

# A Fast Discrete Event Simulation Model for Queueing Network Systems

J. L. Vazquez-Avila, R. Sandoval-Arechiga and R. Parra-Michel  
CINVESTAV-IPN, GDL Unit  
Zapopan, Jalisco, Mexico  
{jl vazquez, rsandoval, rparra}@gdl.cinvestav.mx

## ABSTRACT

Based on Lindley's recursive equations for G/G/1 systems, this paper proposes a Fast Discrete Event Simulation (FDES) model for queueing networks. Equations for multiplexer and de-multiplexer elements are presented, which allows to simulate not only tandem but queueing networks with an arbitrary topology. Time savings obtained with FDES could speed up the analyses of large-scale queueing network systems. Experimental results show that FDES modeling can be two orders of magnitude faster than their counterparts based on the event-scheduling for practical cases.

## Keywords

Discrete-Event, Queueing Network Systems, Fast Simulation.

## 1. INTRODUCTION

Discrete Event Simulation (DES) of queueing network systems with an arbitrary topology is studied in this work. Simulation based analyses have proved their utility in a widely range of applications, from design and evaluation of manufacturing systems and computer networks, to complex stochastic systems [2]. DES obtains performance results where mathematical analyses are restrictive due to its complexity. However, DES's utility may be limited, in turn, by the time spent in the simulation. Therefore, a fast simulation that allows the study of a larger and more complex systems is desirable. Based on Lindley's recursive equation for G/G/1 queueing systems [8], a complete set of equations is proposed. These mathematical statements conforms a fast simulation model that enables the analysis of queueing network systems with an arbitrary topology. Here, this model is referred as *Fast Discrete Event Simulation (FDES)*.

## 2. RELATED WORK

Event scheduling is the main approach used in the literature for the study of DES [2, 3]. Besides, modern discrete

event simulators such as Matlab Simevents [4], and Omet++ [9], employ the event scheduling approach<sup>1</sup>. However, in [3, 7, 5] a simulation approach that turns out faster than event scheduling is reported. Based on Lindley's recursive equation, Chen [3] presents a fast simulation for single-servers tandem queueing systems. Then, Krivulin in [7], formalizes Chen's approach using max-plus algebra. Finally, Kin in [5] generalized this approach to multiple-servers in tandem. Therefore the set of equations for any topology has not been complete. In the following section, this approach is expanded to permit the analysis of queueing network systems with an arbitrary topology.

## 3. FDES MODEL

In this section we present the FDES model for a network of single servers with infinite queue lengths. Besides, a round robin multiplexer and de-multiplexer model are defined, which allows to simulate any network topology.

Table 1: Notation for the Single Server Model.

$E_n$	$n$ th entity
$a_n$	arrival epoch for entity $E_n$
$w_n$	waiting time for entity $E_n$ ; $w_1 = 0$
$b_n$	service time for entity $E_n$
$x_n$	interarrival time; $x_n = a_n - a_{n-1}$
$u_{n-1}$	stability variable; $u_{n-1} = b_{n-1} - x_n$
$v_n$	departure time epoch for entity $E_n$
$y_n$	interdeparture time; $y_n = v_n - v_{n-1}$
$\lambda$	mean interarrival rate
$\mu$	mean service rate

### 3.1 Single Server Model

The model consist on a G/G/1 single server system with an infinite buffer, and operates according to the First-In-First-Out (FIFO) queueing discipline, as is shown in Figure 1. Table 1 presents the notation used for the single server model. The time diagram shown in figure 2, depicts the behavior of the entities as they travel through the queue and the single server from their arrival time  $a_n$  to their departure time  $v_n$ . From figure 2 some important equations are obtained and summarized in Table 2. These equations represent the same results as in [3, 7, 5], but in an alternative representation. Notice that the waiting time is expressed

<sup>1</sup>A comparison between these environments is presented in [1]

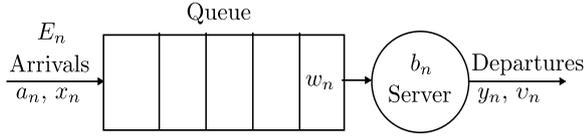


Figure 1: Model of a queueing system.

