

Emulation on a Versatile Architecture for discrete time Queuing Network : Application to High Speed Network

C.Labbé
M2000
4 rue R. Razel
91400 Saclay, France
cyril.labbe@cnet.francetelecom.fr

Vincent Olive
France Telecom/CNET/DTL/ASR
BP98, Chemin du vieux chene
38243 Meylan Cedex, France
vincent.olive@cnet.francetelecom.fr

Jean-Marc Vincent
Laboratoire LMC-IMAG
Domaine Universitaire
BP53X 38041 Grenoble Cedex 9, France
Jean-Marc.Vincent@imag.fr

I Introduction

More and more High Speed Networks are intended to provide a variety of different teleservices on a single "universal" network. Moreover, 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. A complete specification of performance in packet switching networks involves two components : packet loss and packet delay. End-to-end delay is an important measure of performance as too much delay is equivalent to loss when considering a real time context. This measure of performance depends directly on the switch architecture. That is why investigation on performance evaluation and dimensioning buffered-switching elements are so important. Furthermore, these research activities have a practical impact on architectures and protocols.

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 quite natural since all the cells have the same size. A slot is the time needed to serve a cell. A realistic packet loss probability is around 10^{-8} - 10^{-9} . Such losses are rare events which are difficult to capture by direct computation. In fact, software Simulators are too limited to obtain such a probability. However it is currently possible to compute analytically realistic theoretical loss probabilities on the first stage of a broadband multiplexer and to obtain bounded results on the second stage (and this only for simple mathematical model often far from reality). However, even for those simple models, the computation of realistic loss probability for subsequent stages is still an open problem.

The aim of this paper is to validate a new approach, using emulation on a versatile architecture machine for per-

formance evaluation of finite capacities queuing networks in discrete time [15, 2]. It is shown that this technique could be used to highlight rare events, such as realistic packet loss probability in high-speed switched networks.

Emulation is a widely used technique in the design flow of chips. Programmable hardware is used to reproduce the functionalities of a circuit. Emulation is performed by an emulator, which can be seen as an hardware simulator. Since its hardware configuration can be modified, at the basic logic gate level to model other circuits ; this is an "all purpose hardware emulator" based on a versatile architecture [6, 7].

Here we will focus on packet loss probability and examine the traffic perturbation induced by stages of multiplexers. An ATM switch is modeled by a queuing network which is emulated by a dedicated architecture on the versatile machine. Experimental protocols are conducted to collect statistics on losses and traffic characteristics.

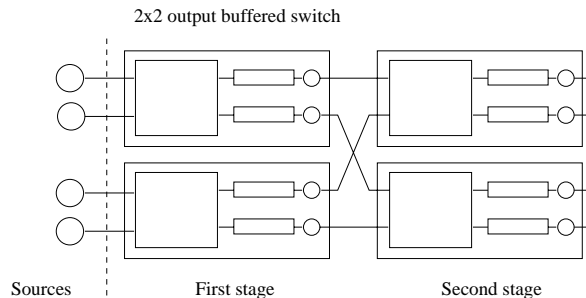


Figure 1: four-by-four double stage switch made of two-by-two output buffered switches.

The structure of the paper is the following. The ver-

satile architecture of the emulator and the software developments used in this study are presented in Section . The design method of a queuing network, using this architecture, is discussed in Section B, using the example of the Geo/Geo/1/k queue which is the equivalent, in discrete time, of a M/M/1/k queue. Section C presents a set of experimental results on a four-by-four multistage ATM switch.

II 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.

A Architecture

The hardware simulator is the M500 machine from Metasystems [6, 7] made with FPGA (Field Programable Gate Array), comprising :

- 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.

All 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. Memories can be loaded (initialized) or dumped using ASCII files. The values of the input signals of the emulated component are directly chosen by the user. The clock frequency is 10 Mhz for a very small occupation of the emulator but, under normal conditions, the frequency 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.

B Software environment

The software flow leads to the files required by the emulator to reproduce the functionalities of a circuit. These fonctionnalities are described in terms of concurrent 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 presented in Figure 2 and is detailed above :

- a VHDL (VHSIC Hardware Description Language) description of the chip is used to describe the system in terms of concurrent processes [9].

