A Simulation Setup Validation Framework for Modeling Architectures and Algorithms in Designing Large Scale Wireless Communication Systems

P.M.PAPAZOGLOU[1,3], D.A.KARRAS[2], R.C.PAPADEMETRIOU[3]
[1]Department of Informatics & Computer Technology
Lamia Institute of Technology, Greece
p.m.papazoglou@hotmail.com, papaz@teilam.gr
[2]Department of Automation Engineering
Chalkis Institute of Technology, Greece
dakarras@ieee.org, dimitrios.karras@gmail.com, dakarras@teihal.gr
[3]Department of Electronic & Computer Engineering
University of Portsmouth, United Kingdom

Abstract – Channel allocation is one of the fundamental issues in designing wireless communications systems due to the fact that it determines how the available bandwidth will be managed over the changeable user demands. The limited channel capacity and the increasing requirements for advanced services such as real time video grant channel allocation strategies a special role. Plethora of channel allocation strategies have been proposed in the literature for supporting GSM and 3G communication as well as multimedia data services and tested through generic or specific simulation architectures for medium size networks. The problem, however, still remains, that is how reliable are the simulation results attained, compared to the performance of real world mobile communication systems. In large scale networks, the problem is getting even worse, since generic simulation systems are well adapted to medium scale problems but not to higher complexity systems, where the traffic conditions cannot be predicted. Therefore, some strategies should be involved to investigate the validity of the simulation models implemented in the proposed simulators. This paper, presents a validation framework for such simulation systems based on an architecture involving a hierarchy of several key components in order to manage this important issue. Some initial experimental results are compared favourably to the theoretical ones.

Key-words: Simulation Systems, Validation Framework, Bandwidth management, voice services, multimedia services, Channel allocation,

1 Introduction

Models and real experiments, to some degree, can only be approximations as having complete control over all of the factors is simply not fully achievable in any real system. This is a basic notion affecting work in simulation verification in general but, also, in mobile communication systems specifically, starting from the seminal research ideas of D.B. Johnson [1]. The basic directions introduced then, are:

1) The "trace emulation" approach, where a trace of the desired network's behaviour is generated using simulation, and then uses this trace to drive the standard trace modulation system in the operating system kernel of the real machines on the real network
2) The Initial approach to the validation of simulation work is to check the operation of the system according to a number of logical consistency checks. Although initial validation checks give considerable confidence in results, they do not actually fully validate the simulator results matching to the real world.
3) The approach involving a comparison based on the progression of some performance metric as a function of time is considered effective by comparing the results from simulations and real measurements (or emulations).
4) An alternative approach to validation could be to record a trace of all significant events (e.g., all packets sent, received, or forwarded) during the experiment in the real network, and to create a similar trace during a corresponding simulation run.

Such guidelines are considered in the majority of older as well as recent research for verification of simulation test beds in wireless communications as e.g [2]-[4].

The authors, in a series of papers [6]-[10] have presented a competent simulator for designing a large scale mobile communications system in order to efficiently manage bandwidth allocation per user demands. All the implemented algorithms for the proposed strategies and approaches (mainly event interleaving, multi-agent modelling, channel allocation) in those studies have been tested in a simulation setup that integrates the basic simulation and network components. The basic strategies implemented by the authors to solve the channel allocation problem in cellular communication systems are related to the ones presented in table 1, but different novel algorithms are involved in the proposed approach of [6]-[10] and the associated simulator. In table 1 one can see how these relevant research works tested the suggested strategies. However, there was no attempt then, to consider verification of the custom simulators involved at those times.

In this paper we consider verification of the simulation models of [6]-[10] based on an alternative of the strategies proposed in [1] and mentioned above. That is, instead of the similar Initial approach, we employ a component wise based verification of the specific basic implementation blocks, of the proposed simulation setup, in a hierarchical manner. The advantage of the proposed verification process is that there are no restrictions in simulating large scale systems, where the other directions at [1] are not practical in verifying such large scale communications. Instead, however, of checking logical consistency conditions we verify the operation of simulator's basic building components with regards to theoretical component performance (empirical component performance could be considered too)

The necessary validation of the proposed simulation setup consists of three different validation levels/components which are:

- Calendar Queue (CQ) (State of the art) scheduling mechanism implementation algorithm (section 2.1 below)
- Network environment which includes signal propagation, interference and signal measurements (section 2.2 below)
- Network performance compared to theoretical computations (section 2.3 below)

The main objectives of this paper are:

- To show how the CQ algorithm has been developed in relation to the original pseudo code of (Brown, R. 1988) [11].
- To analyze how the network environment has been implemented and evaluated through the corresponding theoretical formulae and theories.
- To show that theoretical formulas for network performance (e.g. blocking probability) have been confirmed.

