, TESTS AND DEMONSTRATIONS

A number of experiments, tests and demonstrations are either underway or planned for 2003-4. These will be conducted by

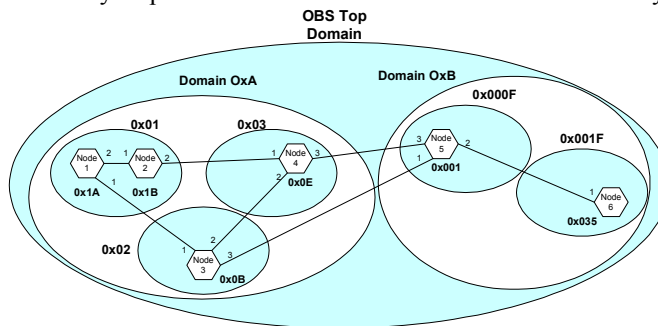


Fig. 7. JIT OBS address structure for the 3-level, 8-domain example.

(A) NODE ADDRESSES

Node	Address
1	0xA.0x01.0x1A
2	0xA.0x01.0x1B
3	0xA.0x02.0x0B
4	0xA.0x03.0x0E
5	0xB.0x000F. 0x001
6	0xB.0x001F. 0x035

TABLE I
(B) FORWARDING TABLE FOR NODE 1

Target	Port	By way of
[Prefix = Top Level]		
0xB	1	0xA.0x02.0x0B
0xB	2	0xA.0x01.0x1B
[Prefix = 0xA]		
0x02	1	0xA.0x02.0x0B
0x03	1	0xA.0x02.0x0B
0x03	2	0xA.0x01.0x1B
[Prefix = 0xA.0x01]		
0x1B	2	Direct

MCNC-RDI and collaborators from academia and government, and are briefly summarized in this section.

Baseline performance – JIT OBS stress-testing (OXC configuration tests, offset and **KEEPALIVE** timer configuration, simulated path delays, simulated congestion on the signaling channels, alternate route behavior, *etc.*), performance of the proof-of-concept FPGA, and performance implications of using a commercial MEMS-based OXC with switching times of a few milliseconds *vs.* experimental lab devices that switch in microseconds or nanoseconds.

Applications – in November 2002, the research team began trials by successfully switching a 1.5 Gbit/sec HDTV signal over the JIT OBS testbed as optical information. No O/E conversion was performed on the data channel.

Several testbed applications are underway. The first involves petabyte file transfers using *gridFTP*. The transport protocol is a NASA-sponsored enhancement of *Scheduled Transfer (ANSI INCITS 337-2000)* for JIT OBS, running on IRIX and Linux. Immersive real time visualization of latency-sensitive satellite imagery is planned for 2004.

Ancillary research – MCNC-RDI and NC State have several ancillary JIT OBS research efforts underway. Relevant results will be applied as work is completed, and will drive future testbed experiments and demonstrations:

- (i) ARDA-sponsored research into a robust, QoS-aware routing architecture and protocol(s) for JIT OBS;
- (ii) ARDA-sponsored research into a network management architecture and protocol(s) for JIT OBS;
- (iii) NASA-sponsored development of network adaptors for a LAN architecture based on JIT OBS signaling;
- (iv) NASA-sponsored research into a new transport layer and burst assembler/scheduler optimized for JIT OBS;
- (v) a cost and performance analysis of JIT OBS;
- (vi) security, authentication, and accounting extensions.

VII. CONCLUSIONS

This is the first operational JIT OBS field trial known to the authors (see also [6, 13]). Unlike some OBS variants, JIT OBS is a simple tell-and-go protocol embedded in inexpensive network controllers. It does not require optical buffers, global time synchronization, in-band signaling, or a delayed reservation scheduler. All OBS variants perform

reasonably well in simulations and laboratory trials, but their implementations will likely differ by several orders of magnitude in cost and complexity.

We expect the testbed trials to confirm that more complex and expensive OBS variants have marginally better performance than JIT OBS in some situations; *viz.*, when switch configuration times are short relative to the distance (in time) from the source to the OBS ingress switch. We also expect that JIT OBS mechanisms to support loss-, error-, order-, delay-, and jitter-sensitive traffic (outlined in Section V) will compensate for most or all performance disparities.

ACKNOWLEDGMENT

The authors gratefully acknowledge the valuable contributions of Wai Ng (NSA LTS), Hank Dardy and Linden Mercer (NRL), Ray McFarland, and the constructive comments of the anonymous reviewers.

REFERENCES

- [1] S. Yao, S. Dixit, and B. Mukherjee, "Advances in photonic packet switching: an overview," *IEEE Communications*, **38**(2), 84-94 (2000).
- [2] J. Turner, "Terabit burst switching," *J. High Speed Networks*, **8**(1), 3-16 (1999).
- [3] C. Qiao and M. Yoo, "Optical burst switching (OBS) – a new paradigm for an optical Internet," *J. High Speed Networks*, **8**(1), 69-84 (1999).
- [4] L. Xu, H. Perros, and G. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE Communications*, **39**(1), 136-142 (2001).
- [5] G. Hudek and D. Muder, "Signaling analysis for a multi-switch all-optical network," in *Proc. ICC 1995*, (IEEE, 1995).
- [6] J. Wei and R. McFarland, "Just-in-time signaling for WDM optical burst switching networks," *J. Lightwave Technology*, **18**(12), 2019-2037 (2000).
- [7] M. Yoo and C. Qiao, "Just-enough-time (JET): a high speed protocol for bursty traffic in optical networks," in *Proc. LEOS Tech. Global Information Infrastructure*, (IEEE, August 1997).
- [8] http://www.cs.buffalo.edu/~yangchen/OBS_Pub_year.html
- [9] <http://www.ind.uni-stuttgart.de/~gauger/BurstSwitching/HTML/>
- [10] <http://www.utdallas.edu/~vinod/obs.html>
- [11] <http://www.atd.net/>
- [12] <http://www.sgi.com>
- [13] G. Hudek and I. Richer, "Signaling protocol implementations on a burst-switched all-optical networking testbed," in *Proc. SPIE OFC Conf. 1996*.
- [14] P. Mehrotra, I. Baldine, D. Stevenson, and P. Franzon, "Network processor design for optical burst switched networks," in *Proc. Intl. ASIC/SOC Conf. 2001*, (IEEE, September 2001).
- [15] I. Baldine, G. Rouskas, H. Perros, and D. Stevenson, "JumpStart: a just-in-time signaling architecture for WDM burst-switched networks," *IEEE Communications*, **40**(2), 82-89 (2002).
- [16] <http://www.ic-arda.org/>
- [17] A. Zaim, I. Baldine, M. Cassada, G. Rouskas, H. Perros, and D. Stevenson, "JumpStart just-in-time signaling protocol: a formal description using extended finite state machines," *Optical Engineering*, **42**(2) (2003).
- [18] I. Baldine, G. Rouskas, H. Perros, and D. Stevenson, "Signaling support for multicast and QoS within the JumpStart WDM burst switching architecture," *Optical Networks* (to appear).
- [19] L. Xu, H. Perros, and G. Rouskas, "A queuing network model of an edge optical burst switching node," in *Proc. Infocom 2003*, (IEEE, San Francisco CA, April 2003).