Journal of Universal Computer Science, vol. 14, no. 7 (2008), 1061-1079
submitted: 1/10/07, accepted: 21/1/08, appeared: 1/4/08 © J.UCS

An Improved Multi-Agent Simulation Methodology for Modelling and Evaluating Wireless Communication Systems Resource Allocation Algorithms

Panagiotis Minas Papazoglou
(Lamia Institute of Technology Greece, University of Portsmouth, UK, ECE Dept.
Anglesea Road, Portsmouth, United Kingdom, PO1 3DJ
p.m.papazoglou@hotmail.com)

Dimitrios Alexios Karras
(Chalkis Institute of Technology, Greece, Automation Dept., Psachna, Evoia, Hellas, Greece
P.C. 34400
dakarras@teihal.gr, dakarras@ieee.org, dakarras@usa.net)

Rallis Constantine Papademetriou
(University of Portsmouth, UK, ECE Department, Anglesea Road, Portsmouth
United Kingdom, PO1 3DJ
rallis.papademetriou@port.ac.uk)

Abstract: Multi-Agent Systems (MAS) constitute a well known approach in modelling dynamical real world systems. Recently, this technology has been applied to Wireless Communication Systems (WCS), where efficient resource allocation is a primary goal, for modelling the physical entities involved, like Base Stations (BS), service providers and network operators. This paper presents a novel approach in applying MAS methodology to WCS resource allocation by modelling more abstract entities involved in WCS operation, and especially the concurrent network procedures (services). Due to the concurrent nature of a WCS, MAS technology presents a suitable modelling solution. Services such as new call admission, handoff, user movement and call termination are independent to one another and may occur at the same time for many different users in the network. Thus, the required network procedures for supporting the above services act autonomously, interact with the network environment (gather information such as interference conditions), take decisions (e.g. call establishment), etc, and can be modelled as agents. Based on this novel simulation approach, the agent cooperation in terms of negotiation and agreement becomes a critical issue. To this end, two negotiation strategies are presented and evaluated in this research effort and among them the distributed negotiation and communication scheme between network agents is presented to be highly efficient in terms of network performance. The multi-agent concept adapted to the concurrent nature of large scale WCS is, also, discussed in this paper.

Keywords: Simulation Methodology, Multi-Agent Systems, Agent Negotiation Strategies, Wireless Communication Systems, Resource Allocation, Dynamic Channel Allocation Algorithms
Categories: C.2, I.2.11, I.6.3, I.6.5, I.6.7

1 Introduction

1.1 Simulation of Wireless Communication Systems (WCS)

The performance and behaviour of a real cellular network can be evaluated using simulation systems without the need to perform field experiments and develop prototypes. The simulation solutions offer the opportunity to develop channel allocation schemes, network structures, etc, towards designing a desired cellular network. Due to the complexity of real cellular networks the simulation software development strategy becomes a very critical factor that influences the resulted network model. The challenge of wireless network simulation is finding the way to approach the real network behaviour and not just to speedup the execution time using parallel machines. Three are the main simulation technologies [Chaturvedi et al. 2001]: Discrete Event Simulation (DES), System dynamics and Multi-Agent Systems (MAS).

1.2 MAS in WCS

The MAS technology has sense and produces the desired results only in cases where the multi-agent concept can be applied and the agent features such as adaptability [Splunter et al. 2003], [Russel and Norvig 2002], autonomy [Huhns et al. 1998], [Norman and Long 1995], [Ekdahl 2001], collaboration, interactivity, etc are used in practice.

An overview of agent technology in communication systems is presented in [Hayzelden and Bigham 1999]. This overview is focused on software agents that are used in communications management. More precisely, agents can be used to cope with some important issues such as network complexity, Mobile User (MU) mobility and network management. A MAS for resource management in wireless mobile multimedia networks is presented in [Iraqi and Boutaba 2000]. Based on the proposed MAS in [Iraqi and Boutaba 2000], the call dropping probability is low while the wireless network offers high average bandwidth utilization. According to [Iraqi and Boutaba 2000], the final decision for call admission is based on the participation of neighbour cells. Thus an agent runs in each cell or Base Station (BS). A cooperative negotiation in a MAS for supporting real-time load balancing of a mobile cellular network is described in [Bigham and Du 2003]. In the above proposed MAS, agents are used (a) for the representation of different service providers in the market and (b) for network operators that manage the radio resource of different network regions. According to [Bigham and Du 2003], agent communication is achieved through messages and the final agreements are based on negotiation. In the above study, negotiation constitutes an intelligent control technique for adjusting dynamically the cell shape and size and for balancing the traffic load over the network.

A comprehensive simulation model for wireless cellular networks has been built by [Bodanese 2000] in order to propose a distributed channel allocation scheme using intelligent software agents. In the above study, intelligent collaborative software agents give autonomy to BSs, increase the network robustness, allow negotiation of network resources and improve resource allocation. For this purpose, several aspects of the cellular network infrastructure and operation have been modelled.