Similar studies in the literature that are focused on wireless network simulation and channel allocation are conducted with custom, free and commercial software platforms but very rarely consider the verification process mentioned herein.

Reference	**Study-Title**	**Simulation Platform / Results Compared with**
Cherriman, P., Romiti, F. and Hanzo, L. (1998) **http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.8618**	Channel Allocation for Third-generation Mobile Radio Systems	Custom/ Custom
Haas, H. (2000) **http:// www.era.lib.ed.ac.uk/bitstream/1842/430/1/thesis.pdf**	Interference analysis of and dynamic channel assignment algorithms in TD–DMA/TDD systems (PhD dissertation, University of Edinburgh)	Custom / theoretical background
Iraqi, Y. and Boutaba, R. (2000) **http://rboutaba.cs.uwaterloo.ca/Papers/Conferences/Archive/DSOM-00.pdf**	A Multi-agent System for Re-source Management in Wireless Mobile Multi-media Networks	Custom/ Custom

Table 1. Previous studies on simulation for channel allocation / resource management strategies, basic building blocks of the proposed by the authors simulator under verification

2 The Proposed Simulator Validation Framework Architecture

Figure 1 shows the logical structure of the proposed validation framework, according to the above mentioned concepts and research directions

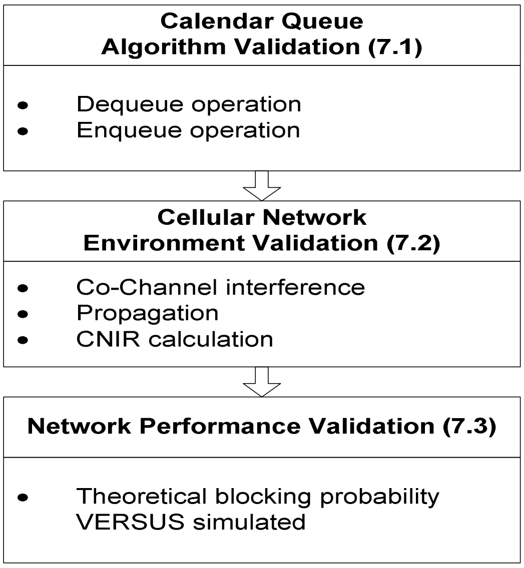

Fig. 1. Logical structure of the proposed validation framework

2.1 CQ Algorithm Validation

The CQ scheduler implementation within ns-2 is based on the proposed algorithms found in (Brown, R. 1988). These algorithms manage the generated events and also the internal functionality of the queue. The implementations of the CQ in this study are also based on the same algorithms. The CQ algorithms define two distinct operations which are:

- CQ operation (Dequeue, Enqueue)
- CQ internal functionality (creation, initialisation, resize)

Figure 2 shows how dequeue operation works according to the proposed pseudo code of (Brown, R. 1988) [11].

```
struct nodetype *dequeue( )

/* This removes the lowest priority node from the

queue and returns a pointer to the node containing it. */

{

register int i;

if (qsize == 0) return(NULL);

for (i = lastbucket; ; ) /* Search buckets */

{ /* Check bucket i */

if (bucket[i] != NULL && bucket[i] → prio < buckettop)

{ /* Item to dequeue has been found. */

Remove item from list;

/ * Update position on calendar. */

lastbucket = i; lastprio = priority of item removed;

--qsize;

/* Halve calendar size if needed. */

if (qsize < bot_threshold) resize(nbuckets/g);

return item found:

}

else

{ /* Prepare to check next bucket or else go to a direct search. */

++i; if(i == nbuckets) i = 0;

buckettop += width;

if(i == lastbucket) break; /* Go to direct search */

}

/* Directly search for minimum priority event. */

Find lowest priority by examining first event of each bucket;

Set lastbucket, lastprio, and buckettop for this event;

return(dequeue( )); / * Resume search at minnode. */

}
```

Fig. 2 C-pseudo code, Dequeue operation (Brown, R. 1988) [11]

The whole dequeue algorithm in figure 2 can be described in a few steps as follows:

(a) Check if the queue contains events

(b) Start to scan all the buckets

(c) If an event with higher priority is found, dequeue it, update variables and queue size, otherwise search in the next available bucket

(d) Return the located event

Initially, the size of the queue is detected. If the queue is empty (without events) then there is nothing to dequeue. But if the queue contains events, the program starts to scan all the bucket tops to find an event with higher priority (lower time stamp). If an event for dequeue is found, then its bucket and priority are stored and the queue size decreases. Based on the remaining queue size, a decision for resizing is taken. If no event is found in current bucket, then the next bucket is scanned. Finally, the located event with higher priority is returned.