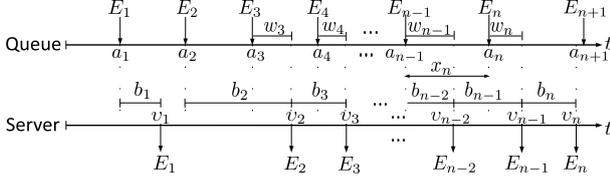


Figure 2: Time diagram for an infinite queueing system.

in terms of the stability variable and results in the Lindley recursion [6, 8], where  $[x]^+$  denotes  $\max(0, x)$ .

### 3.2 Multiplexer and De-multiplexer Models

The main contributions of this paper are the multiplexer and de-multiplexer models. Due to the lack of space, only summarized equations are shown. The multiplexer, shown in figure 3a, consists of  $M$  input ports represented by single servers. Each single server queue behaves as the model presented in the previous section.  $E_{m,k}$  represents the  $k$ th entity (customer) that arrives to port  $m$ . The way in which the entities from each input port are served depends on the arbitration policy. For simplicity (but not restricted to this) we consider a priority round robin arbitration policy.

In the round robin policy used in this work, port 1 is served first, then port 2 and so on. If port  $m$  has the token to transmit and no entity has arrived and port  $m+1$  has entities, then port  $m+1$  can not serve the entities until port  $m$  transmits an entity. Figure 4 shows a time diagram of a multiplexer with 3 input ports operating with a round robin arbitration policy. Table 3 summarizes the equations in the multiplexer system analysis. For all the presented equations  $n = 1, 2, 3, \dots, k = 1, 2, 3, \dots$  and  $1 \leq m \leq M$ . The multiplexed output is represented by  $v_{mux,n}$ , where  $a_{m,k}$ ,  $w_{m,k}$  and  $b_{m,k}$  are the arrival epoch, waiting time and service time, respectively, of the  $k$ th entity  $E_{m,k}$  for the input port  $m$ , according to the round robin policy (see Figure 4). We define the index of the input port as  $m = \text{mod}(n-1, M) + 1$ . The index corresponding to an entity arriving to input port  $m$  is given by  $k = \frac{n+M-m}{M}$ . Besides, expressions for the waiting time for the  $k$ th entity  $E_{m,k}$  in queue  $m$  are also presented in table 3 (with  $w_{1,1} = 0$ ). Notice that when the  $b_{m,k}$  are the same for all  $m$ , we have the case of a multiplexer with a single server. Also, notice that the stability condition for this system is  $\lambda_T < \mu$ , where  $\lambda_T = \lambda_1 + \lambda_2 + \dots + \lambda_M$  and  $\lambda_m$  is the mean arrival rate to port  $m$ .

On the other hand, figure 3b shows a representation for the de-multiplexer model. The de-multiplexer consists of  $M$  output ports.  $v_n$  represents the  $n$ th entity that arrives to the input port. The way in which the entities are served depends

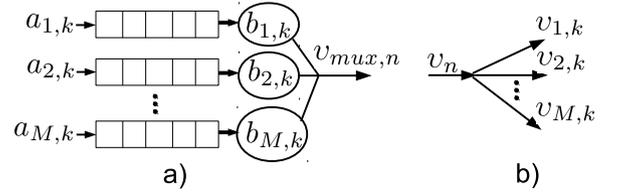


Figure 3: a) Multiplexer model b) De-multiplexer model.