In the present research effort, a novel modelling methodology for supporting wireless network services based on MAS technology is also presented. **The novelty lies on the fact that instead of involving Base stations (BS), service providers, network operators or other similar physical entities as agents in the MAS system, we propose, instead to model concurrent network procedures (services), which are more abstract entities, as agents in the proposed simulation methodology.**

1.3 Channel Allocation

The capacity of a cellular system can be described in terms of the number of available channels, or the number of MUs the system can support. The total number of channels made available to a system depends on the allocated spectrum and the bandwidth of each channel. The available frequency spectrum is limited and the number of MUs is increasing day by day, hence the channels must be reused as much as possible to increase the system capacity. The allocation of channels to cells or mobile users is one of the fundamental resource management issues in a mobile communication system. In the literature, many channel allocation schemes have been widely investigated with the goal to maximize the frequency reuse. The channel allocation schemes in general can be classified into three strategies: Fixed Channel Allocation (FCA) [Zhang and Yum 1989], [Lai and Coghill 1996], [MacDonald 1979], [Elnoubi et al. 1982], [Xu and Mirchandani 1982], Dynamic Channel Allocation (DCA) [Zhang and Yum 1989], [Cimini and Foschini 1992], [Cox and Reudink 1973], [Re et al. 1996], [Sivarajan et al. 1990], and the Hybrid Channel Allocation (HCA) [Zhang and Yum 1989], [Kahwa and Georgans et al.]. In FCA, a set of channels are permanently allocated to each cell based on a pre-estimated traffic intensity. The FCA scheme is simple but does not adapt to changing traffic conditions and MU distribution. In DCA, there is no permanent allocation of channels to cells. Rather, the entire set of available channels is accessible to all the cells, and the channels are assigned on a call-by-call basis in a dynamic manner. To overcome the drawbacks of FCA and DCA, the HCA combines the features of both FCA and DCA techniques.

2 The proposed MAS for large scale WCS

2.1 Network operation

The needed information for the simulation of the cellular network can be categorized as follows:

- MU characteristics - MU attributes including connection status, current position, CHT, allocated channel, communication signal strength and interference (if necessary), etc.
- Network parameters - Cells number, channels per cell, cell positions, BS positions, etc.
- Traffic and QoS - new call arrival schemes, MU movement distribution, call dropping conditions, hand-of conditions, etc.
- Channel allocation schemes – channel allocation strategy according to network conditions, congestions cases, resources limitations, etc.

2.2 Supported network services

New call arrival (NC)
The number or MUs is large, the calls by each MU are limited and so the call arrivals can be assumed as random and independent. In the proposed simulation methodology, the new calls are resulted from a random or a Poisson distribution with regards to a predefined daily model.

Call termination (FC)
The present simulation methodology uses an exponential function that generates the call duration for each new MU. The Call Holding Time is added to current simulation time for later examination. The associated procedure scans the network cells and searches for any connected MUs and examines the progressive call time. If this time is expired the MU is disconnected.

Reallocation check (RC)
The computations are based on signal strength and how is affected from other connected MUs in neighbouring cells. If a MU signal does not fulfil the Carrier to Noise plus Interference Ratio threshold the procedure tries to find another appropriate channel. Firstly, the algorithm calculates the signal strength between MU and Base Station (BS) and at a later time calculates any interference from around connected MUs. If an accepted channel is found, is allocated to the new MU, otherwise the call is dropped.

MU movement (MC)
The algorithm locates the connected MUs and changes their current positions. A MU movement is generated based on Gaussian distribution. This distribution is also used in similar simulation models found in the literature [Nishith et al. 1998].

Network agents
The above services have been modelled as four distinct agents in the proposed simulation methodology. The NC agent (NCA) and RC agent (RCA) constitute the most important agents due to the fact that they affect the network performance in terms of blocking and dropping probabilities, which are the basic evaluation metrics.

2.3 Towards an Analytical Model of the Proposed Multi-Agent Architecture and the Involved Agents Interaction

Network agent definition
According to [Wooldridge and Jennings 1995], "An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives".

An agent interacts with its environment, gets input information from it and performs actions that affect this environment (figure 1).

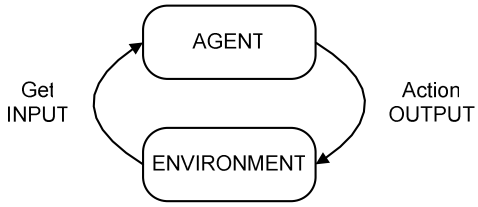

Figure 1: Agent definition

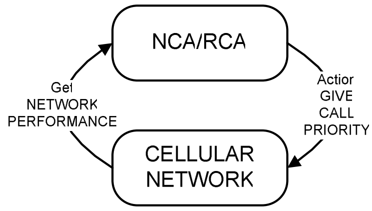

Figure 2: Proposed Network Agents definition

NC and RC agents (NCA,RCA) interact with the cellular network (environment), get input information (blocking probability-network performance for NCA, dropping probability for RCA) and perform actions (give priority to new calls or in handoffs) that affect the network performance (Figure 2).

The basic agent capabilities involved can be summarized as follows:

- *Reactivity.* An agent perceives its environment and responds in order to satisfy the design objectives. NCA perceives network performance and a) gives priority to new calls, b) negotiates with RCA for the best agreement in order to satisfy its design objective which is the minimization of blocking probability (network performance optimization). RCA works similarly but for dropping probability.
- *Proactiveness.* Takes the initiative to exhibit its goal oriented behaviour to satisfy the design objectives. Network Agents send messages to other agents (NCA to RCA and vice versa) in order to get performance benefits for the network.
- *Social ability.* Interaction with other agents for achieving the design objectives. NCA interacts with RCA for achieving the design objectives

Architecture of the intelligent network agents
Assume that the possible discrete states of the environment can be described by the set **E** as

$$E=\{e,e',\dots\} \tag{1}$$

Assume now that the possible discrete states of the wireless network environment can be described by the set **E** as

$$E=\{LL,LH,HL,HH\} \tag{2}$$

Where, the members of E represent the network performance, L represents low level and H the high level. The pairs correspond to blocking and dropping probability respectively.

On the other hand, it is assumed that each agent has a set of possible actions on this environment. These actions change the environment status and are defined by the set

$$Ac=\{\alpha,\alpha',\ldots\} \quad (3)$$

In the case of network environment, the corresponding actions are defined by the set

$$Ac=\{IP_{NC},IP_{RC},DP_{NC},DP_{RC},DN_{NC},DN_{RC}\} \quad (4)$$

Where, IP is the action "Increase Priority", DP represents the action "Decrease Priority" and finally DN is the action "Do Nothing".