In figure 2, *lastbucket* represents the bucket of the last dequeued event, *buckettop* is the corresponding priority at the top of the bucket and *lastprio* is the priority of the last dequeued event.

In the CQ mechanism within the evaluated simulation model, the implementation (Java-pseudo code) of (Brown, R. 1988) [11] is as follows (fig. 3):

```
public tevent[] dequeue()

/* procedure: (a) remove event, (b) return location */

{

  int i;

  if (qsize() == 0) return null;

  for (i = lastbucket; ; ) /* Search buckets */

  { /* Check bucket i */

    if (bucket[i] != null && bucket[i].prio < buckettop)

        {

        Remove item from list;

        lastbucket = i; lastprio = priority of item removed;

        --qsize;

        /* Halve calendar size if needed. */

        if (qsize() < bot_threshold) resize((nbuckets/g));

        return item found:

        }

else

        {

        /* Prepare to check next bucket or else go to a direct search. */

        ++i; if(i == nbuckets) i = 0;

        buckettop += width;

        if(i == lastbucket) break;

        }

/* Directly search for minimum priority event. */

Find lowest priority by examining first event of each bucket;

Set lastbucket, lastprio, and buckettop for this event;

return(dequeue( )); / * Resume search at minnode. */

}
```

Fig. 3 Java-pseudo code, Dequeue operation in the evaluated model

Figure 4 shows the corresponding pseudo code for the enqueue operation (Brown, R. 1988) [11].

```
/* This adds one entry to the queue. */

{

int i;

/* Calculate the number of the bucket in which to
```

```
place the new entry. */

5 i = priority/width; / * Find virtual bucket. */

6 i = i % nbuckets; /* Find actual bucket. */

7 Insert entry into bucket i in sorted list;

8 ++qsize; / * Update record of queue size. */

9 /* Double the calendar size if needed. */

10 if (qsize > top-threshold) resize(2 * nbuckets);

11 }
```

Fig. 4 C-pseudo code, Enqueue operation (Brown, R. 1988) [11]

Lines 5 and 6 of figure 2, are the implementation of basic CQ equations [11]. In line 5 of figure 4, the fraction t(e)/δ is calculated (see CQ equations in [11]). Finally, the number of bucket to store the new generated event is calculated (t(e)/δ) mod M (see CQ equations in [11]). It is obvious from figure 4 that the variable priority represents the time stamp of the generated event.

The corresponding Java-pseudo code for the enqueue operation is as shown in figure 5.

```
1 /* This adds one entry to the queue. */

2 {

3 int i;

4 /* Calculate the number of the bucket in which to
place the new entry. */

5 i = Math.floor(priority/width); / * Find virtual
bucket. */

6 i = i % nbuckets; /* Find actual bucket. */

7 Insert entry into bucket i in sorted list;

8 ++qsize; / * Update record of queue size. */

9 /* Double the calendar size if needed. */

10 if (qsize > top_threshold) resize(2 * nbuckets);

11 }
```

Fig. 5 Java-pseudo code, Enqueue operation in the evaluated model

The corresponding implementations of the (Brown, R. 1988) [11] pseudo code within ns-2 (in C) can be found in (http://www-rp.lip6.fr/ns-doc/ns226-doc/html/index.htm).

2.2 Cellular Network Environment Validation

The proposed models are evaluated through an implemented wireless network environment. This environment has been built on the known theoretical components for radio propagation, signal measurements and cellular network operation. The presented simulation results in this study derive from the proposed models that are implemented in the above wireless network environment. Thus, the validation of this environment is necessary in order to prove the correctness of the results. The validation procedure can be found also in (Haas, H. 2000,**http://www.era.lib.ed.ac.uk/bitstream/ 1842/430/1/thesis.pdf**) and consists of two phases which are:

- Monte Carlo simulations
- pdf evaluation based on theoretical solutions

The transmitted signal suffers from the two most important factors within the wireless environment which are path loss (distance attenuation) and shadowing (obstacles in the signal path). On the other hand, the CNR between BS and MU and the total interference from other co-channel MUs affect the received signal quality (fig. 6).