- Synopsys synthesis : this software, provided by Synopsys, translates the VHDL description into combinational logic and registers (logic gates), which are the basic logic elements used in electronic systems [10].
- The Metasystems compiler computes the links for the set of gates available in the machine. 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 expect for customizing thr final result with Metasystems memory models.

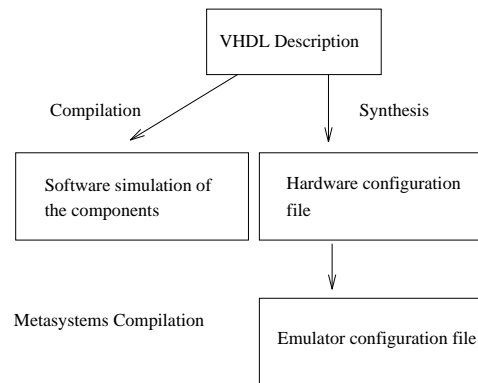


Figure 2: The software flow used to structure the versatile architecture.

III Modeling discrete time queues using VHDL hardware description

In this section, the method for modeling a queue in synthesizable VHDL is described. Different ways of queue building are proposed, showing that the hardware representation depends on the adopted model. The simple case of the Geo/Geo/1/k queue is considered. In the first subsection the updating technique for the number of waiting customers is described. The second subsection presents a random generator technique to produce arrival departure process.

A Hardware queue

A hardware representation of a queue depends on various parameters, such as customer or server characteristics.

For example : if customers are all similar, only a register and some "logic glue" are needed (see Figure 3). As this is a discrete time model, there are two options : arrival first (AF) or departure first (DF) [1]. The register contains the number of waiting customers and is updated according to :

- the number of new customer arrivals,
- the capacity of the queue,
- the number of customers served.

What is important to note is that in this case a slot is equivalent to a clock cycle. This is a poor representation since performance parameters, such as delay or delay variation are not easily measurable. It is however cheap in terms of hardware occupation, since it requires only register and logic glue.

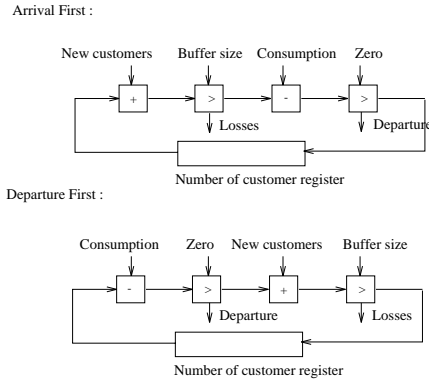


Figure 3: Hardware representation of a queue. This is a discrete time model (AF or DF) where arrival and departure processes are not fixed. In each case only a register and some logic glue is needed.

To introduce information in customers, real packets carrying information can be used. This information has to be queued. This means that each queue of the network is implemented using a memory corresponding in size to the queue capacity. However, in the case of multiple arrivals, multiple information has to be stored in one slot. If packets are in conflict, they all have to be stored in the memory in the same time slot. As a result, a slot has to be decomposed into multiple clock cycles. This increase the simulation time –since one emulated slot requires multiple clock cycles– and the hardware size consumption, since each queue needs memory. This represents, however a considerable extension, as packets can be tagged to take into account priority, address, high level information or time stamps. It is thus possible to study the end-to-end delay and delay variation, or even to study a point to point communication with a background traffic (see section B).

B Random Generator

There are different ways of supplying traffic to the network. It is possible to feed the network with real traffic stored in the memory before beginning the experiment. This is an interesting possibility since the characteristics of real traffic are very hard to reproduce [11]. However, to reach a low loss probability, a very large sample of traffic

is required, which is too large to be stored in the emulator memory. Another solution is to emulate traffic using statistical models.

To reproduce a geometric process (inter-arrival time geometrically distributed) we use a memory-based random generator¹. Every clock cycle a 16 bits random word w is generated ($0 \leq w \leq 2^{16} - 1$). To reproduce the geometric process $geom(\rho)$ this word is compared with :

$$\theta = \rho * 2^{16} - 1$$

For example with $\theta = 52429$ we have $\rho \simeq 0.8$. A cell is emitted if $w \leq \theta$.

C The Geo/Geo/1/k queue

The Geo/Geo/1/k queue is a discrete time queue with a capacity k. This is a queue feed with a geometric arrival process $geom(\lambda)$ and the service time is geometrically distributed with rate μ . So the load of the queue is $\rho = \frac{\lambda}{\mu}$.

