Parallel Discrete Event Simulation of Radio Resource Management in Personal Communication Networks

Michael Liljenstam

Department of Teleinformatics, Computer Systems Laboratory
Royal Institute of Technology (KTH)
S-164 40 Kista, Sweden
Email: mili@it.kth.se, Fax: +46-8-7511793

Abstract

Simulation is widely used in the evaluation and development of Radio Resource Management schemes for Personal Communication Networks. However, limitations are often encountered due to high resource demands in terms of cpu-time and memory usage. Parallel and distributed execution techniques have been proposed as a possible solution to these problems. In this paper an experimental simulation environment, called MobSim++, is presented that employs parallel simulation techniques to reduce execution times and make more memory resources available to the user. The environment is designed to work on both shared memory and distributed memory platforms and present a common interface to the user. Experimental results indicate the potential for significant speed-up depending on the hardware platform and certain properties of the model being studied.

1. Introduction

Personal Communication Networks (PCNs) provide wireless communication services to mobile users. Tremendous growth in the number of subscribers in cellular telephony networks over the last couple of years increases demands on network operators and manufacturers to efficiently utilize radio resources while maintaining acceptable quality of service. Hence, tools for the study of such systems are essential to the industry and telecommunications operators. Tractable analytical models of PCNs are often not general or flexible enough for these studies and simulation has consequently become widespread. However, detailed simulations of such networks are often highly resource demanding in terms of cpu time and memory requirements.

We explore the possibility of tackling the large resource demands in simulations of cellular networks through the use of parallel discrete event simulation (PDES) techniques. The advantage of PDES compared to other techniques is that model fidelity is not compromised and the statistical properties of the output are not altered. Up to now our primary focus has been on applying PDES techniques to simulations of Radio Resource Management in wireless cellular networks.

The remainder of this paper is organized as follows: Section 2 introduces some basic concepts of PCNs and defines the problem being studied. Section 3 gives an introduction to the fundamentals of PDES. Related work in the area of parallel simulation of PCNs is discussed in Section 4, and the simulation model under consideration here is described in Section 5. The MobSim++ experimental platform is described in Section 6, and some experimental results on speed-up are presented in Section 7. Section 8, finally, gives conclusions and some mention of areas for future work.

2. Personal Communication Networks

The area covered by a Personal Communication Network is populated by a number of *base stations* (BSs), where the radio coverage of a BS is called a *cell*. Users carrying a *mobile terminal* (MT) can communicate with other parts of the system by transceiving with the BS covering the cell they are located in at that specific time. Available radio frequency spectrum is limited and it is important to utilize it efficiently. For this reason, radio channels are *reused* in cells that are sufficiently spaced apart, since interference caused by transmission (on the same channel) will be negligible if the station is sufficiently far away.

Different schemes exist to allocate channels to cells so as to ensure that interference falls below the limit of what is tolerable. In Fixed Channel Assignment (FCA) K cells are grouped into a cluster and a disjoint subset of the available channels are assigned to each cell in the cluster. Dynamic Channel Assignment (DCA) schemes strive to improve channel utilization compared to FCA by dynamically reassigning channels among cells based on the current state of the system in terms of e.g. traffic load or measured interference.

As MTs move around it may become necessary to switch an ongoing call over to another BS or to use another channel on the same BS, this is referred to as a *handover* (HO). When moving closer to the BS, the interference caused to other stations can be reduced by reducing the transmission power, so called *power control*. An attempt to initiate a call may fail because there are no available channels in the cell, this is referred to as a *blocked call*. Similarly a HO may fail, due to unavailability of channels, which may result in poor reception for the MT for a prolonged time and eventually causing the call to be terminated. Such a forced termination is termed a *dropped call*. The fraction of blocked and dropped calls used as a quality of service measure for the network. The goal is to engineer the system to make best possible use of the available radio frequency spectrum while maintaining low probabilities of blocked or dropped calls (typically < 1%).