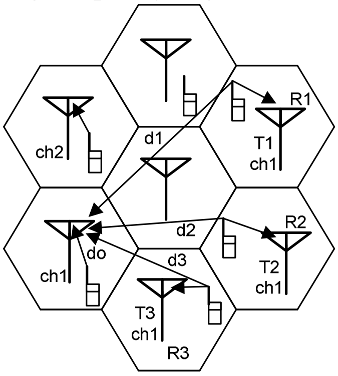

Fig. 6 Co-Channel Interference

The CNIR for the MU T_0 can be derived from the following formula:

$$R_{cni} = \frac{AP_0 d_0^{-a} 10^{\frac{\xi_0}{10}}}{N + \sum_{i-1}^{3} AP_i d_i^{-a} 10^{\frac{\xi_i}{10}}} \quad (1)$$

where A is a proportional coefficient, P_i is the transmitted power of the Mobile Unit (MU) T_i, d_i is the distance between MU T_i and BS R_0, ξ_i is the shadowing distortion between T_i and R_0 and α is a path loss factor. The average received signal strength decays as a power low of the distance transmitter-receiver. Path loss decays the transmitted signal with a factor α (eq. 1). For an urban area α=3.5 (Rappaport, T.S. 2002) [5].

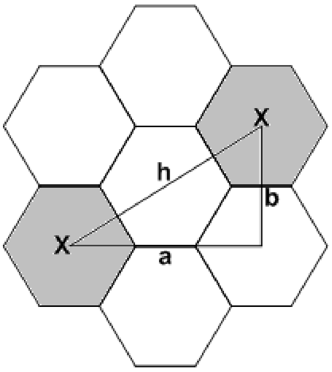

Fig. 7 Distance calculation

Initially the distance

$$h=\sqrt{\left(x_2-x_1\right)^2+\left(y_2-y_1\right)^2} \tag{2}$$

is calculated (fig. 7). The path loss factor between these points is $h^{-\alpha}$. The shadowing is subject to log-normal distribution with σ as standard deviation. Shadowing corresponds to ξ_i of equation 7.1. The shadow attenuation (Rappaport, T.S. 2002; Lee, W.C.Y. 1995) [5] is obtained as follows:

$$sh=10^{\frac{\sigma\cdot n}{10}} \tag{3}$$

where σ is the standard deviation of shadowing and n is a number from the normal distribution. Using the shadow attenuation and distance between MU and BS, the distance attenuation dw can be derived. The CNR is calculated between MU and BS (Rappaport, T.S. 2002; Lee, W.C.Y. 1995) [5]

$$cn=10^{\frac{cnedge}{10}}\cdot dw \tag{4}$$

Now, the CNIR can be calculated as:

$$CNIR=\frac{C}{N+I}=\frac{1}{\frac{N}{C}+\frac{I}{C}}=\frac{1}{\left(\frac{C}{N}\right)^{-1}+\left(\frac{C}{I}\right)^{-1}} \tag{5}$$

The CNIR can be calculated by knowing the ratios C/N and C/I. The ratio C/N is already known from (4) and the C/I is determined from the ratio *dw/uw*, where *uw* represents the total signal attenuation caused by other co-channel MUs.

2.3 Network Performance Validation

The most known statistical metrics for the wireless network performance evaluation are blocking and dropping probabilities. Blocking probability represents the blocked calls, while dropping represents the unsuccessful channel reallocation for an ongoing call. The dropping probability is strongly connected to R_{cni} (eq. 1), because when this ratio is not above the accepted threshold and the network cannot allocate an appropriate channel, the call is dropped. On the other hand, blocking probability can be theoretically calculated. If the received power of each MU is high enough, it is assumed that the interference from other MUs can be ignored. The theoretical formula is as follows:

$$P_{blocking\_theoretical}=\frac{\binom{n-1}{s}\left(vh\right)^s}{\sum_{i=0}^{s}\binom{n-1}{i}\left(vh\right)^i} \tag{6}$$

where **n** is the number of users, **s** is the number of channels, **v** is the average call arrival rate (for no connected MU) and **h** is the average call holding time.

Equation 6 shows only the basic relation between channels and users and does not take into consideration critical factors that affect the blocking probability such as traffic conditions, service type, channel allocation strategy, etc. Figure 8 shows the theoretical blocking probability that derives from eq. 6 as compared to simulated blocking probability. The simulated probability has been generated from the large scale network based on the HDCA and network services [6]-[10].

3 Conclusion and Prospects

For the evaluation of algorithms and architectures for GSM/3G/4G mobile communications, a specialized simulation setup has been developed by the authors through the years. We herein investigate an approach to verify such simulation models for large scale networks. Three different levels have been suggested for validating the simulation setup: (a) scheduling mechanism verification, (b) network environment verification and (c) network performance verification. Initial experimental and analytical results on these component-wise validation levels indicate the correctness of the implemented models and the suggested verification approach.

