

On Improved Modeling and Simulation of Wireless Communication Systems using Multi-agent and Multi-threading Systems Architectures for efficient spectrum reuse

P.M.Papazoglou¹, D.A.Karras², and R.C.Papademetriou³

¹ Sterea Hellas Institute of Technology, Greece

² Sterea Hellas Institute of Technology, Dept. Automation, Chalkis, P.C 34400, Greece,
e-mails: dimitrios.karras@gmail.com, dakarras@ieee.org, dakarras@teiste.gr

³ Dept. Electronics and Computer Engineering, Univ. of Portsmouth, Anglesea Road, Portsmouth, PO1 3DJ UK

Abstract — Efficient design and evaluation of complex wireless communication systems in terms of spectrum reuse can be accomplished using advanced simulation environments. To this goal, network modelling and simulation methodologies are the most critical issues, thus, several Discrete Event Simulation (DES) schemes have been introduced in the literature. In a real wireless network, services such as new call admission, reallocation, etc, should be offered at the same time for different mobile users in the network coverage area. Thus, the network modelling must be based on the network operational concurrency and services independence. The simulation model adaptation to real network behaviour is a challenging goal. Herein, a novel simulation methodology based on multi-threading for simulating wireless networks, compared to the state of the art approach, is presented, preliminarily analyzed and discussed. Moreover, various implementations based on multi-threading and multi-agent concepts are also presented and evaluated.

Keywords — Discrete Event Simulation, Cellular networks, Multi-Agent Systems, Multi-Threading Technologies, Channel Allocation, Spectrum Reuse, Calendar Queue.

I. INTRODUCTION

Existing approaches for the implementation of simulation systems are classified in three main categories [1-5] which are: Discrete Event Simulation (DES), Agent-Based Simulation (ABS) and System Dynamics (SD). DES is a well established approach where the events happen at discrete points in time. In this paper, a comparison between the traditional DES approach based on the Calendar Queue (CQ) scheme and a novel Multi-threading/Multi-Agent (MAS) approach is presented, preliminarily analyzed and discussed for a reference model network.

In order to simulate accurately the behaviour of a real wireless network, several components must be modelled efficiently such as network infrastructure, services, user behaviour, etc. These components can be summarized as follows:

- Network infrastructure (Base Stations, coverage area, transmitting and receiving features, etc)
- Network services (new call, handoff, call stop, etc.)
- Network mathematical models (propagation and models, statistical distributions, user mobility, etc)
- Network operational parameters (cell coverage, channel assignment policy, Base station positions, etc)

The major goal of a channel allocation policy is to

minimize the call blocking (the network can not assign any channel to a new call) and dropping (unsuccessful reallocation) probability. Thus, four basic network procedures- user services are to be basically involved within such a simulation system:

- New call admission (NC, which affects the blocking probability, based on statistical event generation)
- Hand-off (RC, Reallocation, due to user movement or not accepted signal quality, affects the dropping probability)
- User movement (MC, based on user mobility profile and traffic model, based on statistical event generation)
- Call termination (FC, Forced Call, based on call holding time)

The organization of this paper is as follows. In the next section we discuss event generation and scheduling in DES based wireless systems simulation. Section III presents the proposed multi-agent and multi-threading DES modelling approach. In section IV illustrative simulation results are given comparing CQ and MT (4 and 7 threading architectures) strategies. Finally, section V concludes the paper.

II. EVENT GENERATION AND SCHEDULING IN DES SYSTEMS

A. Scheduling in Conventional DES Systems

The three-step cycle scheduling mechanism is the most known among known DES systems [2],[3]. This mechanism operates in three distinct levels which are: (1) Dequeue: Removal of an event with highest priority (the lower time stamp) from the PES (Pending Event Set), (2) Execute: processing of the dequeued event. (3) Enqueue: Insertion of a new event in the PES, NS-2, which is adopted by the 44.4% of the scientific community, uses this scheduling mechanism for event execution. In ns-2, the next earliest event is selected by the scheduler, executing it till completion, and returning to execute the next event (highest priority or lower time stamp among the remaining events). Only one event can be executed at any given time in ns-2, and so it is a single-threaded simulator. If two or more events are scheduled to take place at the "same time", their execution is performed on a first scheduled – first dispatched manner (based on the time stamp of each event). CQ (Calendar Queue) first introduced by Brown R. [2] and analysed in [3] is the basic scheduling mechanism among the most known DES

systems, such as ns-2(Berkeley, USA), Ptolemy II (developed in Berkeley, USA), Jist (developed in Cornell University, USA), etc. Several variations of the CQ for performance improvement of the queue itself (e.g. optimal resizing of the queue, etc) can be found in [2,3]. The time stamp of each event defines its priority. The earliest event on the calendar is dequeued by searching the page for today's date (like a desktop calendar) and removing the earliest event written on that page [2,3]. A CQ is implemented in computer as an array of lists. Each list contains future events. A large list of N events is partitioned to M shorter lists called Buckets. Each bucket (list) corresponds to a specific time range that represents future events.

