A Reconfigurable Hardware Tool for High Speed Network Simulation

Cyril Labbé[1], Serge Martin[2], Frédéric Reblewski[1], and Jean-Marc Vincent[3]

[1] M2000, 4 rue R. Razel , 91400 Saclay, France
cyril.labbe@cnet.francetelecom.fr
[2] France Telecom/CNET/DTL/ASR , BP98, Chemin du vieux chene , 38243 Meylan Cedex, France
serge.martin@cnet.francetelecom.fr
[3] Laboratoire LMC-IMAG , Domaine Universitaire , BP53X 38041 Grenoble Cedex 9, France
Jean-Marc.Vincent@imag.fr

Abstract. Estimation of rare events probabilities (such as loss rate) in high speed network remains in most cases an open problem. To address this problem, a flexible hardware testbed for simulation of ATM-based networks has been used. The goal of this article is to present this simulation technique. It is shown that this technique can be used to highlight rare events, such as realistic packet loss probability in high-speed networks.

1 Introduction

More and more High Speed Networks are intended to provide a variety of different services on a single "universal" network. Such services can have widely differing Quality of Service (QoS) requirements. At the packet (cell) level, this means differences in permissible cell loss and cell transfer delays. This measure of performance depends directly on the switch architecture and algorithms for congestion control and scheduling. That is why investigation on performance evaluation are so important.

Models used for this research are often discrete time queuing networks. This is especially true in the case of ATM (Asynchronous Transfer Mode), where slotted time is natural since all the cells have the same size. A slot is the time needed to serve a cell. Because of the small size of the ATM cell and the high link-speeds, a large number of cell events may need to be simulated to ensure satisfactory confidence intervals. A realistic packet loss probability is around 10^{-8}-10^{-9}. Such losses are rare events which are difficult to capture. Software Simulators are too limited to obtain such a probability. Although analytical techniques may be used to bound the worst-case performance, [3] these are often inadequate for modeling the switch algorithms at the needed level of detail.

The aim of this paper is to show a new approach, using emulation on a versatile architecture machine for performance evaluation of high speed networks [8, 2]. This technique is used to highlight rare events, such as realistic packet

loss probability. This technique is also used to make performance evaluation on congestion control and scheduling algorithms of an ATM switch developed at the CNET.

Programmable hardware (emulation) is widely used to reproduce the functionalities of a circuit. Emulation is performed by an emulator, which can be seen as an hardware simulator. Its hardware configuration can be modified to model other circuits ; this is an "all purpose hardware emulator" based on a versatile architecture [4].

Here we will focus on the architecture, the use, and the possibilities of this tool. An ATM switch is modeled by a queuing network which is emulated by a dedicated architecture on the versatile machine. The structure of the paper is the following. The versatile architecture and software used are presented in Section 2. Section 3 presents experimental results on a eight-by-eight multistage ATM switch.

2 Hardware architecture and software environment

This section presents the hardware architecture and the software environment used to emulate queuing networks. The software is used to describe a component modeling the queuing network and the hardware simulator emulates this component.

2.1 Architecture

The hardware simulator is the M500 machine from Metasystems [4]. It acts like a giant FPGA (field programmable gate array) on which the circuit to be tested and debugged can be mapped. The emulator is based on a building bloc called PLB (Programmable Logic Bloc), static RAM and VRAM. PLBs provide register and basic logic gates, the static RAMs provide possibilities to map memories described in the netlist. The VRAMs sample all the internal nodes for logic analysis of the signal values.

All this give to the user the effective use of : 500,000 programmable logic gates (connected to each other through a programmable network), 17 Mbytes of memory (single or double port), adjustable clock frequency from 1 to 10 Mhz.

This hardware can be shaped to emulate any digital and synchronous circuit. The description of a chip is given to the Emulator by configuration files. The clock frequency, under normal conditions, is usually close to 1 Mhz. The emulator clock is under user control. All signals and register values are available on the last 7000 clock cycles, which is very useful for debugging.

This machine is from the *first generation* (1995). An up to date machine has at least 20 time more logic gates.

2.2 Software environment

The software flow leads to the files required by the emulator to reproduce the functionalities of a circuit. These functionalities are described in terms of concur-

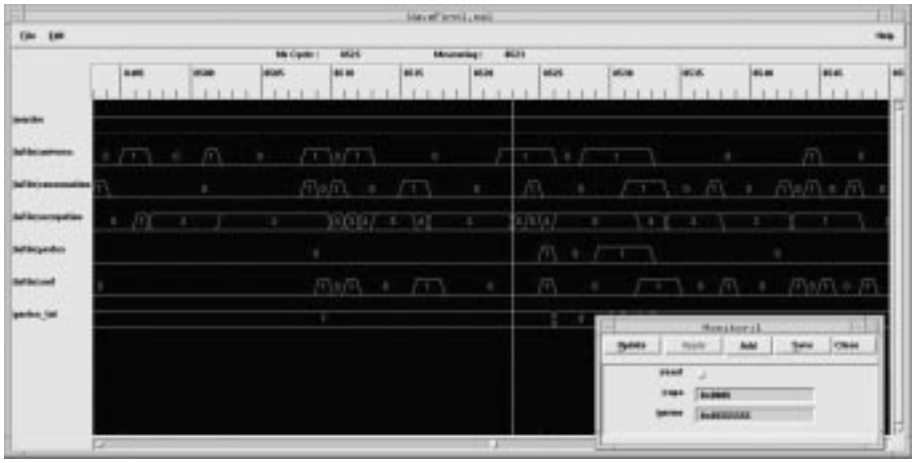

Fig. 1. The waveform window display all signals and register values on the last 7000 clock cycles.

rent processes using the VHDL language. VHDL is an efficient way of obtaining a high level description of a hardware component, which is then translated into gates by the Synopsys synthesis tools. From this representation of the components, the Metasystems compiler produces the data base required by the emulator. The software flow is detailed above :

- a VHDL (VHSIC Hardware Description Language) description of the chip is used to describe the system in terms of concurrent processes [5].
- Synopsys synthesis : this software, provided by Synopsys, translates the VHDL description into combinational logic and registers (logic gates)[5].
- The Metasystems compiler. This is the routing operation, which results in connecting the gates to each other through the programmable network of the emulator.

