Modeling and Analysis of Ultra High Capacity Optical Networks

Arnold Bragg
MCNC Research and Development Institute
Advanced Networking Research Division
Box 13910
Research Triangle Park, NC 27709 USA
abragg@anr.mcnc.org

Harry Perros
North Carolina State University
Department of Computer Science
Box 7534
Raleigh, NC 27695 USA
hp@csc.ncsu.edu

Keywords: optical networks, network emulation.

Abstract

Ultra high capacity optical networks that are able to provide bandwidth on demand cannot be modeled with high fidelity using today's methods and tools. Discrete event and hybrid simulators, and (near) real time network emulators capable of 'processing' several hundred thousand packets per second cannot cope with aggregate traffic volumes four or five orders of magnitude larger. Analytical techniques provide first-order steady state approximations, but cannot capture dynamic and transient behaviors.

All relevant information about the network's *data plane* is contained in the signaling messages that transit the network's *control plane*, so it is not necessary to precisely simulate data plane traffic, nor to use parallel or concurrent simulation techniques to synchronize the operation of the data plane's components.

Modeling these networks in (near) real time requires a combination of techniques: emulators to mimic the network's control plane; an inference engine to deduce data plane behavior and performance from traffic observed in the control plane; analytics to model core network behaviors; a fast hybrid simulator to inject traffic and network impairments; and a supervisory kernel to interconnect and manage the components. This approach allows one to model ultra high capacity networks at full operational load in (near) real time with very high fidelity.

1. ULTRA HIGH CAPACITY NETWORKS

Many optical networks have link rates of 2.5 Gigabits per second, and research teams are deploying 10 and 40 Gbit/sec optical links in experimental network test beds (*e.g.,* the Internet2 Abilene backbone, and the NSF's TeraGrid) [Dr00]. If the optical links are conditioned to carry multiple channels via wavelength division multiplexing, then a network's capacity can exceed tens of Petabits per second.

An important feature in some ultra high capacity networks is the ability to provision wavelength connections between endpoints ("light paths"). This can done in a number of ways (*e.g.,* IETF GMPLS, and the User-Controlled Lightpath Provisioning protocol [Wu03]). Several experimental architectures are capable of provisioning end-to-end light paths in a few milliseconds [*e.g.,* Ch99, Ch00, Ve00, Ya00, Du02, Ma02, Ra02a, Vu02, Yu02a], and several applications actually utilize user- or application-initiated light path provisioning. *E.g.,* the high energy physics community uses "lambda grids" for Terabyte file transfers; others are experimenting with visualization applications in which data is speculatively pre-fetched using a set of ultra high bandwidth "Terapipes", and stored in close proximity to endpoints so that the latency seen by the application is quite small. Many believe user- and application-initiated provisioning applications will migrate to carrier and ISP optical networks within five years.

The Global Grid Forum has identified basic requirements for applications and services that use optical network resources; these include: **(1)** a scalable, flexible, and rapidly reconfigurable optical network infrastructure; **(2)** ultra high bandwidth on demand between arbitrary endpoints; and **(3)** user/application provisioning and control of bandwidth with sub-wavelength granularity [GRID].

Provisionable optical networks have been identified by the U.S. National Science Foundation and various Federal agencies as an essential and critical part of the Nation's information infrastructure. Unfortunately, large-scale, ultra high capacity optical networks cannot be realistically modeled and analyzed using today's methods and tools. A new approach is required to provide insight into how applications and services that rapidly provision and release network resources might behave in these networks.

2. LIMITATIONS OF TODAY'S METHODS

Networks with Petabit per second capacities (and billions of packets in transit) supporting services that opportunistically 'scavenge' network bandwidth cannot be realistically *analyzed* with today's performance tools. In some networks, the behavior of packets transiting a single ultra high capacity optical link cannot even be *measured* – optical monitoring equipment is prohibitively expensive, there may be tens of channels sharing the link, and the sheer volume of data crossing the link can incur enormous sampling, storage and analysis costs.