The environment changes its state according to the above actions. A sequence of actions causes a sequence of state changes. Thus, a run, **r**, of an agent within the environment can be expressed as

$$r : e_0 \xrightarrow{\alpha_0} e_1 \xrightarrow{\alpha_1} e_2 \xrightarrow{\alpha_2} e_3 \xrightarrow{\alpha_3} \cdots \xrightarrow{\alpha_{u-1}} e_u \quad (5)$$

A run r for the cellular network becomes

$$r : HL \xrightarrow{IP_{NC},DN_{RC}} LL \xrightarrow{DN_{NC},DN_{RC}} LH \xrightarrow{DN_{NC},IP_{RC}} LL \xrightarrow{DN_{NC},DN_{RC}} \cdots \xrightarrow{\alpha_{u-1}} e_u \quad (6)$$

Let also, define the following sets:

R. Set of possible finite sequences (over E and Ac)
$\mathbf{R^{AC}}$. Subset of R, that ends with an action
$\mathbf{R^{E}}$. Subset of R, that ends with an environment state

A state transformer function is introduced [Wooldridge 2004] in order to describe the effect of an agent (action) on the environment:

$$\tau : R^{Ac} \to \gamma(E) \quad (7)$$

The above function maps a run to a set of possible environment states. When no successor state exists to **r**, **τ(r)** becomes

$$\tau(r) = \varnothing \quad (8)$$

The whole environment is expressed (states, transformer function) as

$$Env = \langle E, e_0, \tau \rangle \quad (9)$$

Where, **E** is the state set, $\mathbf{e_0}$ is the initial state and τ the transformer function. For modelling agents, it is assumed that an agent represents a function for mapping runs to actions and so:

$$Ag : R^E \to Ac \quad (10)$$

In other words, an agent makes decisions about action (what action to perform) based on the history of the system. For representing now the whole system (agents, environment) a set is defined

$$R(Ag, Env) \quad (11)$$

Finally, the sequence $(e_0, \alpha_0, e_1, \alpha_1, e_2, \ldots)$ represents a run of an agent Ag (in the environment) $Env = \langle E, e_0, \tau \rangle$, if :

$$\alpha_0 = Ag\left(e_0\right) \tag{12}$$

A run of agent NCA or RCA is represented by the sequence $(\mathrm{HL}, IP_{NC}, \mathrm{LL}, DN, \mathrm{LH}, ...)$ if $IP_{NC} = Ag\left(HL\right)$ and for u > 0,

$$e_u \in \tau\left(\left(e_0, \alpha_0, ..., \alpha_{u-1}\right)\right) \tag{13}$$

$$\alpha_u = Ag\left(\left(e_0, \alpha_0, ..., e_u\right)\right) \tag{14}$$

In the case of network environment and for u>0

$$e_u \in \tau\left(\left(HL, IP_{NC}, DN_{RC}, ..., \alpha_{u-1}\right)\right) \tag{15}$$

$$\alpha_u = Ag\left(\left(HL, IP_{NC}, DN_{RC}, ..., e_u\right)\right) \tag{16}$$

where, Ag represents NCA or RCA.

Network Agent interaction

As mentioned before, an agent perceives environment and acts on it. These two distinct activities are represented by two functions respectively (figure 3).

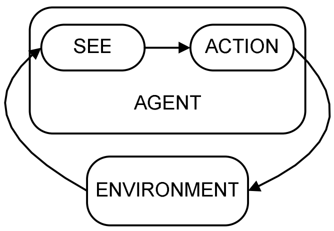

Figure 3: Agent interaction

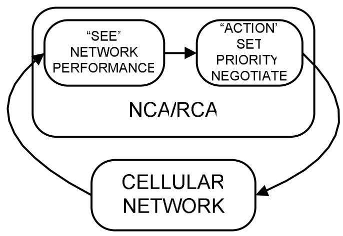

Figure 4: Network Agent interaction

Similarly, NCA and RCA, perceive environment and act on it. These two distinct activities for NC and RCA are represented by two functions respectively (figure 4).

The "see" function maps environment states to perceptions and "action" maps sequences of perceptions to actions.

As an example of the above approach let x represent the statement "metric M1 is acceptable" and let y represent the statement "metric M2 is acceptable". Thus, the set E contains four combinations of x and y. This set can be expressed as follows:

$$E = \left\{\left\{\overline{x}, \overline{y}\right\}, \left\{\overline{x}, y\right\}, \left\{x, \overline{y}\right\}, \left\{x, y\right\}\right\} \tag{17}$$

with (18)

$$e_1 = \left\{\overline{x}, \overline{y}\right\}, e_2 = \left\{\overline{x}, y\right\}, e_3 = \left\{x, \overline{y}\right\}, e_4 = \left\{x, y\right\}$$

Network behaviour is evaluated through two basic statistical metrics which are the blocking and dropping probability. Thus, (17) and (18) will be expressed in terms of the above metrics as follows:

$$E = \left\{ \left\{ \bar{B}, \bar{D} \right\}, \left\{ \bar{B}, D \right\}, \left\{ B, \bar{D} \right\}, \left\{ B, D \right\} \right\} \quad (19)$$

with (20)

$$e_1 = \left\{ \bar{B}, \bar{D} \right\}, e_2 = \left\{ \bar{B}, D \right\}, e_3 = \left\{ B, \bar{D} \right\}, e_4 = \left\{ B, D \right\}$$

where, B represents the statement "Blocking probability is acceptable" and D represents the statement "Dropping probability is acceptable". Now, the set E contains four combinations of B and D.