The problem of Radio Resouce Management (RRM) [15], which we are targeting here, is that of selecting for each ongoing call:

- base station
- radio wave form (channel in this case)
- transmission power

so as to maximize the carried traffic while maintaining the Quality of Service constraints.

3. Parallel Discrete Event Simulation

Parallel discrete event simulation (PDES) refers to the execution of a single Discrete Event Simulation (DES) program on a parallel computer. The common approach to PDES is to view the system being modeled, usually referred to as the physical system, as a set of sub-systems, each simulated by a logical process (LP). Thus, a parallel simulator is constructed as a set of N (N>1) communicating LPs. Since time advances asynchronously in the different LPs, special care must be taken to ensure that events are executed in time order (i.e. that causality errors are avoided). Two main paradigms have been proposed for synchronization of LPs in a PDES: *conservative* and *optimistic* methods. In conservative approaches, only *safe* events (i.e., events that do not create causality errors) are executed. In this way, conservative methods strictly avoid any causality errors ever occurring. Optimistic approaches, on the other hand, assume that respective events at different LPs occur in correct order. However, optimistic methods employ a detect and recovery mechanism to correct causality errors, if they occur. The Time Warp mechanism [9] is the most well-known optimistic protocol. The Time Warp approach allows an LP to simulate forward, in parallel with other LPs, until it receives an event message that causes *rollback* to an earlier simulated time. The event causing rollback is called a *straggler*. The method requires both time and space to maintain the past history of each LP, and to perform the rollback operation whenever necessary. State saving is an important issue in Time Warp and several optimizations, including incremental and sparse state saving, have been proposed. Time Warp has shown good performance for many applications, where the associated state saving and rollback costs lie within an acceptable range.

An excellent introduction to PDES can be found in [4]. Applications of PDES to different fields are surveyed in [6] and [5], comprising areas as diverse as queueing networks, digital logic circuits, computer architectures, telecommunication networks, transportation, manufacturing, and combat models.

4. Related Work

Early work in the field of parallel simulation of PCNs considered simplified models where the network is considered to consist of homogeneous cells. Typically in the form of a tesselation of hexagonal or quadratic cells, as shown in Figure 1. The models then considered the changes of resource demands as users entered the system, moved from cell to cell, and departed. This type of nearest neighbor communication pattern provides good opportunity for a straight-forward decomposition of the model in the geographic plane so that one or several cells constitute an LP. (One or more LPs are then mapped to each processor.) Good parallel performance was reported in cases where interprocessor communication could be limited in this way [3][12]. In other cases, where communication patterns were different, performance was seen to depend heavily on the range of interaction in the model [7].

This level of abstraction is sufficient for certain classes of problems relating to RRM. However, in reality radio coverage is irregular (Figure 2) and more detailed models are often needed to study other aspects and a larger class of RRM algorithms.

Later studies [13][14], including ours, are aimed at parallelizing more detailed models that permit taking radio propagation issues into account. Taking more detailed information of interfering stations into account tends to increase the range of interaction between objects in the model. Thus, model decomposition and partitioning poses more of a challenge in this case.

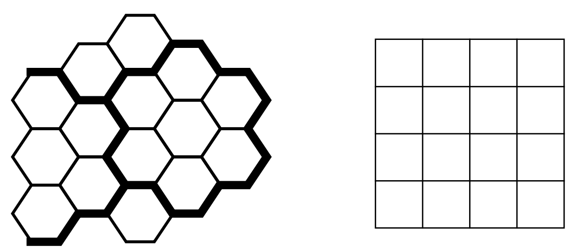

Figure 1. Examples of hexagonal mesh and quadratic mesh cell layouts.

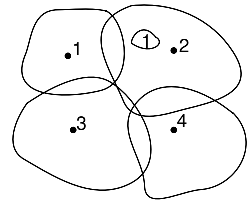

Figure 2. Cell coverage is irregular due to obstacles in the terrain, resulting in partially overlapping cells and possibly uncovered areas.