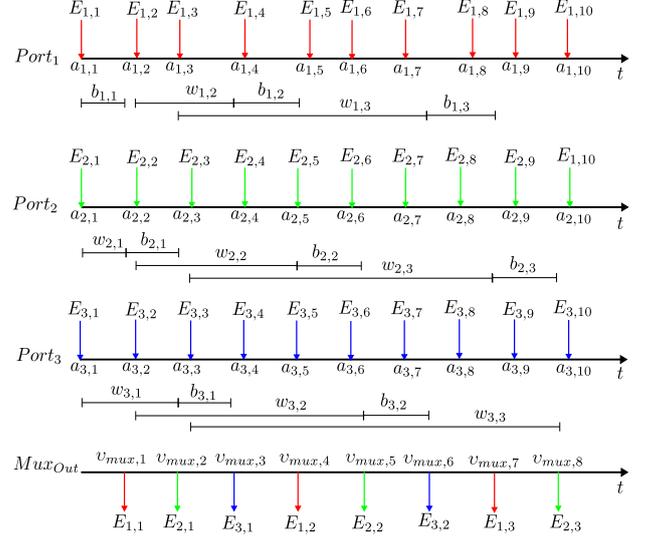


Figure 4: Time diagram of a multiplexer with 3 input ports and operating with a round robin arbitration policy.

Table 2: FDES equations for the Single Server Model.

Waiting time	$w_n = [0, w_{n-1} + u_{n-1}]^+$
Dep. epoch	$v_n = a_n + w_n + b_n$

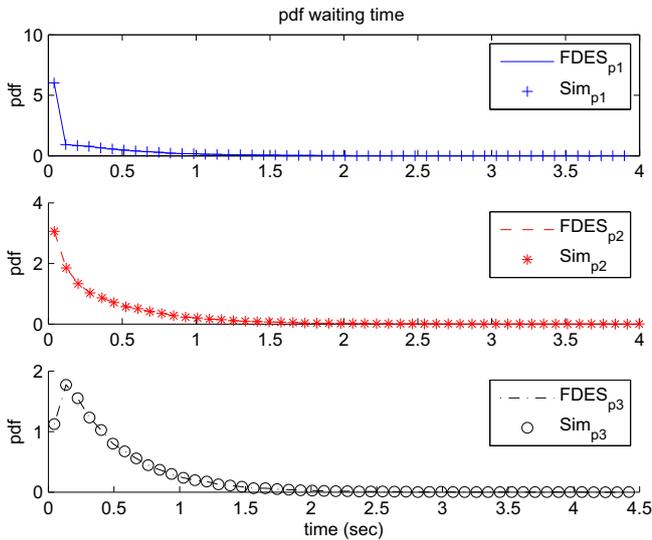
Table 3: FDES equations for the round-robin Multiplexer Model.

Waiting time	$w_{1,k} = [a_{M,k-1} + w_{M,k-1} + b_{M,k-1} - a_{1,k}]^+$
	$w_{m,k} = [a_{m-1,k} + w_{m-1,k} + b_{m-1,k} - a_{m,k}]^+$
Dep. epoch	$v_{mux,n} = a_{m,k} + w_{m,k} + b_{m,k}$

on the arbitration policy and in general on the routing algorithm. Again, we consider a round robin arbitration policy. Notice that in this case the departure epochs for port  $m$  and entity  $k$  is given by  $v_{m,k} = v_n$ , where  $n = M(k-1) + m$ . The complete set of equations presented in table 3 and table 1 confirms the FDES model. As the next section will show, FDES model is expected to outperform event scheduling approaches for the tested cases.

## 4. RESULTS

In this section we present results for the FDES model. First, the probability density functions (pdf's) of the waiting time, for the input queues of the multiplexer model, are presented. A multiplexer with 3 input ports is considered.



**Figure 5: pdf of the waiting time of input queues of the multiplexer utilizing FDES and SimEvents.  $p_1$ ,  $p_2$  and  $p_3$  represent the input ports of the multiplexer.**

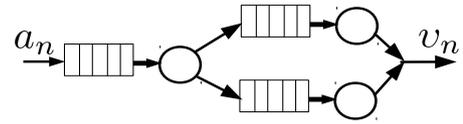
The arrival process to each input port are Poisson with rates  $\lambda_1 = \lambda_2 = \lambda_3 = 2$  entities/sec. Entities service times have an exponential distribution with a mean service rate  $\mu = 10$  entities/sec. The number of simulated entities is  $10^5$ .