The "see" function of the agent, will have two perceptions in its range, P1 and P2 that indicate if the metric M1 is acceptable or not. The behaviour of the "see" function can be described as follows:

$$see(e) = \left\{ \begin{array}{lllll} P_1 & if & e = e1 & or & e = e2 \\ P_2 & if & e = e3 & or & e = e4 \end{array} \right\} \quad (21)$$

The "see" function of the NCA, will have two perceptions in its range, P1 and P2 that indicate if the blocking probability is acceptable or not. The behaviour of the "see" function can be described as follows:

$$see(e) = \left\{ \begin{array}{llllll} P_1 & if & e = e_1 & or & e = e_2 & bad \\ P_2 & if & e = e_3 & or & e = e_4 & good \end{array} \right\} \quad (22)$$

Similarly for RCA, the see(e) is formulated as follows:

$$see(e) = \left\{ \begin{array}{llllll} P_1 & if & e = e_1 & or & e = e_3 & bad \\ P_2 & if & e = e_2 & or & e = e_4 & good \end{array} \right\} \quad (23)$$

With two given environment states $e \in E$ and $e' \in E$, then e~e' can be written only if see(e)=see(e'). An agent has perfect perception if the different environment states are equal to distinct perceptions.

2.4 Multi-Layered/Multi-Agent simulation model

Figure 5, illustrates the proposed multi-layered multi-agent architectural model. This model represents the integration of the modelled network procedures (services) as agents inside the simulation system environment. The whole simulation model is divided in three layers based on the corresponding functionality. Layers one and two constitute the whole network framework. Agents in layer two interact with the core cellular network environment and guarantee the network functionality in terms of calls management. Layer three controls the whole simulation process by synchronizing the agent activation. The simulation system has been implemented in JVM multi-threading environment.

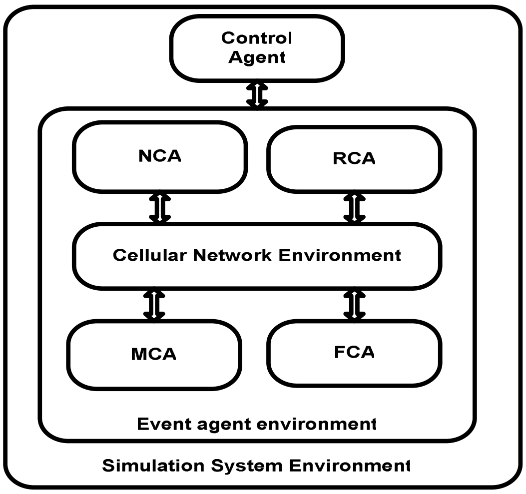

Figure 5: Multi-Layered / Multi-Agent core Architecture

2.5 The proposed Negotiation strategies analysis

Each agent has a "see" function for environment perception and an "action" function that maps sequences of perceptions to actions. The result of every negotiation is the new priority settings based on the current environment status. This network environment status is examined in terms of thread delays (for servicing MUs, accessing shared resources, etc) or network performance. Thus the resulted negotiation strategies are thread delay based or performance based respectively. Each agent is responsible to operate for a specific target. For example, FCA is responsible for the call termination service and MCA for the MU movement. On the other hand, the agents NCA and RCA are responsible for new call admissions and handoffs respectively. When two or more requests arrive in the network (e.g. a new call and a handoff request) the corresponding services must be offered in an optimal way such as to keep in balance the desired network performance level.

Thread delay based negotiation

The first target using this scheme is to create a balanced delay environment for all proposed network agents and analyze the results. This first strategy is based on a central negotiation management scheme. The control agent (fig. 6) receives information from other agents in order to control agent priorities. By controlling agent priorities we can control the mentioned timing attributes (such as delay for accessing data, thread life time, etc) with direct influence to the simulation results. The key for controlling agent behaviour in terms of timing constraints is the priority control. Priority can be controlled by keeping in memory previous time behaviour for each agent. Such agent priorities are changing over simulation time in a dynamical manner. The central control agent acquires new knowledge at every new simulation step concerning the timing behaviour of each agent (thread delays etc.) and can accordingly modify their priorities.

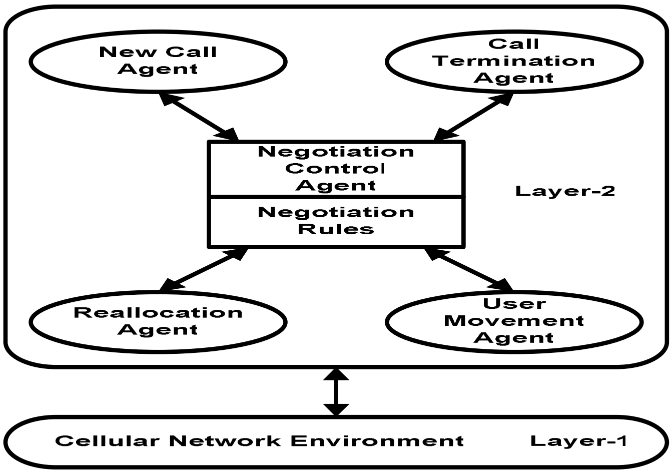

Figure 6: Thread delay based negotiation architecture

The algorithm is simple but efficient and is based on the following simple priority setting rule, next defined:

$$NewP = NP \cdot \left[1 + \frac{Sumc[i]}{a \cdot SumMax}\right] \qquad (24)$$

Where *NewP* is the resulting new priority within a new simulation step of a selected agent, *NP* is the normal priority, *Sumc[i]* is the total delay of agent *i* for accessing common resources, *a* is a user defined coefficient, *SumMax* is the highest delay of an agent. Whenever this rule is applied for setting agents priorities, the control agent compares each agent delay with the current maximum in order to decide for the significance of this delay. The more significant the delay is realized, the more priority is given to the corresponding agent. Thus, it is expected that agent delays would tend to zero and balanced conditions would be created between the agents. Fig. 7, shows the complete negotiation dialog between network agents and the control agent. Initially, every agent informs the control agent for its completion. After the completion of all agent tasks within the simulation step, the control agent exchanges information messages with all other agents in order to state the final priority assignment decision to each of them. Finally, each agent follows this decision of the control agent.