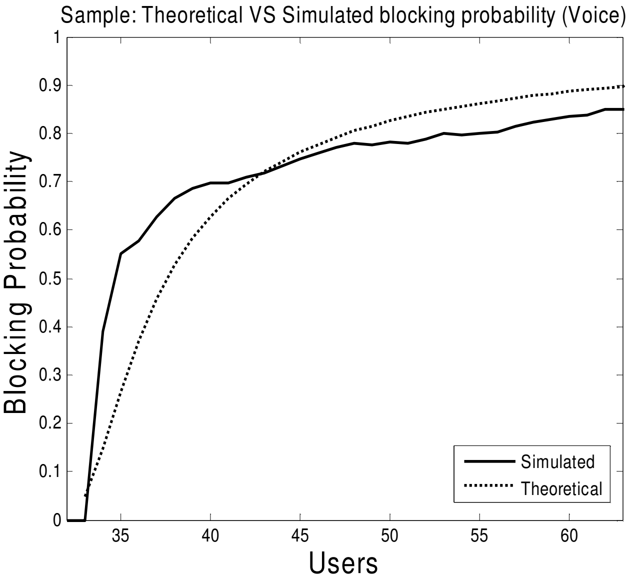

Fig. 8 Theoretical blocking probability versus simulated for the HDCA algorithm implemented in the proposed simulator

References:

[1] David B. Johnson, “Validation of Wireless and Mobile Network Models and Simulation”, *In DARPA/NIST Network Simulation Validation Workshop,* 1999, http://www.cs.rice.edu/~dbj/pubs/darpa99-validation.pdf

[2] Erik Nordström, Per Gunningberg, Christian Rohner, Oskar Wibling, “Evaluating wireless multi-hop networks using a combination of simulation, emulation, and real world experiments”, *in MobiEval '07 Proceedings of the 1st international workshop on System evaluation for mobile platforms,* Pages 29 – 34, ACM New York, NY, USA, 2007

[3] G. Wittenburg and J. Schiller, "A quantitative evaluation of the simulation accuracy of wireless sensor networks," in *Proceedings of the 6. Fachgesprch "Drahtlose Sensornetze" der GI/ITG-Fachgruppe "Kommunikation und Verteilte Systeme"*, pp. 23-26, Aachen, Germany, July 2007.

[4] G. P. Halkes and K. G. Langendoen, "Experimental evaluation of simulation abstractions for wireless sensor network mac protocols," in *Proceedings of the 14th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD '09)*, Pisa, Italy, June 2009.

[5] Rappaport, T.S., “*Wireless Communications Principles and Practice”*, Prentice Hall, 2002

[6] Papazoglou, P.M., Karras, D.A., Papademetriou, R.C, “Modeling efficient dynamic channel assignment schemes for quality of service wireless multimedia telecommunications through an extended and integrated Java simulation framework”, *1st International Conference on Future Generation Communication Technologies, FGCT 2012; London; United Kingdom; 12 December 2012 through 14 December 2012; Code 96561*

[7] Papazoglou, P.M., Karras, D.A., Papademetriou, R.C, “On Multi Agent based modeling and control in large scale wireless communication systems for improved resource allocation performance”, ***Telecommunication Systems***, 2011, Pages 1-19.

[8] Papazoglou, P.M., Karras, D.A., Papademetriou, R.C, “New strategies based on multithreading methodology in implementing ant colony optimization schemes for improving resource management in large scale wireless communication systems”, ***Studies in Computational Intelligence***, Volume 299, 2010, Pages 537-578.

[9] Papazoglou, P.M., Karras, D.A., Papademetriou, R.C, “On the efficient implementation of a high performance multi-agent simulation system for modeling cellular communications involving a novel event scheduling algorithm”, ***International Journal of Simulation: Systems, Science and Technology***, Volume 10, Issue 1, January 2009, Pages 42-52

[10] Papazoglou, P.M., Karras, D.A., Papademetriou, R.C, “On a new generation of event scheduling algorithms and evaluation techniques for efficient simulation modelling of large scale cellular networks bandwidth management based on multitasking theory”, ***WSEAS Transactions on Communications***, Volume 7, Issue 10, 2008, Pages 1024-1034

[11] Brown, R., “Calendar queues: A fast O (1) priority queue implementation for the simulation event set problem”, *Communications of the ACM, 31*(10), 1220–1227, 1988.