Furthermore, most ultra high capacity networks cannot be realistically *modeled* with today's methods and tools; *viz*., discrete event simulation, analytical techniques, hybrid techniques, and network emulation.

(1) *Discrete event simulation* is widely used to analyze the performance of low capacity communications networks, but is severely taxed when applied to higher capacity networks. Discrete event simulators typically process 10 to 100 thousand events per second per simulation instance, and a single packet transfer can spawn tens to hundreds of events (depending on the granularity of the simulation). A simulator's resources are usually exhausted when its event list exceeds several million entries. Sophisticated parallel and distributed simulation techniques can extend the total event processing rate to several tens of millions of events per second at high fidelity [Fu90, Fu00], but are unlikely to scale to the traffic volume found in ultra high capacity networks.

(2) *Analytical techniques* have been successfully used to model networks at an abstract level. They typically describe steady-state network behavior using exact or approximate mathematical formulae derived from queueing theory, Markov processes, and numerous extensions [*e.g.,* Ja57, Ba75, Ge76, Ke79, Br80, To80, Ch83, Di83], and/or from operational analysis and its extensions [*e.g.,* Bu76, De78, Bu82]. These techniques provide coarse first-order approximations of network performance without the run time and memory requirements and the scalability ceiling of discrete event simulators.

However, analytical techniques cannot model a dynamic system in great detail, nor can they model feedback-based algorithms. Ultra high capacity networks are extremely difficult to model analytically due to behaviors induced by: widely disparate traffic sources; closed loop controls (*e.g.,* TCP rate controls, resource provisioning); network elements (*e.g.,* routers with active queue management schemes, switches with preemptive blocking); cross-layer protocol effects (*e.g.,* wavelength routing *and* IP routing at different layers of the same network); asymmetries in bandwidth requirements and session lengths, *etc*. The confounding effects caused by interactions among these factors can also have a profound impact on the dynamics of the network.

(3) *Hybrid techniques* use discrete event simulation to model parts of the network in great detail. The remaining sub-systems are represented by analytic expressions that have been derived by studying these sub-systems in isolation. Crafting a hybrid model requires expert knowledge, careful tuning, and a precise balance between model fidelity and the resources available to the hybrid modeling tool (*e.g.,* CPU cycles, simulator memory, simulation run time) [*e.g.,* Je95, Lu97, Bo02, Ko02].

(4) *Network emulation* is an effective technique for some types of networks, especially those using IP as the network layer protocol. Emulators run (nearly) unmodified protocol stacks on a few tens to a few thousands of 'instances' – commodity PCs, or processes executing on multi-tasking hosts, or both – in (near) real time. The instances are interconnected over a virtual network core that routes traffic, simulates network bandwidth and other resources, and models congestion, latency, loss, and error [*e.g.,* Ri97, Ri98, Fa99, NSF02, Va02, Wh02, Yo02]. Some implementations emulate transmission of several hundred thousand packets per second per instance. It is not clear whether these data-intensive emulators might realistically scale to the aggregate volume of traffic found in ultra high capacity optical networks (which may be four or five orders of magnitude larger).

3. APPROACH

3.1 Overview

Ultra high capacity optical networks typically use wavelength division multiplexing to subdivide an optical link into some number of wavelength channels. The number (eight to hundreds) depends on channel spacing, physical characteristics of the optical fiber, transmitter and receiver complexity, cost, *etc*.

A wavelength channel may be assigned to a single user or application. However, assigned circuits are inefficient for traffic that has not been carefully groomed or statistically multiplexed. If the sustained traffic volume does not require a full 2.5, 10, or 40 Gbit/sec wavelength channel, then the channel may be sub-divided and shared in some way – by time slicing (*e.g.,* time division multiplexing), by destination (*e.g.,* an MPLS-like tunnel), by 'burst' (*e.g.,* optical burst switching), *etc*.