B. Multi-Threading Technology

In a computer system with $N=P$ where N and P are the processes and processors respectively, parallel execution can be achieved. In most cases, the condition $N>P$ leads to CPU time sharing among the running applications. This execution mechanism is called threading (multi-threading-MT). MT technology, reflects more realistically concurrent events compared to traditional sequential approach [4]. The thread execution scheduling in our case is controlled by the JVM (Java Virtual Machine) but alternative execution methods can be applied. Scheduling can be non-pre-emptive where the current thread is running forever and has to inform the scheduler explicitly if it is safe to start another waiting thread. In the pre-emptive case, a thread is running for a specific CPU time slice and then the scheduler “pre-empts” it, (calling suspend()), and resumes another thread for the next available time slice. Figure 1, illustrates three active threads that share a single processor through switching. The sleep() method deactivates temporarily the current thread in order to give time for execution to another thread.

III. PROPOSED MULTITHREADING / MULTIAGENT PROTOTYPE DES MODELS FOR CELLULAR SYSTEMS

Initially, a prototype reference model has been built in order to verify the correctness of the MT approach compared to the existing CQ approaches. Due to the simplicity of this reference model, one only cell is assumed for serving mobile users within its coverage area. Figure 2, illustrates this cell that offers n channels for supporting new call admission, reallocation and user movement. This model does not contain any advanced mathematical model (e.g. for signal propagation) because it is focused on the way that the clients (mobile users) are served by the channels provided by this cell.

Both prototype models have been tested initially with deterministic scenarios for event generation. Figure 2, depicts how some of the five (NC1, NC2, RC1, MC1, RC2) events in the list shown are executed. The scheduler restores from the queue the event with lower time stamp (highest priority) for the next execution. When the same events are also generated in the MT system, the execution sequence remains the same while the given time slice (Time Slice Width - TSW) provided by the scheduler is greater or equal to the Event Computational Time (ECT).

In other words, if $TSW>ECT$ then, one time slice is enough for the completion of each generated event in the predefined sequence. Thus, these two prototype reference

models produce the same results.

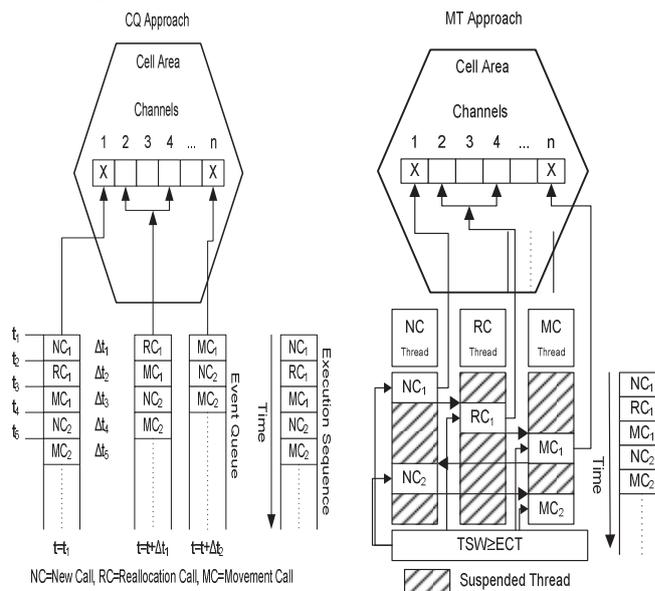


Figure 1.
User service operation based on CQ approach

Figure 2.
User service operation based on MT approach

When the TSW is less than the required computational time for event completion or the TSWs are asymmetrically assigned to active threads, the results are totally different from the traditional CQ approach, since the execution sequence for the list events is totally changed in the MT methodology. This happens because of the competition between the running threads (e.g. individual users) for the common resources (e.g. radio channels) and because of the decision making process concerning the final channel assignment at different points in time.