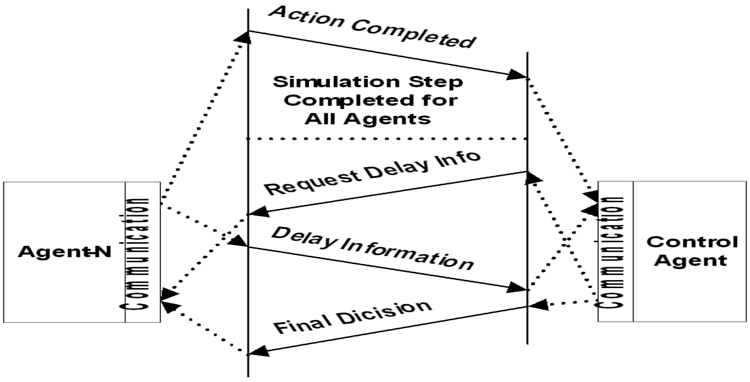

Figure 7: Negotiation dialog between agents and control agent

Performance based negotiation
The negotiation in this second proposed approach is made directly between the active agents (fig. 8) and the interaction is competitive or cooperative based on the current performance status of the network. RCA and NCA are the main agents that affect the network performance in terms of blocking, dropping probabilities and other statistical metrics.

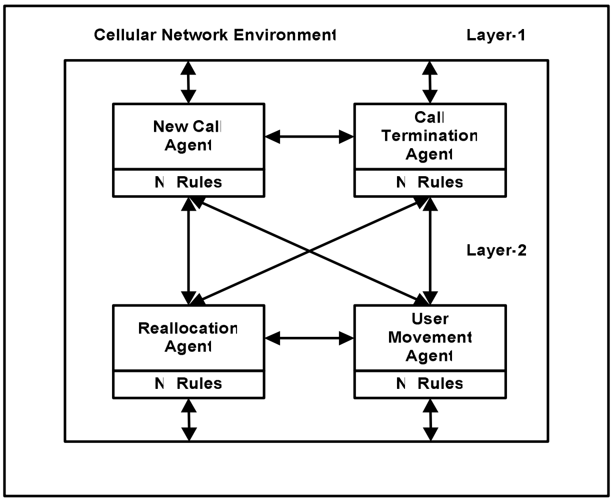

Figure 8: Performance based negotiation architecture

After every simulation step completion, each agent checks its current status in terms of how it affects the network behaviour and takes decisions for negotiation or not with other agents. In the implemented negotiation scheme the main agent interaction is between NCA and RCA. NCA, after each simulation step completion checks the ratio of blocking probability between current and previous simulation step. If this ratio shows that the blocking probability in current step is greater than in previous step, the new call agent increases its priority and leaves a message for the RCA (request for priority decrement). If current blocking probability is less than in previous step, the new call agent decreases its priority to help the performance of RCA. NCA and RCA checks for incoming message from the other agent and takes decision according to this incoming message and its current status. Figure 9, shows a complete negotiation dialog between the above mentioned network agents.

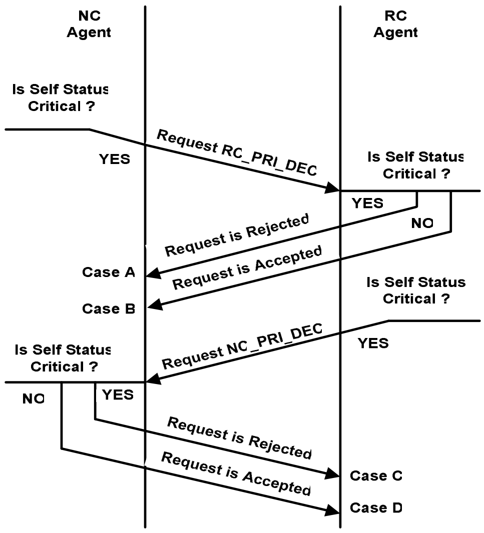

Figure 9: Negotiation dialog for performance based strategy

//decision making (NCA)
Check Blocking Probability (BP-NCA)
Is BP-NCA critical?
  NO:
    Is BP-NCA stable?
      YES:
        Do nothing
      NO: Decrease self priority (help RCA to perform better)
  YES:
    Increase self priority
    Start negotiation with RCA (request priority decrement from RCA)

Figure 10: Decision making algorithm for NCA based distributed negotiation strategy between NCA, and RCA

If NCA or RCA affects positively the network performance, then any possible negotiation is cooperative. On the other hand, if both agents affect negatively the network performance, then any possible negotiation is competitive. Figure 10 shows the decision making algorithm for NCA and RCA agents distributed negotiation strategy.

The above decision making algorithm might be based on dropping probability in the case of RCA. When an agent receives a message (request for priority decrement), first checks its status (how it affects the network performance) and based on that status, accepts (cooperation) or rejects (competition) the request. Additionally, a request for priority decrement is rejected when the lowest level of priority is already reached.