Most ultra high capacity optical networks have two components: a large number of optical channels for data traffic (the *data plane*) and a very small number of channels reserved for control traffic (the *control plane*). In some architectures, the data plane payloads are transparent to the network elements; *i.e.,* data transits the network without optical-electrical-optical (OEO) conversion at intermediate nodes, so a data channel can carry traffic in any format or encoding scheme, and at any rate. In these networks, data is not buffered at intermediate nodes, so data transmissions will experience blocking in the core of the optical network if resources are not available.

The control plane is used to carry signaling messages that configure network elements (switches, routers, add-drop multiplexers) at the network nodes, and that convey routing and resource scheduling information. The control plane is also responsible for conveying network management messages, alarms, failure indications, *etc*. In general, control messages undergo OEO conversion and processing at each intermediate network node. If the control channel is congested, control messages are buffered and may incur queueing delay or loss due to buffer overflow.

A data plane light path may be configured in a number of ways: as a provisioned circuit dedicated to a specific application, as a tunnel that is shared by a number of applications with the same destination, or as a short-lived 'on-the-fly' conduit that may carry only a few tens of packets in its lifetime. Data plane light paths can be configured to support multicast, quality of service, prioritization, and preemption. They can be provisioned in tens of microseconds to a few milliseconds (in some architectures), and can have lifetimes from a few tens of milliseconds to months. Once a data plane light path is provisioned, the control channel carries relatively little information about that light path.

Figure 1 shows one variation. A data transmission is 'announced' by a **SETUP** message on the control channel. The **SETUP** message informs each switch along the path so that switch configuration and routing decisions can be made prior to the data's arrival. When a **SETUP ACK**nowledgment is received, the source begins transmitting data. Long transmissions may require **KEEPALIVE** messages to maintain state since the data is transparent to the switches along the path. Resources may be implicitly released if the data transmission's duration is known (and is conveyed with the **SETUP** message), or explicitly **RELEASE**d as in Figure 1.

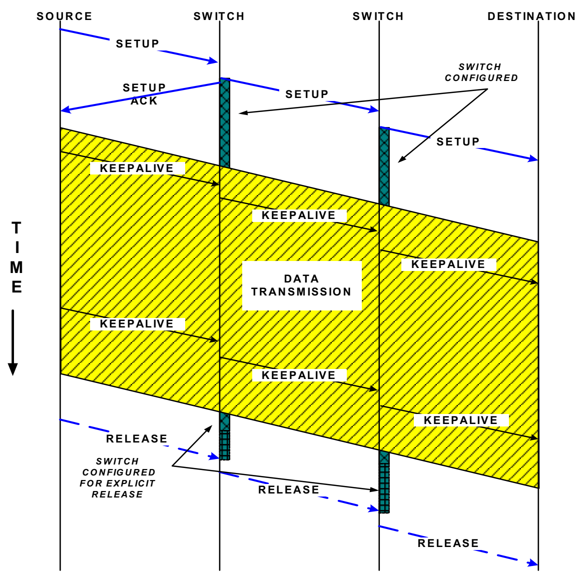

Figure 1. Optical data transmission with **SETUP**, **KEEPALIVE**, and **RELEASE** control messages

There are many architectural variations – wavelength routed, wavelength switched, hybrid wavelength routed/switched, optical burst switched, optical flow switched, sub-rate multiplexed, optical packet switched, *etc*. Many of these schemes experimental, with the exception of wavelength-routed networks that have been commercially deployed and burst switched networks that are deployed in network test beds. Each differs in how it reserves, schedules and releases resources, in its signaling architecture, and in the way information is transmitted in the data plane [Qi99, Yo99, Qi00, Tu00, Ya00, Xu01].

In most of these ultra high capacity optical network architectures, nearly all of the *relevant* information about the data plane is contained in the signaling messages that transit the control plane. This suggests that it is not necessary to model the data plane with detailed discrete event or hybrid simulation techniques, nor to use parallel or concurrent simulation methods to synchronize the operation of the data plane's components.