5. The PCN Model

A model that includes signal propagation issues has been designed and implemented. The model originated from an existing sequential simulator called MaDRAS developed at the Radio Communication Systems Laboratory, KTH in cooperation with Winlab at Rutgers University. The MaDRAS model has been used in e.g. [1].

As shown in Figure 3, the simulator consists of three replaceable parts (submodels) in a manner similar to the MaDRAS simulator: The propagation model, the mobility model, and the teletraffic model.

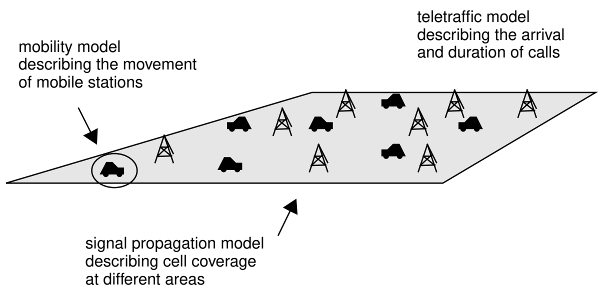

Figure 3. The three submodels of the PCN.

The teletraffic model describes the arrival process of new calls and their duration. Call arrivals form a Poisson process with a mean arrival rate λ, and exponential call duration with mean $1/\mu$.

The mobility model describes the movements of the mobile terminals (MTs). We have used a simple model where each MT moves with constant speed in one of four directions: north, east, west, or south (NEWS), and a MT only exist for the duration of a call. When a call arrives a new MT is created, it is placed at a uniformly distributed position x_m,y_m in the area and it is assigned a truncated Gaussian distributed speed v_m and a uniformly distributed initial direction. When the position is updated for an MT it will make a decision whether or not to turn. With a probability of p_{turn} it will turn left and with the same probability it will turn right.

The propagation model describes propagation path loss using a statistical model for correlated log-normal shadow fading. The path losses at a specific position are regarded as constant over time, so that at a position p the gain from a transmitting station s is $G_{s,p}$ forming a two-dimensional matrix for each base station that describes the geography.

Link gain is modeled in a similar way as in [1]. The link gain, $G_{ij}(d_{ij})$, between transmitter j and receiver i is modeled as

$$G_{ij}(d_{ij}) = \frac{S_{ij}(d_{ij})}{d_{ij}^4}$$

where d_{ij} is the distance between the transmitter and the receiver, and where the shadow fading component $S_{ij}(d_{ij})$ is modeled to be log-normally distributed. The fading process is assumed to have the exponential correlation function proposed in [8], which is derived from experimental results. Let $U_{ij}(d_{ij}) = 10*\log[S_{ij}(d_{ij})]$, then we have

$$E\{U_{ij}(d_{ij})U_{ij}(d_{ij}+x)\} = \sigma^2 \cdot \exp(-|x|/D)$$

where σ^2 is the log-variance, and the parameter D determines how fast the correlation decreases with distance. This is used to determine the correlation horizontally and vertically in the gain matrix.

Link quality is measured as the Signal to Interference Ratio (SIR). The downlink SIR γ_m for MT_m for a certain channel k is defined as:

$$\gamma_m = \frac{G_{m,i} \cdot P_i}{\sum_I G_{m,j} \cdot P_j + \eta_t}$$

where P_i is the transmitted power from the BS_i (the BS that MT_m is connected to), $G_{m,i}$ is the gain (path loss) from BS_i to MT m (at its current position), S is the set of base stations such that $I=\{j \neq i: BS_j \text{ is transmitting on channel } k\}$, and η_t is the thermal noise (constant). $G_{m,i}$ and $G_{m,j}$ depend on the terrain, but are regarded as constant for a certain position in the terrain throughout the simulation. The uplink SIR is defined analogously. It is assumed here that the radio channels are orthogonal, so adjacent channel interference has been neglected.

The currently running version uses Fixed Channel Assignment (FCA), but the model itself will support e.g. interference adaptive Dynamic Channel Assignment schemes with only modest changes. Similarly, some other features have been omitted from the model at this point but are planned to be included in the future: e.g. transmission power control and a link quality threshold for channel reassignment (for intracell handovers).