2.6 Scaling Up

As mentioned before in the Multi-Agent model, four agents represent the offered services by the network. When the network under investigation is a large scale network, the proposed network agents have to be distributed in the whole network. Assuming that the cellular network has ***N*** cells distributed in cell clusters where each cluster contains ***i*** cells, the total number of clusters is ***N/i***. Each set of the four agents is duplicated in every cluster. Thus, the total required agents are ***4*(N/i)***. In order to achieve acceptable adaptation of the MAS to the current traffic conditions and to improve the network performance the agent negotiation takes place between NCA and RCA of a set of clusters that belong in the same influence area.

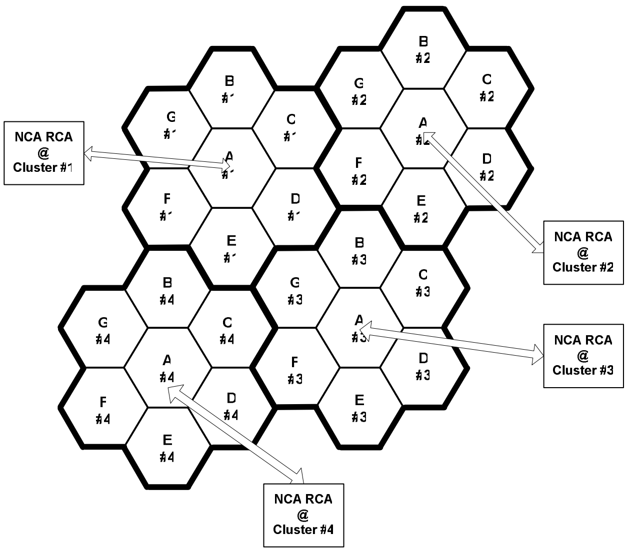

Figure 11: Scaling up

For a wireless network with 28 cells (fig. 11), the total number of 7-cell clusters is 28/7=4. The total number of agents needed is 4*4=16 and especially for NCA and RCA the needed agents are 2*4=8 agents.

3 Evaluation of the Proposed Simulation Model

The blocking probability is one of the most important characteristics for measuring the performance of a cellular network. When a new call arrival occurs and the network can not allocate a channel then, we say that this call is blocked. The blocking probability $\mathbf{P_{blocking}}$ is calculated from the ratio

$$P_{blocking} = \frac{number\ of\ blocked\ calls}{number\ of\ calls} \tag{25}$$

The dropping probability is also an additional and very important characteristic for measuring the cellular network performance. When a call is in progress and the required quality conditions are not met then, this call is obligatorily driven to termination. The dropping probability $\mathbf{P_{fc}}$ is calculated from the ratio

$$P_{fc} = \frac{number\ of\ forced\ calls}{number\ of\ calls - number\ of\ blocked\ calls} \tag{26}$$

4 Experimental Results

The experimental results have been generated using Monte Carlo executions and are divided in two categories based on the type of the Agents negotiation scheme proposed in section 2.

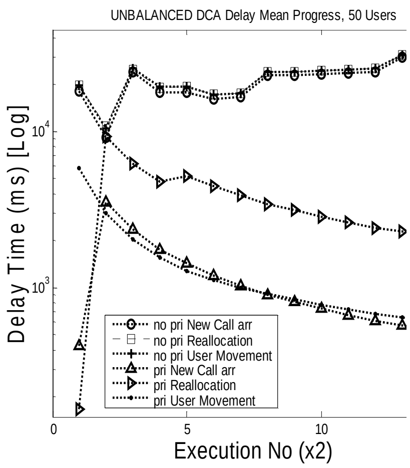

Figure 12: Agent delay time before and after priority settings (no pri means no priority involvement, pri means with taking into account priority)

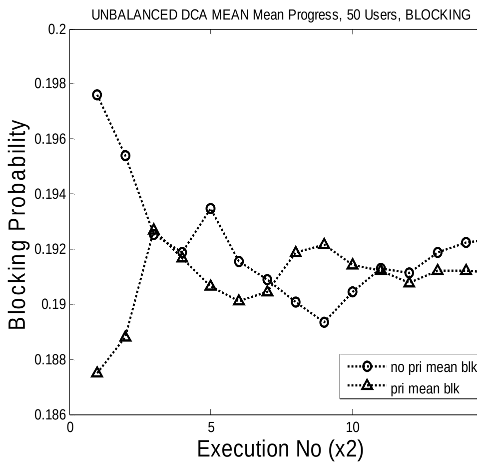

Figure 13: Blocking probability of Classic DCA before and after agent negotiation dialog (no pri means no priority involvement, pri with taking into account priority, blk means blocking probability)

Thread delay based (TDB) negotiation

The experimental results have been generated using 30 Monte Carlo executions. Without setting dynamically the thread priorities, the system gives equal priorities to all threads and the average total delay is above 4.4x104 ms (fig. 12). Activating the negotiation dialog between agents and control agent (dynamically setting of priorities) the average delay time is kept below 1000ms (1sec).

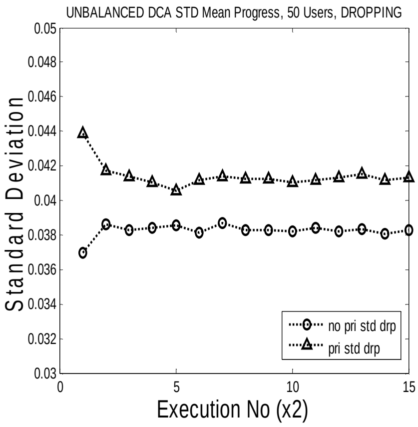

Figure 14:. Dropping probability of Classic DCA before and after agent negotiation dialog (no pri means no priority involvement, pri means with taking into account priority, std means standard deviation and drp means dropping probability)

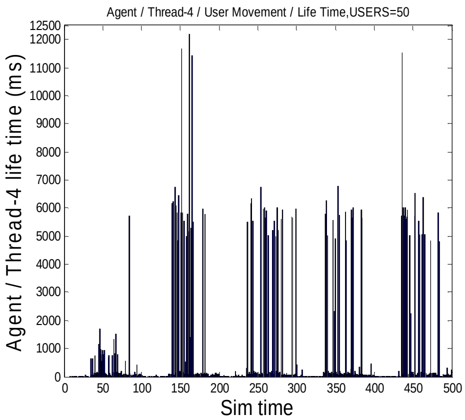

Figure 15: Sample graph of Agent life time for agent "User Movement"

Figures 13 and 14 illustrate the impact of agent negotiation dialog to the simulation model behaviour in terms of blocking and dropping probability.