Agent technology has been used for distributed resource management in telecommunication systems [1],[5]. All the supported network procedures such as new call admission, reallocation (handoff), user movement, etc, are independent one another, interact with network environment, act autonomously, can take decisions in terms of network performance and can be faced as agents. Figure 3, shows the layered Multi-Agent architecture. According to this architecture, there are three different layers that represent the cellular network, the event agent environment and the simulation system environment respectively. This hierarchical structure is built based on different types of functionality. The four network agents in layer-2, interact with the cellular network and a main agent called control agent in layer-3 synchronizes the whole simulation operation. Additionally, efficient negotiation policies can be applied for changing the behaviour of each agent towards improving the whole simulation system performance. All the system agents are implemented as threads within the JVM environment. The final MT environment can be implemented by using two different MT architectures. Figure 4, illustrates the implementation with seven threads (control thread, initial procedures, final procedures, NC,RC,MC,FC) and figure 5 illustrates the implementation with four threads (NC,RC,MC,FC). In both implementations, the scheduling mechanism can be based on JVM environment or in time slices controlled by the developer. Different implementations, based on different number of threads, create different time conditions for the simulation entities and these conditions

affect the simulation results.

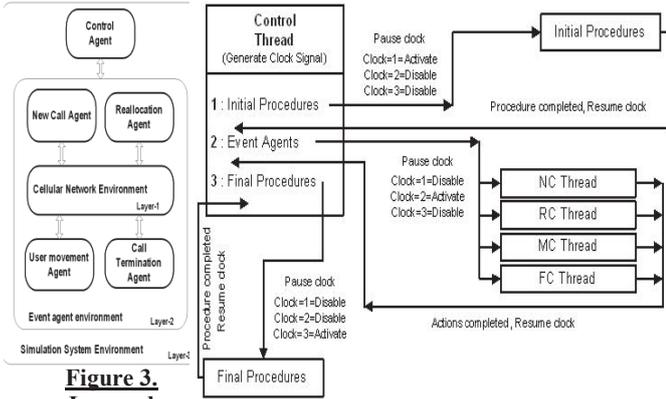


Figure 3.
Layered Multi-Agent Architecture

Figure 4.
The 7 threads architecture

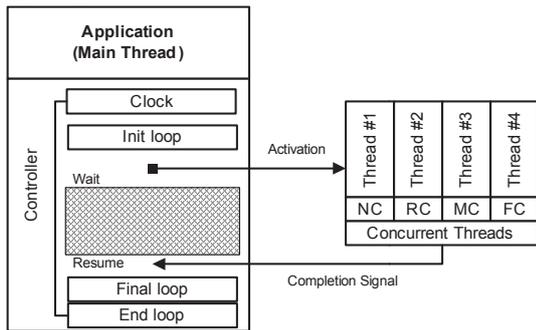


Figure 5. The 4 threads architecture

C. A Preliminary Analysis and Discussion on Multi-threading/ Multiagent Model Advantages in Cellular Systems Spectrum Reuse Simulation Strategies

As we mentioned earlier, in the CQ approach, only one event is executed at any given time. When a new call occurs, the network tries to serve this call by using the selected channel allocation policy. For performing the call establishment, the system takes into account the new call request in combination with the existing mobile users (calls in progress). Thus, this problem of channel allocation is optimized in 1 to N manner. When additional users exist inside the queue (e.g. for a new connection or reallocation), according to the CQ approach, the best channel allocation is performed for the current user only under service. This procedure can improve the communication quality for the current user but also can degrade the communication quality for the next few users that attempt a new call almost at the same time while the current user is being served. This problem can be faced effectively by upgrading the 1 to N relation to an M to N one. In other words, if the system could take into account in its decision making process the few upcoming users apart from the current one under consideration, then the channel allocation could be more sophisticated in order to satisfy communication demands for more users and to keep the operational status of the cellular network more balanced. Assume, for instance, that a new user requests five or more channels for performing real time video. According to the CQ approach, the channel allocation algorithm may assign all the available channels of the associated cell resulting in cell congestion. With a capability, however, for the aforementioned detailed “view” of the nearly simultaneous few upcoming users,