Let λ' and μ' be such that :

$$\lambda' = \frac{\lambda}{n}, \quad \mu' = \frac{\mu}{n}$$

Let n grows to ∞ with ρ constant. If $\frac{1}{n}$ is taken as the size of the slot. Increasing n as the same effect as decreasing the size of the slot. The asymptotic model is a M/M/1/k queue (see [1]). This is illustrated by simulation results in figure 4. This figure shows the distribution of the queue unutilisation.

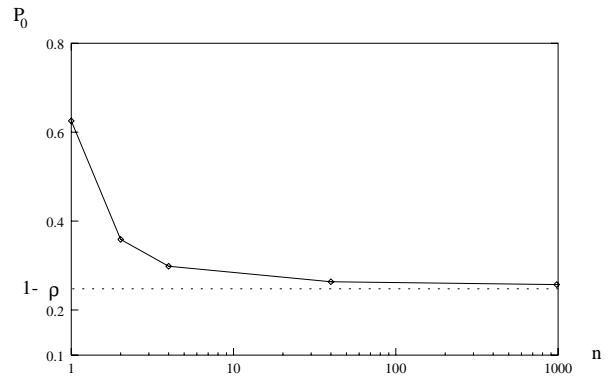


Figure 4: The Geo/Geo/1/k queue tending to the M/M/1/k when discret time is going closed to the continuous model. P_0 is the probability for the queue to be empty.

IV Application to the two stages four-by-four switch

This section is devoted to the study of a four-by-four switch (see figure 1). The first subsection concerns the

¹The random generator is based on the polynomial $1 + X + X^{127}$ (see [16])

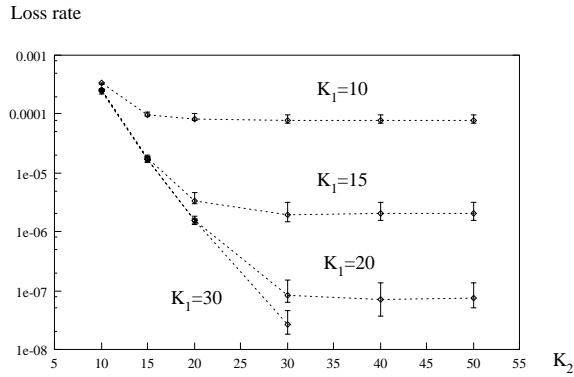


Figure 5: Loss rate versus capacities of the first and second stage, $\rho = 0.8$.

cell loss probability. This leads to the study of the traffic perturbation introduced by the switch. More precisely, the properties of the perturbation introduced by a random background traffic on a periodic traffic are studied in the second subsection. The traffic model adopted is geometric, (see Section B), servers of queue are deterministic. This traffic is also call uniform traffic [17, 14].

A Loss probability function of capacities.

Figure 5 shows the global packet loss probability, which is the loss probability over all the four-by-four switch. The x axis is the second stage queue capacity K_2 varying from 10 to 50. Each curve corresponds to a different size of first stage buffer ($K_1 = 10, 15, 20, 30$).

A plateau is observed on each curve for high values of K_2 . This can be explained by the fact that the second stage is large enough and thus all observed losses arise from the first stage. In this case, increasing the capacity of the second stage will be useless.

On the other hand, for low values of K_2 , the curves are merged. In fact the capacity K_1 of the first stage is large with respect to that of the second stage. As a result, all losses arise from the second stage. In this case increasing the size of the first stage without increasing the capacity of the second will not be useful.

For example, with $K_1 = 20$, the "optimal" configuration, in terms of losses with respect to memory cost, is obtained for $K_2 \simeq 30$.

B Traffic perturbations

In this section the impact of a background traffic on a point to point communication is studied. For the Geo/Geo/1/k queue the model of queue without contents was used (see section B). In this subsection customer contain specific information which concerns the origin and destination of the packet. So, tagged cell are used to differentiate background traffic from the point to point communication.

A particular source is devoted to the periodic traffic and the other sources generate the "background noise". So

one of the four random sources used in section A is replaced by a periodic source of period 4. That is to say every four slot a packet is emitted.

We can see in figure 6 the perturbation introduce by the random traffic on the periodic one. The curve indicates the distribution of the inter-arrival time at the output of the switch for the point to point periodic communication. This perturbation is more important when the load of the traffic is more important. Perturbation increases with the traffic load.