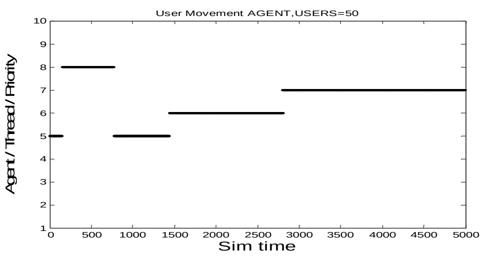

Figure 16: Sample graph of priority settings

Figure 15 shows a typical diagram for the life time (active time) of a network agent. More specifically, in this graph a diagram is illustrated for the user's movement agent. Finally, figure 16 presents the properties dynamic change for the same agent, as in fig. 15, and more specifically the time evolution of its priority settings.

Network Performance based (PB) negotiation

Figures 17 through 20, show that the performance based (PB) negotiation scheme performs better in terms of network behaviour compared to thread delay based (TDB) negotiation and the no negotiation approach.

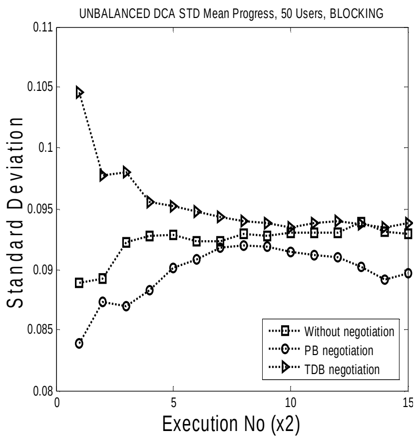

Figure 17: STD (Standard Deviation) mean progress of blocking probability for all negotiation schemes

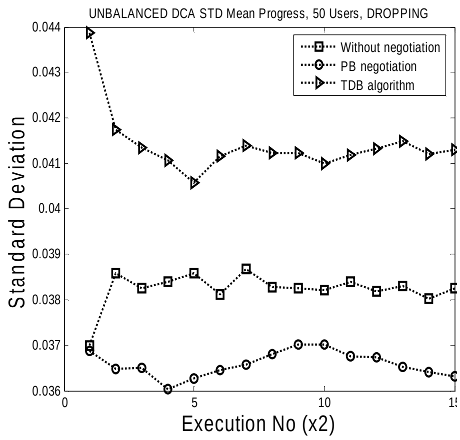

Figure 18: STD (Standard Deviation) mean progress of dropping probability for all negotiation schemes

Another important aspect of the simulation system behaviour, when the performance based negotiation is applied, is the stability in terms of standard deviation of the blocking and dropping probability respectively (figures 17 and 18). This stability is mainly based on the most fair thread control due to balancing conditions between performance metrics within the network.

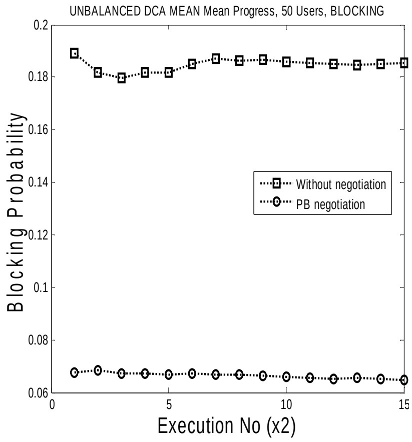

Figure 19: Mean progress of blocking probability

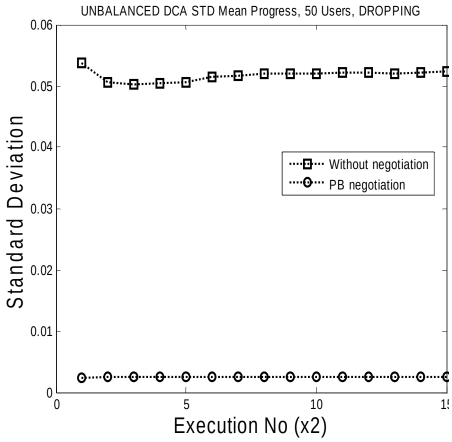

Figure 20: Mean progress of dropping probability

5 Conclusions and Future work

An advanced multi-layer multi-agent architecture for simulating network services modelled as agents and integrated in a wireless simulation system environment has been presented and analyzed in this paper. The most known behavioural and structural features of the multi-agent concepts have been adapted in the design and development of an efficient simulation model for WCS. Two negotiation strategies between the modelled network agents have been proposed. The proposed negotiation strategies within the multi-agent model can lead to a higher adaptation of the simulation model to the real resource allocation needs of the wireless network under investigation. The thread delay negotiation (TDB) is strongly connected with the timing behaviour of the implemented threads that are controlled by the JVM (Java Virtual Machine). This approach does not give always efficient results and has the drawback of poor control. On the other hand, the performance based negotiation (PB) has been demonstrated to be very effective, it is totally independent from the implementation technology (Java for instance), it can be applied in different wireless networks and it can be adapted to the dynamically changing traffic conditions of a wireless network. Giving specific priority to network agents, the channel assignment and the MU servicing can be efficiently optimized in terms of blocking and dropping probability which are the most known performance metrics for a wireless network. Finally, the performance based negotiation scheme integrated within the multi-agent model as well as the proposed MAS system can be easily scaled up for supporting, also, large scale wireless networks.