3.2 Reference Model

Our reference model for modeling and analyzing ultra high capacity networks has five components (Figure 2):

(1) A set of network *emulators* to model the network's control plane (connection management, resource scheduling, routing and forwarding, restoration, alarms, and related services). The control plane of an N-node optical network runs as N independent protocol stack instances on some number of commodity hardware devices. ($N = 12$ in Figure 2.)

(2) A distributed *inference engine* to deduce the state, behavior, and performance of the data plane's links, channels, and devices based on information gleaned from traffic that it observes transiting the emulated control plane.

(3) An *analytical model* to represent the behavior of the core network elements (light path/flow/burst blocking probabilities, resource allocation, *etc*.).

(4) A thin, fast, *hybrid simulator* to model topological details, and to inject traffic, network impairments and other dynamic and transient events based on actual traffic traces and/or stochastic models.

(5) A *supervisory kernel* to interconnect and manage the components. All communication between components is coordinated by the kernel.

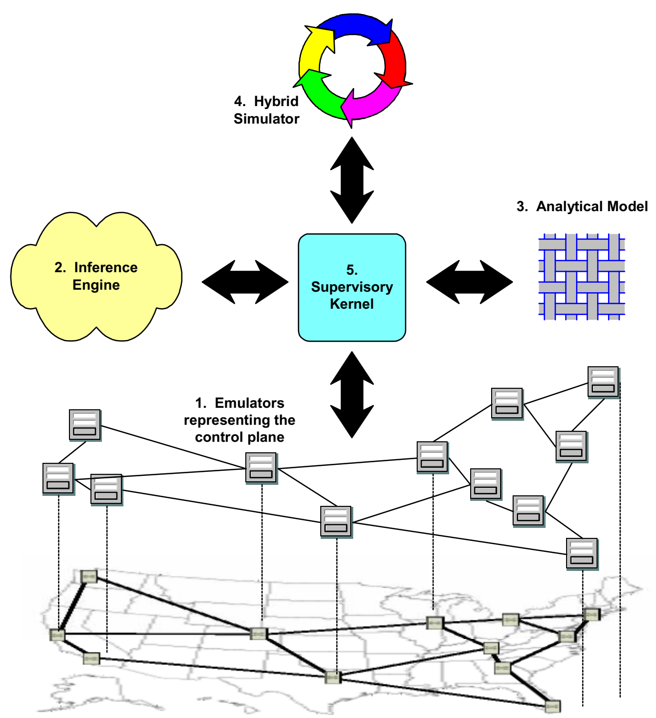

Figure 2. Reference model showing the five major components

3.3 Emulator

We emulate the control plane of an *N*-node optical network by running the control plane's unmodified protocol stacks either on *N* commodity hardware devices, or as *N* concurrent tasks on a commodity multitasking platform, or in some combination of devices and tasks. (See [Va02 and Yo02] for an implementation that uses commodity devices to emulate edge nodes in an electronic wide area network with Megabit per second links and no wavelength division multiplexing.)

Emulating the control plane for circuit-switched optical networks is straightforward. Once provisioned, circuits tend to be long-lived. Emulating wavelength division multiplexed/routed networks and burst-switched networks is somewhat more challenging (and far more interesting) because transmissions and light path holding times can range from a few milliseconds to months, and because each transmission generates signaling and control message traffic that completely describes the data transmission [Ba02, Ba03, Za02, Za03].

The *N* emulator instances are virtually and/or physically interconnected to mimic the topology of an arbitrary ultra high capacity wide area network (including transmission and processing delays, and distance-related propagation delays). The emulators are capable of reproducing the control plane behavior of circuit switched, hard provisioned, soft-provisioned, wavelength-routed, burst-switched, and closely related optical network architectures.

A single control plane instance is capable of emulating about 40 *thousand* control tasks per second, which we believe is comparable to between 40 and 80 *million* discrete event simulation events per second per instance. The emulators operate concurrently and in parallel. Hence, an *N*-instance emulator test bed can process between 40*N* and 80*N* million equivalent control plane events per second. As noted, it is only necessary to emulate the control traffic on the optical network, which is a very small fraction of the total traffic on the network. In other words, one control channel event can describe hundreds to tens of thousands of data channel events.