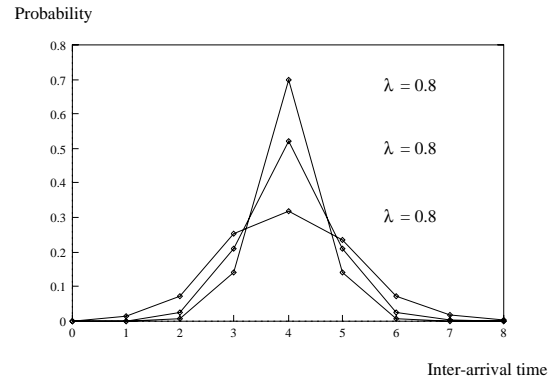


Figure 6: Perturbation introduce by a random traffic on a periodic traffic. λ is the load of the background traffic.

V Conclusion and extension

In this article, a new methodology for simulation of discrete time queuing networks with finite capacities has been presented. This new methodology uses a versatile architecture, configured for maximum efficiency for a given problem. This method has to be compared to software tools which only allow the computation of higher (10^{-5}) cell loss rates. What is important to note is the fact that this architecture is quite easy to configure, and that this configuration is very easy to debug. This last point results from the fact that all states of the configuration are available.

This new approach has been applied to the study of high-speed packet switched networks. This has allowed for simulation of realistic cell loss probabilities ($10^{-8}, 10^{-9}$) at all stages of a multistage ATM switch. Using this technology, it is thus possible to highlight rare events with a high degree of accuracy.

For further studies, the model has to be extended to real service policies. In particular studies on Fair Queuing disciplines.

More generally, this type of machine could be used to emulate numerous types of network protocols, and also to solve different discrete time queuing network problems.

References

- [1] A.Gravey and G.Hébuterne. Simultaneity in discrete-time single server queues with Bernoulli 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] F.Guillemain, G.Rubino, B.Sericola, and A.Simonian. Transient characteristics of an $M/M/\infty$ system applied to statistical multiplexing on an ATM link. Publication interne 874, IRISA, october 1994.
- [4] F. Hübner. Discrete-time analysis of the busy and idle period distributions of a finite-capacity ATM multiplexer with periodic input. *Performance evaluation North-Holland*, 21, 1-2:23–36, 1994.
- [5] J.Pellaumail. Majoration des retards dans les réseaux ATM. *Rairo recherche opérationnelle*, 30:51–64, 1996.
- [6] L.Burgun and F.Reblewski. Première génération d’emulateurs matériels meta-systems. In *Quatrième symposium sur les architectures nouvelles de machines*, Metasystems, France, 1996.
- [7] 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.
- [8] G. Pujolle. Commutateurs ATM: Classification et architecture. *Technique et Science Informatique*, 11, 1:11–29, 1992.
- [9] R. Airiau, J.-M. Berge, and V. Olive. *VHDL du langage à la modélisation*. Presses polytechniques et universitaires romandes, France Telecom, 1990.
- [10] R. Airiau, J.-M. Berge, and V. Olive. *Circuit Synthesis with VHDL*. Kluwer Academic Publishers, France Telecom, 1994.
- [11] S. Robert and J.-Y. Le Boudec. Can self-similar traffic be modeled by markovian processes? *Lecture Notes in Computer Science*, 1044, 1996.
- [12] J. Roberts and F. Guillemin. Jitter in ATM networks and its impact on peak rate enforcement. *Performance Evaluation North-Holland*, 16, 1-3:35–48, 1992.
- [13] Sheldon M. Ross. *A Course in Simulation*. Mamilan Publishing Company, University of california, Berkeley, 1991.
- [14] R.Y.Awdeh and H.T.Mouftah. Survey of ATM switch architectures. *Lecture Notes in Computer Science*, 27:1567–1613, 1995.
- [15] D. Stiliadis and A.Varma. A reconfigurable hardware approach to network simulation. *ACM Transaction on Modeling and Computer Simulation*, 7, 1997.
- [16] Shu Tezuka. *Uniform Random Numberd : Theory and practice*. Kluwer Academic Publishers, IBM Japan, 1995.
- [17] L. Truffet. *Méthodes de Calcul de Bornes Stochastiques sur des Modèles de Systèmes et de Réseaux*. PhD thesis, Université Paris VI, 1995.