Chapter III

Object-Oriented Modeling and Simulation of Optical Burst Switched Mesh Networks

Joel J. P. C. Rodrigues, University of Beira Interior, Portugal
Mário M. Freire, University of Beira Interior, Portugal

Abstract

This chapter presents an object-oriented approach for the development of an optical burst switching (OBS) simulator, called OBSim, built in Java. Optical burst switching (OBS) has been proposed to overcome the technical limitations of optical packet switching and optical circuit switching. Due to the high costs of an OBS network infrastructure and a significant number of unanswered questions regarding OBS technology, simulators are a good choice for simulation and estimation of the performance of this kind of networks. OBSim allows the simulation and evaluation of the performance of IP over OBS mesh networks. A detailed description of the design, implementation and validation of this simulation tool is presented.
Optical burst switching (OBS) (Baldine et al., 2002; Qiao & Yoo, 1999; Turner, 1999; Wei & McFarland, 2000) is a technical compromise between circuit switching (wavelength routing) and optical packet switching (Murthy & Gurusamy, 2002). OBS has been proposed to overcome the technical limitations of optical packet switching, namely the lack of optical random access memory, and to the problems with synchronization since it does not require optical buffering or packet-level processing, and it is more efficient than circuit switching if the traffic volume does not require a full wavelength channel. In OBS networks, IP datagrams are assembled into very large-sized packets called data bursts. These bursts are transmitted after a burst header packet (set-up message), with a delay of some offset time. Each burst header packet contains routing and scheduling information and is processed at the electronic level before the arrival of the corresponding data burst. OBS has some special characteristics (Xu, 2002), such as: i) Granularity: the size of a transmission unit in OBS is between OCS and OPS; ii) Data and control separation: control information is transmitted in a separate channel; iii) Unidirectional reservation: resources are reserved using a unidirectional messaging system (assuming one-way reservation scheme); iv) Burst with variable size: the size of each burst may not be fixed (Qiao & Yoo, 2000); v) No buffering of data: once the data is sent, it must reach destination only with the delay inherent to the medium – the propagation delay of the signal in the optical fiber (assuming that no limited buffering is used; e.g., fiber delay lines). More details about OBS and a comparison with other optical switching paradigms may be found in Rodrigues et al. (2005).

OBS technology raises a number of significant questions related with the analysis of the performance of different resource reservation protocols. Several network parameters may be taken into account; namely, the network size, the network topology, the number of channels per link, the number of edge nodes per core, the edge to core node delay, the propagation delay between core nodes, the burst offset length, the processing time of setup messages and the optical switch configuration time. These OBS parameters may have a significant impact on the network performance. Therefore, these questions may be answered with a toll that simulates the behavior of an OBS network, given the inexistence of such networks in the real-world, although there are some testbeds, as reported in Baldine et al. (2005), Baldine et al. (2003), Baldine et al. (2002) and McAdams et al. (1994).

Previous works about optical networks simulators are based on packet traffic (e.g., IP networks), which is significantly different from the bursty traffic in an OBS network, since bursts are transmitted through the OBS network in a transparent way, in the sense that the network does not recognizes neither the end of burst nor its content. Therefore, new tools are needed in order to include the specific features of OBS traffic at the network layer. This chapter presents a proposal of an object-oriented approach for the development of an OBS simulator, called OBSim, built in Java. OBSim supports studies to evaluate the performance of the following resource reservation protocols: JIT (Wei & McFarland, 2000), JumpStart (Baldine et al., 2005; Baldine et al., 2002; Baldine et al., 2003; Zaim et al., 2003), JIT+, (Teng & Rouskas, 2005), JET (Qiao & Yoo, 1999), and Horizon (Turner, 1999).

This simulation tool is designed to implement a model of OBS networks based on objects and it was programmed in an object-oriented programming (OOP) built model, with the following objectives: i) to compare the performance of different resource reservation protocols...