3.4 Distributed Inference Engine

The distributed inference engine is used to deduce the state of the network's data plane (channels and devices) and to calculate performance measures from the signaling messages that it observes in the network's control plane. As noted, the data plane can be represented as a simple abstraction containing minimal information because core network data plane devices have no buffers or queues; some are little more than passive optical switching devices.

Signaling messages transiting the control plane contain the following types of information:

- Addressing – source, destination or destinations (if multicast), port numbers, session numbers, *etc*.
- Data transmission – beginning and ending timestamps, unicast/multicast, implicit or explicit release of resources, single or multiple transmissions, multicast scope and join details, *etc*.
- Payload encoding – analog or digital, format, modulation, rate, *etc*.
- Routing and forwarding – label allocation, forward and reverse paths, 'pinned' or on-the-fly routes, *etc*.
- Data plane quality of service requirements – bandwidth, priority, preemption, *etc*.
- Optical channel characteristics – channel description, frequency, signal-to-noise ratio, bit error rate, dynamic range, wavelength conversion capability, *etc*.
- Exceptions – failure or early release cause, alarms, *etc*.
- Network management – recovery, restoration, OAM, network topology, routing and forwarding tables, *etc*.

3.5 Analytical Model

The inference engine takes the control traffic generated by the emulators as input, and uses algorithms and heuristics to predict a number of performance measures.

Performance measures include blocking and preemption events on the *data* channels; queueing delay, delay variation (jitter), loss, and error events on the *control* channels; data channel characteristics represented as transmission counts, transmission durations, transmission arrival and interarrival processes and distributions; short- and long-range dependencies in each realm; and confounding effects arising from interactions between the data and control plane realms.

The inference engine also uses topological information (nodes, links, faults) gleaned from forwarding and routing messages carried on the control channels, and information about network impairments and other events artificially injected into the simulation.

3.6 Hybrid Simulator

A thin, fast, hybrid simulator is used: **(1)** to represent and maintain the state information for the arbitrary network topology (nodes, links, bandwidth, delay and error characteristics, *etc*.); **(2)** to generate and inject data plane traffic, and to model traffic flows and behaviors; **(3)** to generate and inject impairments and other stochastic network conditions; and **(4)** to maintain a small amount of additional global information.

The traffic generators are based on results from a large body of recent work on traffic in high performance networks; *e.g.,* [Do00, Ge00, Mo01, Iz02, La02, Sh02, Xu02, Yu02]. It is not clear whether observations about the performance of other signaling networks (*e.g.,* the PSTN's SS7 overlay) might apply to control traffic in ultra high capacity optical networks.

Network impairments affect loss and error rates in optical networks, and arise from a number of phenomena:

crosstalk, attenuation, dispersion, power transients, self-phase and cross-phase modulation, four-wave mixing, scattering effects, and other nonlinear effects that accumulate over long optical paths. Results from recent work will be used to simulate losses, errors, and other faults in ultra high capacity optical networks; *e.g.,* [An00, Ke00].

3.7 Supervisory Kernel

The supervisory kernel is the 'glue' that interconnects and manages the components, and coordinates their interworking. The emulators and the inference engine are tightly coupled – the inference engine distills the control plane traffic into a concise set of state, behavior and performance attributes. Both components respond to various network- and traffic-related events and interrupts injected by the analytical model and the hybrid simulator components.

The kernel consists of a small event list, a message passing and processing module, master scheduling and clock modules, a resource manager, an interrupt handler, and modules to support a global shared-memory abstraction.

4. VALIDATION

We shall validate the approach using applications running in the ATD*net* all-optical network test bed [ATDN]. These include Petabyte file transfers using gridFTP and an enhancement of the Scheduled Transfer Protocol (**`ANSI INCITS 337-2000`**) running on high performance SGI hosts, and immersive real time visualization of latency-sensitive satellite imagery with 5-meter granularity. Results from the simulator will be compared with performance results from the ATD*net* tested.