Figure 5 shows the pdf of the waiting time for the input queues of port  $p_1$ ,  $p_2$  and  $p_3$  of the multiplexer. The curves are labeled as FDES for the FDES model and Sim for a simulation implemented on SimEvents. The same data that feed the simulation with SimEvents also feed the FDES model. We can observe that the curves match exactly. As can be expected port 1 presents the lower waiting time since it is first served, and port 3 presents the higher waiting time since it is the last served according to the round robin police.

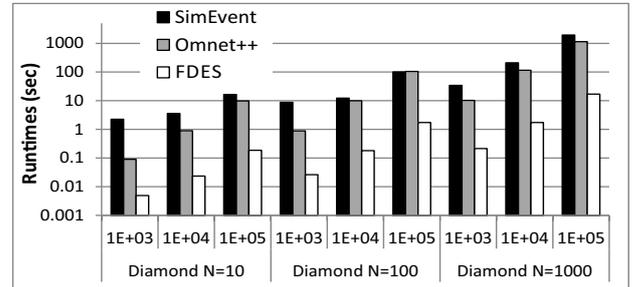
In addition, a simple case study for queueing networks to show a comparison among the FDES model, SimEvents and Omnet++ in terms of the runtime is presented. A tandem of Diamond networks, as the one shown in figure 6, is considered.  $N = \{10, 100, 1000\}$  diamonds were evaluated. The number of simulated entities were  $10^3$ ,  $10^4$  and  $10^5$  for each scenario. The mean arrival rate and the mean service rate were fixed to 0.5 and 1 entities/sec, respectively. An Intel core i7 2600 @ 3.4 GHz computer with 8 GB of RAM, running on a 64-bit Windows 7, was used to obtain model outcomes. Runtimes for SimEvents, Omnet++ and FDES<sup>2</sup> models for different scenarios are shown in figure 7. Results show that Omnet++ and SimEvents runtimes are close to each other, with Omnet++ slightly faster than SimEvents. In contrast, FDES outperforms between one and two orders of magnitude, the aforementioned approaches. Notice that these results can be improved with parallel implementations of FDES models.

## 5. CONCLUSIONS

<sup>2</sup>Implemented in C++.



**Figure 6: Diamond network.**



**Figure 7: Runtimes (log scale) comparison for a tandem of diamond networks at different scenarios.**

A Fast Discrete Event Simulation (FDES) model for queueing network systems was presented. The proposed representation complements the works reported in the literature since allows to simulate queueing network systems with an arbitrary topology. Runtime results show that FDES model outperforms more than one order of magnitude event scheduling approaches for the practical cases evaluated.

## 6. REFERENCES

- [1] T. Baumann, B. Pfitzinger, and T. Jestädt. Simulation driven design of the german toll system profiling simulation performance. In *FedCSIS*, pages 923–926, 2013.
- [2] C. G. Cassandras and S. Lafortune. *Introduction To Discrete Event Systems*. Springer, 2008.
- [3] L. Chen and C.-L. A. . L. A. Chen. A fast simulation approach for tandem queueing systems. In *Proceedings of the 22nd conference on Winter simulation*, pages 539–546. IEEE Press, IEEE, 1990.
- [4] M. I. Clune, P. J. Mosterman, and C. G. Cassandras. Discrete event and hybrid system simulation with simevents. In *Discrete Event Systems, 2006 8th International Workshop on*, pages 386–387. IEEE, 2006.
- [5] W. Kin and V. Chan. Generalized lindley-type recursive representations for multiserver tandem queues with blocking. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 20(4):21, 2010.
- [6] L. Kleinrock. *Queueing systems*. Wiley, 1975.
- [7] N. K. Krivulin. Recursive equations based models of queueing systems. In *Proceedings, European Simulation Symposium 1994*, pages 252–256, 10 1994.
- [8] D. V. Lindley. The theory of queues with a single server. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 48, pages 277–289. Cambridge Univ Press, 1952.
- [9] A. Varga and R. Hornig. An overview of the omnet++ simulation environment. In *Simutools*, page 60, 2008.