the system would allocate the available channels in the neighbouring cells in such a way that all users could be optimally served with the required communication quality. Using MT technology with controlled time slices for the running threads, this concept can be implemented and additionally, a sophisticated decision making system for channel allocation adapted to current network performance can be developed. The adaptation of the channel allocation policy to the special needs of every individual user in combination with existing and candidate users is a critical goal in contrast to existing approaches where a fixed channel allocation algorithm is applied for only one new user in the queue. To say it in other words, the event generation in a real network is not sequential but several events are generated concurrently and this fact should be reflected in the scheduler of the simulation methodology. Multithreading/ Multiagent technology supports such a concept. Moreover, a real wireless network is a complex dynamical system where the event sequence is not known as in a static system and there is not even known possibilities for each event to be generated so as for the developer to involve Markov chains in the simulation methodology, as regards the scheduler and the event generation model. It is much more realistic to assume that the events described in section III are randomly generated within a large terrain, following a uniform or Gaussian probability model as well as following an unknown probability distribution for concurrency for certain of them.

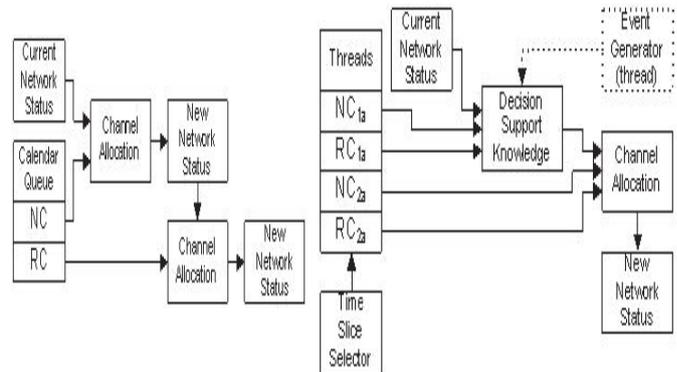


Figure 6.
CQ Approach - Channel Allocation

Figure 7.
MT Approach - Channel Allocation

Figure 6, shows the channel allocation according to the Calendar Queue approach. In CQ, only one user is served at a time and the network does not know anything for any other user except the current one. In MT approach (fig. 7) the network has knowledge for the current as well as an initial knowledge for certain upcoming few users. The active threads that represent the two types of events are divided according to the selected time slice and the system gathers step by step the needed data from every event. Thus, the final channel allocation is optimized not only between current users and the one new upcoming user but between current users and all the “concurrently” upcoming users.

IV EXPERIMENTAL RESULTS

Three statistical metrics have been used in this study. The first two, constitute the network performance measures and the third represents the system stability over Monte Carlo Executions. Blocking probability is defined as

$$P_{blocking} = \frac{\text{number of blocked calls}}{\text{number of calls}} \quad (1)$$

Dropping probability, represents the current capability of the network to give new channels due to unaccepted signal conditions of already connected users.

$$P_{fc} = \frac{\text{number of forced calls}}{\text{number of calls} - \text{number of blocked calls}} \quad (2)$$

The third metric is the standard deviation which measures over the performed Monte Carlo executions. All the simulation scenarios, figures 8-11, use the UDCA (the traditional Unbalanced Distributed Channel Allocation scheme, where the channel allocation is done within the initiated cell) policy for channel assignment in the one-cell model network depicted in section III., for 4 and 7 threading architectures as examples, but without loss of generality.

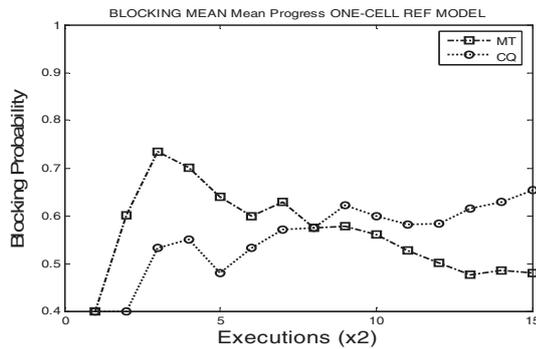


Figure 8. Blocking probability of CQ and MT approaches

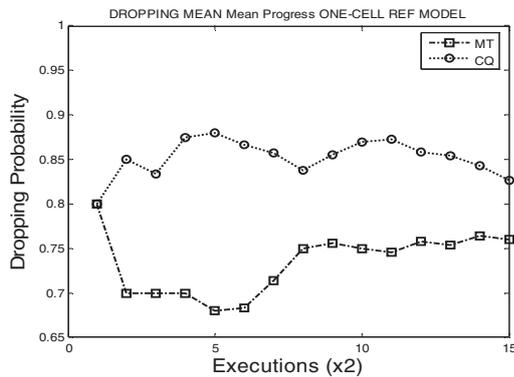


Figure 9. Dropping probability of CQ and MT approaches

Simulation Approach	Std deviation Blocking Probability	Std deviation Dropping Probability
CQ	0.255976	0.12228
Threading	0.248424	0.11832