5. CONCLUSIONS

This approach has a number of novelties and advantages for modeling user- and application-provisionable ultra high capacity optical networks. (As noted, many believe that applications requiring user- and application-initiated provisioning will migrate to carrier and ISP optical networks within five years.) *Viz.*:

- Simulations run at (nearly) the same rate as in the modeled network.
- It provides insight into very complex problems in these networks. Optical networks with dynamic resource provisioning are impossible to analyze by direct measurements, and discrete event simulations on very small subsets may require days of computational time for each minute of simulated time.
- It models services on optical networks; *viz*., scalable, flexible, and reconfigurable provisioning and control of bandwidth on demand between arbitrary endpoints in full and sub-wavelength increments. Circuit switched and wavelength-routed optical networks are commercially deployed today, and researchers are deploying hybrid wavelength routed/switched, burst, and sub-rate optical multiplexed networks in test beds.
- It supports research and experimentation in the design, implementation, and testing of new routing and control protocols and services for dynamic optical networks.
- It delivers very high fidelity by executing (nearly) unmodified control plane protocols for resource provisioning, routing, and network management.
- It captures the behaviors of the most influential components – resource allocation and blocking, cross (transit) traffic, delay, delay variation, loss, errors, and congestion and routing effects.
- It is scalable over a range of optical network topologies, diameters, numbers of nodes, and control plane technologies (*e.g.,* GMPLS, UCLP, JIT [Ba02]). It does not require the tool's components to be modified to accommodate larger problems. It does not require a massive global event list or precise synchronization of events (unlike discrete event simulators with parallel and distributed extensions).
- It can be implemented as one control plane instance per commodity computer, as multiple processes on a multitasking platform, or any combination of these.
- It supports arbitrary network topologies specified by the analyst.
- It generates results that are 100% reproducible, which is important to diagnostic analyses that focus on influential factors and parameters, the sensitivity of factors and parameters, interactions among factors, and the confounding effects of those interactions.
- It supports injection of traffic, impairments, and other events and behaviors collected from real networks.
- It is low cost. Commodity computers are inexpensive, and a typical ultra high capacity optical network may only require a dozen or so control plane emulator instances.

We believe this approach will allow analysts to model data plane traffic in volumes three to five orders of magnitude larger than possible with today's tools, and with very high fidelity.

6. REFERENCES

[An00] Anderson W. T. *et al*. The MONET project - a final report. *Journal of Lightwave Technology*, 18(12):1988-2009, December 2000.

[ATDN] Advanced Technology Demonstration Network (ATDnet), http://www.atd.net/ .

[Ba02] Baldine I., 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).

[Ba03] Baldine I., G. Rouskas, H. Perros, and D. Stevenson, "Signaling support for multicast and QoS within the JumpStart WDM burst switching architecture", *Optical Networks* (in press).

[Ba75] Baskett F. *et al.*, "Open, closed and mixed networks of queues with different classes of customers", *J. ACM*, 22:248-260, 1975.

[Bo02] Bolla R. *et al.*, “Hybrid analytical/simulation mechanism for access control and on-line bandwidth optimization in diffserv environments”, *Proc. PlanetIP and Nebula Joint Workshop*, Italy, 2002.

[Br80] Bruell S., *Computational Algorithms for Closed Queueing Networks*, Elsevier, 1980.

[Bu76] Buzen J. “Fundamental Operational Laws of Computer System Performance,” *ACTA Informatica* 7,2, 1976.

[Bu82] Buzen J., “Operational Derivations of some Basic Results in Queuing Theory”, *Proc. Int. CMG Conference*, pp. 356-361, 1982.

[Ch00] Chaskar H. *et al.* “A framework to support IP over WDM using optical burst switching.” *Proc. Optical Networks Workshop*. 2000. 10 pp.