References

[Bigham and Du 2003] Bigham, J., Du, L.: “Cooperative Negotiation in a MultiAgent System for Real Time Load Balancing of a Mobile Cellular Network”, AAMAS’03, July 14–18, 2003

[Bodanese 2000] Bodanese, E.L.: "A Distributed Channel Allocation Scheme for Cellular Networks using Intelligent Software Agents", PhD thesis, University of London, 2000

[Chaturvedi et al. 2001] Chaturvedi, A., Dickieson, J., Dolk, D.R., Scholl, J.: “Introduction To Agent-Based Simulation And System Dynamics Minitrack”, Proceedings of the 34th ,Hawaii International Conference on System Sciences, 2001

[Cimini and Foschini 1992] Cimini L.J., Foschini, G.J.: “Distributed Algorithms for Dynamic Channel Allocation in Microcellular Systems”, IEEE Vehicular Technology Conference, pp.641-644, 1992.

[Cox and Reudink 1973] Cox D. C., Reudink, D. O.: “Increasing Channel Occupancy in Large Scale Mobile Radio Systems: Dynamic Channel Reassignment”, IEEE Transactions on Vehicular Technology, vol.VT-22, pp.218–222, 1973.

[Ekdahl 2001] Ekdahl, B.: “How Autonomous is an Autonomous Agent?”, Proc. of the 5th Conference on Systemic, Cybernetics and Informatics (SCI 2001), July 22-25, 2001, Orlando, USA.

[Elnoubi et al. 1982] Elnoubi, S. M., Singh, R., Gupta, S.C.: “A New Frequency Channel Assignment Algorithm in High Capacity Mobile Communication Systems”, IEEE Transactions on Vehicular Technology, vol. VT-21, no. 3,pp. 125–131, 1982.

[Hayzelden and Bigham 1999] Hayzelden A., and Bigham, J.: “Software Agents for Future Communications Systems”, Springer-Verlag, Berlin, 1999

[Huhns et al. 1998] Huhns, M., Singh M. (Eds.): “Agents and Multiagent Systems: Themes, Approaches, and Challenges”. Readings in Agents, Chapter 1, Morgan Kaufmann Publishers, USA, 1998, pp. 1-23.

[Iraqi and Boutaba 2000] Iraqi Y., Boutaba, R.: "A Multi-agent System for Resource Management", in Wireless Mobile Multimedia Networks, LNCS 1960, pp. 218–229, Springer-Verlag Berlin Heidelberg 2000

[Kahwa and Georgans et al.] Kahwa, T.J., Georgans, N.D.: “A Hybrid Channel Assignment Schemes in Large-Scale, Cellular Structured Mobile Communication Systems”, IEEE Transactions on Communications, vol.26, pp 432–438, 1978.

[Lai and Coghill 1996] Lai W.K., Coghill, G.C.: “Channel Assignment through Evolutionary Optimization”, IEEE Transactions on Vehicular Technology, vol.45, no.1, pp91-96, 1996.

[MacDonald 1979] MacDonald, V.H.: “The cellular Concepts,” The Bell System Technical, Journal, vol.58, pp.15–42, 1979.

[Nishith et al. 1998] Tripathi, N.D., Reed N.J.H., VanLandingham, H. F.: " Handoff in Cellular Systems ", IEEE Personal Communications, 1998

[Norman and Long 1995] Norman, T., Long, D.:“Goal Creation in Motivated Agents”. In: Wooldridge, Jennings (Eds.), Intelligent Agents: Theories, Architectures, and Languages, LNAI 890: Springer, 1995.

[Re et al. 1996] Del Re, E., Fantacci, R., Giambene, G.: "A Dynamic Channel Allocation Technique based on Hopfield Neural Networks", IEEE Transactions on Vehicular Technology, vol.VT-45, no.1, pp.26–32, 1996.

[Russel and Norvig 2002] Russell, S., Norvig, P.: "Artificial Intelligence: A Modern Approach". Prentice Hall, 2nd ed, 2002.

[Sivarajan et al. 1990] Sivarajan, K.N., McEliece, R.J., Ketchum, J.W. "Dynamic Channel Assignment in Cellular Radio", IEEE 40th Vehicular Technology Confer-ence, pp.631–637, 1990.

[Splunter et al. 2003] Splunter, S., Wijngaards, N., Brazier, F.: "Structuring Agents for Adaptation". In: E. Alonso et al (Eds), Adaptive Agents and Multi-Agent Systems, LNAI, Vol. 2636, 2003, pp. 174-186.

[Wooldridge and Jennings 1995] Wooldridge, M. and Jennings, N.R: Intelligent agents: theory and practice. The Knowledge Engineering Review, 10(2), 115-152, 1995.

[Wooldridge 2004] Wooldridge M., An Introduction to Multi-Agent Systems, Wiley, 2004.

[Xu and Mirchandani 1982] Xu Z., Mirchandani, P.B.: "Virtually Fixed Channel Assignment for Cellular Radio-Telephone Systems: A Model and Evaluation", IEEE International Conference on Communications, ICC'92, Chicago, vol. 2, pp. 1037–1041, 1982.

[Zhang and Yum 1989] Zhang, M., Yum, T. S.: "Comparisons of Channel Assignment Strategies in Cellular Mobile Telephone Systems", IEEE Transactions on Vehicular Technology, vol.38, no.4, pp.211-215, 1989.