Those two last steps are entirely automatic.

2.3 Simulation control

Emulation is performed using the MEL tool, which loads the emulator with the configuration file, and allows run control, logic analysis, triggering features, and patterns verification. MEL can be driven by procedures written in a C-like code, which is useful for complex simulation.

All the signals or vectors (busses) can be displayed in a waveform window (cf Figure 1). Control of input signals or registers can be done through the monitor window (cf Figure 1). Any signal and register value can be displayed without recompilation.

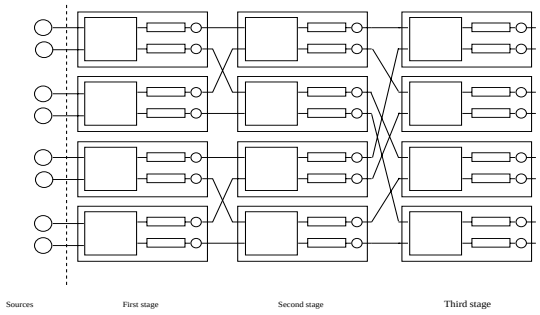

Fig. 2. A three stages eight-by-eight ATM switch modeled with discrete time queues.

3 Application to a three stages eight-by-eight switch

This section is devoted to the study of a eight-by-eight switch (figure 2). The traffic model adopted is geometric, servers of queue are deterministic, with arrival first [1]. This traffic is also call uniform traffic [9, 7]. Figure 3 shows the packet loss probability per stage. The x axis is the queue capacity K varying from 10 to 50. Each curve corresponds to a different stage. The queues of each stage have the same capacities K.

It should be noted that losses are always greater on higher stage. This is explained by the fact that the traffic following a buffer stage is more bursty than the one at the entrance. This is easily observed when doing a statistical analysis of burst length. This has been done thanks to a traffic analyzer which has been build to characterize the traffic perturbation introduce by buffers. Tagged cell can also be used to differentiate background traffic from the point to point communication.

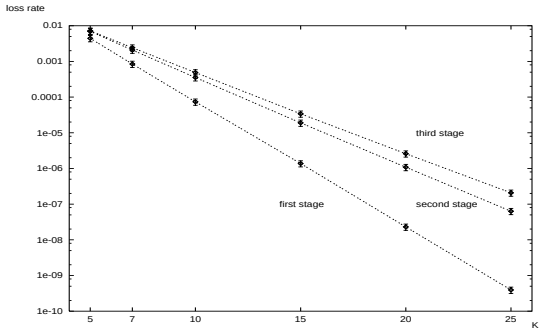

Fig. 3. Loss rate at different stages versus capacities K of queues (same capacities K at each stage), $\rho = 0.8$.

4 Conclusion and extension

In this article, a new technique for simulation of high speed network has been presented. This methodology uses a versatile architecture configured for maximum efficiency for a given problem. Analytical techniques are often inadequate for modeling the commutation algorithms at the needed level of detail. In software simulation, estimation of the probability of rare events are very difficult to obtain. The proposed tools and method overcomes the problem by a parallel approach. In one time slot, the number of treated events is in the order of the number of queues.

This new approach has been applied to the study of rare events in ATM networks. This has allowed simulation of realistic cell loss probabilities $\left(10^{-8}, 10^{-9}\right)$ in a multistage ATM switch. This technology could be used to highlight other rare events with a good degree of accuracy.

This model has been extended to real service policies. In particular for studies on Fair Queuing disciplines and congestion control algorithms. More generally, this type of machine could be used to emulate numerous types of performance evaluation problems using discrete time queuing network, graphs or Petri nets.

References

1. A.Gravey and G.Hébuterne. Simultaneity in discrete-time single server queues with Bernouilli inputs. *Performance Evaluation North-Holland*, 14:123–131, 1992.
2. C.Labbé, F.Reblewski, and J-M Vincent. Performance evaluation of high speed network protocols by emulation on a versatile architecture. *RAIRO, Systèmes à événements discrets stochastiques : théorie, application et outils.*, to be published.
3. J.Pellaumail. Majoration des retards dans les réseaux ATM. *Rairo recherche opérationnelle*, 30:51–64, 1996.
4. L.Burgun, F.Reblewski, G.Fenelon, J.Barbier, and O.Lepape. Serial fault emulation. In *Proceedings of the 33rd Design Automation Conference 1996 (DAC 96)*, pages 801–806, Metasystems, France, 1996.
5. R. Airiau, J.-M. Berge, and V. Olive. *Circuit Synthesis with VHDL*. Kluwer Academic Publishers, France Telecom, 1994.
6. S. Robert and J.-Y. Le Boudec. Can self-similar traffic be modeled by markovian processes? *Lecture Notes in Computer Science*, 1044, 1996.
7. R.Y.Awdeh and H.T.Mouftah. Survey of ATM switch architectures. *Lecture Notes in Computer Science*, 27:1567–1613, 1995.
8. D. Stiliadis and A.Varma. A reconfigurable hardware approach to network simulation. *ACM Transaction on Modeling and Computer Simulation*, 7, 1997.
9. L. Truffet. *Méthodes de Calcul de Bornes Stochastiques sur des Modèles de Systèmes et de Réseaux*. PhD thesis, Université Paris VI, 1995.