[Ch83] Chandy K. and A. Martin, “A Characterization of Product-Form Queuing Networks”, *J. ACM*, 30 no.2, p.286-299, April 1983.

[Ch99] Chen Y. and J. Turner. “WDM burst switching for petabit capacity routers.” *Proc. IEEE Milcom 1999*. 6 pp.

[De78] Denning P., and J. Buzen, “The Operational Analysis of Queueing Network Models”, *ACM Computing Surveys*, Vol. 10, No. 3, Sept. 1978, pp. 225-261.

[Di83] Disney R. and D. Konig, “Queueing Networks: a Survey of their Random Processes”, *SIAM Review*, 27:335-403, 1985.

[Do00] Dolzer K. *et al.* “A simulation study on traffic aggregation in multi-service networks.” *Proc. IEEE Conf. High Performance Switching and Routing (ATM 2000).* 9 pp.

[Dr00] Dries J. C. *et al.*, “Demonstration of a 40 Gb/s, 4 Channel Optical Receiver Array”, *Proc. LEOS’00*, 344-345 (2000).

[Du02] Duser Michael, *et al.*, “Analysis of a dynamically wavelength-routed optical burst switched network architecture”, *IEEE Journal of Lightwave Technology*, Vol. 20, no. 4, April 2002.

[Fa99] Fall K., “Network Emulation in the Vint/NS Simulator,” *Proc. Fourth IEEE Symp. Computers and Communic*., pp. 244-250, July 1999.

[Fu00] Fujimoto R., *Parallel and Distributed Simulation Systems*, Wiley Interscience, New York, 2000.

[Fu90] Fujimoto R., “Parallel discrete event simulation”, *Communic. ACM*, 33(10): 30-53, October 1990.

[Ge 76] Gelenbe E., and R. Muntz, “Probabilistic models of computer systems--Part 1 (exact results)”, *ACTA Informatica* 7, 35 60, 1976.

[Ge00] Ge A. *et al.* “On optical burst switching and self-similar traffic.” *IEEE Communic. Letters*. March 2000. 3 pp.

[GRID] Simeonidou D. (ed.), “Optical Network Infrastructure for Grid”, Grid High Performance Networking Research Group, draft-ggf-ghpn-opticalnets-0, September 2003, http://forge.gridforum.org/projects/ghpn-rg/ .

[Iz02] Izal M. and J. Aracil. “On the influence of self-similarity on optical burst switching traffic.” *Proc. IEEE GLOBECOM*. 2002. 5 pp.

[Ja57] Jackson J., “Networks of Waiting Lines”, *Operations Research,* 5:518-521, 1957.

[Je95] Jelenkovic B. and J. Melamed, “Automated TES modeling of compressed video”, *Proc IEEE INFOCOM’95*, pp. 746-752, 1995.

[Ke00] Keiser G. *Optical Fiber Communications*. McGraw-Hill, 2000.

[Ke79] Kelly F., *Reversibility and Stochastic Networks*, Wiley, 1979.

[Ko02] Kovalev M. *et al.*, “Self-Adapting Hybrid Simulation Models of Heterogeneous Telecommunication Networks”, *Information Processing*, Vol. 2, No. 2, 2002, http://www.jip.ru/2002/29.pdf .

[La02] Laevens K. “Traffic characteristics inside optical burst switched networks.” *Proc. SPIE OptiComm 2002*. 12 pp. (Proc. SPIE Vol 4874.)

[Lu97] Lucas M. *et al.*, “(M,P,S)-an efficient background traffic model for wide-area network simulation,” *Proc. IEEE Globecom*, Vol. 3, pp. 1572-1576, 1997.

[Ma02] Masetti F. *et al.* “Design and implementation of a multi-terabit optical burst/packet router prototype”, *Proc. OFC 2002*, Post-deadline paper, 3 pp.

[Mo01] Morato D. *et al.* “On linear prediction of Internet traffic for packet and burst switching networks.” *Proc. Tenth Intl. Conf. Computer Communications and Networks*. 2001. 6 pp.