Table 1. Standard deviation of CQ and Thread approaches over 15 Monte Carlo executions in the one-cell reference model network

V. CONCLUSIONS AND FUTURE WORK

Preliminary analysis, discussion and simulation results obtained by analyzing network performance related statistical metrics, through experimentation with respect to a prototype reference wireless network, show that the new proposed simulation methodology for cellular systems based on the MT technology is a very promising approach compared to the CQ methodology used in the most popular simulation tools for wireless networks. The one-cell reference model proves that the MT implementation model at prototype level is correct due to the same behaviour obtained when compared to the CQ prototype model. In the CQ approach only one user is processed at a

time. When two different users request a service from the wireless network almost at the same time, a competition may occur for accessing the same radio resources.

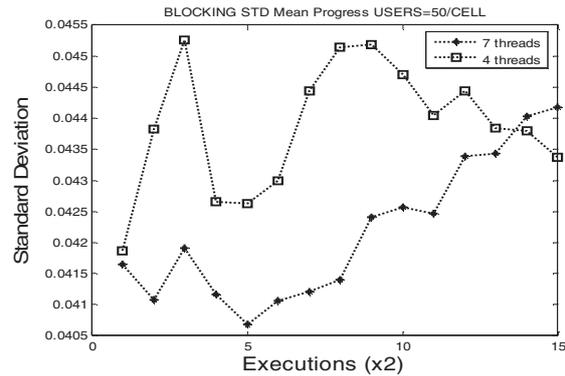


Figure 10. Standard deviation of blocking probability for 4 and 7 threads respectively

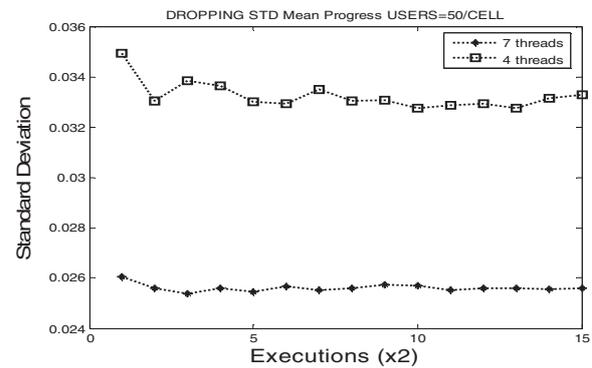


Figure 11. Standard deviation of dropping probability for 4 and 7 threads respectively

Using MT technology, this competition can be reflected more realistically. On the other hand, due to single user processing in CQ, the channel allocation is optimized between a single new user and the already connected ones. If the network knows for example that a set of requests exist from two or more different users for the same radio resources, then the channel allocation would be performed more sophisticated between these new few users (competitive users) and the already connected ones. Thus, the MT approach offers the opportunity for (a) realistic reflection of the competition of various users for the radio resources and (b) optimization of channel allocation to more than one user when new calls occur at almost the same time. Future work includes algorithm development for intelligent channel allocation adapted to current network traffic, based on the MT technology.

REFERENCES

- [1] Megha Kamble, Roopam Gupta (2014), "Software Agent Structure for Performance Index Improvement of Cellular Network", Int. J. Communications, Network and System Sciences, 2014, 7, 331-345.
- [2] Clarence Lehman, Adrienne Keen, and Richard Barnes (2012), "Trading Space for Time: Constant-Speed Algorithms for Managing Future Events in Scientific Simulations", Int'l Conf. Scientific Computing, CSC'12.
- [3] T.Siangsukone, C.Aswakul, L.Wuttisittikulkij, (2003), Study of Optimised bucket widths in Calendar Queue for Discrete Event Simulator, Thailand's Electrical Engineering Conf. (EECON-26).
- [4] Kramer J.M., (2006), Concurrency: State Models & Java Programs, John Wiley & Sons, 2nd Edition.
- [5] P.M.Papazoglou, D.A.Karras, R.C.Papademetriou, (2007), "A Multi-Agent Architecture for Designing and Simulating Large Scale Wireless Systems Resource Allocation", 1st KES Symposium on Agent and Multi-Agent Systems, Springer proceedings LNCS, Poland.