[NSF02] “Report of NSF Workshop on Network Research Testbeds”, November 2002, http://www-net.cs.umass.edu/testbed_workshop/testbed_workshop_report_final.pdf.

[Qi00] Qiao C. and M. Yoo. “Choices, features and issues in optical burst switching.” *Optical Network Magazine*. April 2000. 9 pp.

[Qi99] Qiao C. and M. Yoo, “Optical burst switching (OBS) – a new paradigm for an optical Internet”, *J. High Speed Networks*, 8(1), 69-84 (1999).

[Ra02a] Ramamirtham J. and J. Turner. “Design of wavelength converting switches for optical burst switching.” *Proc IEEE INFOCOMM. 2002*. 9 pp.

[Ri97] Rizzo L., “Dummynet: a simple approach to the evaluation of network protocols”, *ACM CCR*, January 1997.

[Ri98] Rizzo L. “Dummynet and Forward Error Correction.” *Proc. USENIX Technical Conf*., June, 1998.

[Sh02] Sheeshia S. *et al.* “Supporting Ethernet in optical-burst-switched networks*.” J. Optical Networking*. August 2002. 13 pp.

[To80] Towsley D., “Queueing Network Models with State-Dependent Routing”, *J. ACM*, 27 n.2, p.323-337, April 1980.

[Tu00] Turner J. “WDM burst switching for petabit data networks.” *Proc. OFC Conf. 2000*. 3 pp.

[Va02] Vahdat A. *et al.*, “Scalability and accuracy in a large-scale network emulator”, *Proc. 5th Symp. Operating Systems Design and Implementation*, December 2002.

[Ve00] Verma S. *et al.* “Optical burst switching: a viable solution for terabit IP backbone.” *IEEE Network Magazine*. November 2000. 6 pp.

[Vu02] Vu H. and M. Zukerman. “Blocking probability for priority classes in optical burst switching networks.” *IEEE Communic. Letters*. May 2002. 3 pp.

[Wh02] White *et al.*, “An integrated experimental environment for distributed systems and networks”, *Proc. 5th Symp. Operating Systems Design and Implementation*, 2002.

[Wu03] Wu J. *et al.*, “User-managed end-to-end light path provisioning over CA*net 4”, *Proc. NFOEC 2003*, Orlando, FL, pp. 275-282.

[Xu01] Xu L., H. Perros, and G. Rouskas, “Techniques for optical packet switching and optical burst switching”, *IEEE Communications*, 39(1), 136-142 (2001).

[Xu02] Xue F. *et al.* “The performance improvement in optical packet-switched networks by traffic shaping of self-similar traffic.” *Proc. OFC Conf. 2002*. 3 pp.

[Ya00] Yao S., S. Dixit, and B. Mukherjee, “Advances in photonic packet switching: an overview”, *IEEE Communic*., 38(2), 84-94 (2000).

[Yo02] Yocum K. *et al.* “Modelnet: scalability and accuracy in a large-scale network emulator”. *Proc ACM SIGCOMM*. August 2002.

[Yo99] Yoo M. and C. Qiao “A novel switching paradigm for bufferless WDM networks.” *Proc. OFC Conf*. 1999. 3 pp.

[Yu02] Yu X. *et al.* “A study of traffic statistics of assembled burst traffic in optical burst switched networks.” *Proc. SPIE OptiComm. 2002*. 13 pp. (Proc. SPIE Vol 4874.)

[Yu02a] Yu X. *et al.* “Performance evaluation of optical burst switching with assembled burst traffic input.” *Proc. IEEE GLOBECOM. 2002*. 5 pp.

[Za02] Zaim A., *et al.* “Formal description of the JumpStart just-in-time signaling protocol using EFSM.” *Proc. OptiComm 2002*. July 2002. Boston MA. pp. 160-173. (Proc. SPIE Vol 4874.)

[Za03] Zaim A., *et al.*, “JumpStart just-in-time signaling protocol: a formal description using extended finite state machines”, *Optical Engineering*, 42(2) (2003).