Each mobile station performs the following procedure, resource reallocation assessment (RRA), periodically (with a period of t_{rra} seconds):

1. Calculate downlink and uplink Signal-to-Interference Ratio (SIR) γ for the channel currently used. If SIR on either the up- or downlink is *less than* γ_{drop} *for* t_{outage} *consecutive seconds* the call is dropped.
2. Find the best BS, by calculating the received power of every other BS and compare it to the received power of the presently selected BS. If the difference is greater than the handover hysteresis threshold a new BS will be selected. A handover to the new BS will be attempted. If the handover fails due to lack of free channels, however, the MT will stay on its current connection.
(The MT measures the received power on a special BS "pilot"-channel. The transmitted power on this channel remains constant throughout the simulation.)

Channel selection is done stochastically with equal probability among the available free channels on the BS. The uplink and downlink channels are changed at the same time and hence treated as one unit. Path losses are regarded as equal in both directions due to reciprocity.

The positions of the mobiles are updated every time there is new data to be read from the gain matrix, i.e. each mobile schedules a position update event for itself to occur when it has travelled the distance between two samples in the gain matrix.

6. The MobSim++ Environment

The Mobile Telecommunication Simulator has been implemented in C++ to run on top of the Parallel Simulation Kernel (PSK). An object oriented design approach was used for the simulator to make it flexible and to allow easy replacement of different submodels. A separate visualization tool written in C++ and TCL/TK is also provided.

The Parallel Simulation Kernel (PSK) was designed to be a common platform for research on parallel simulation applied to different areas (such as mobile telecommunications, computer architecture, and hardware simulations). The kernel handles synchronization on shared memory multiprocessors (such as SUN SparcCenter 2000) or in a distributed memory environment (such as IBM SP) using a common interface. The kernel was implemented in C++ to provide support for object oriented programming. The user creates a simulation application by creating an instance of the kernel and deriving objects from some of the kernel defined classes.

Due to poor look-ahead in the PCN application and because optimistic synchronization mechanisms tend to offer a higher level of transparency for the user, it was decided that the kernel should employ optimistic synchronization by using the Time Warp mechanism.

The overall structure and different layers of a simulator, in this case the mobile telecommunications simulator, implemented on top of the PSK are shown in Figure 4. The PSK was designed to be portable across a number of platforms and to have a modular structure to allow easy replacement and experimentation with key components of the system. The kernel is internally made up of a parallel simulation mechanism (in this case Time Warp) and libraries for additional functionalities such as random number generation and visualization support that have some ties to the simulation mechanism. The random number generator uses separate random number streams for each of the LPs. The Time Warp mechanism includes modules for key components of the system. Event set management, state saving mechanisms and Logical Process scheduling policies were determined to be mechanisms that may require modification to adapt to specific application requirements. Also, there is a low level memory management module that is common to all different parts of the Time Warp kernel.

As mentioned previously, efficient state saving is known to be an important issue for optimistic synchronization. The PSK implement three different state saving methods: copy state saving (CSS), sparse (or periodic) state saving, and incremental state saving (ISS). Copy state saving saves the whole state of an LP after each event execution. The sparse state saving mechanism saves the state periodically and reconstructs the intermediate states when they are needed. An adaptive algorithm is used that tries to find an optimal state saving interval. The incremental state saving mechanism saves only the changes made to the state which is more efficient if generally only a small fraction of the state is updated.

In order to make applications portable, we have based the kernel on standards for concurrency control and message passing. On shared memory machines, we use the concurrency control primitives (locks, barriers, etc.) defined in the shared memory part of the Portable Programs for Parallel Processors (P4) macro package [2]. This package has been ported to a wide range of shared memory multiprocessors. In the case of distributed memory environments we use an implementation of the Message Passing Interface (MPI) [16].

Our experiences to date, in the design and implementation of the simulation environment have been reported in [11].

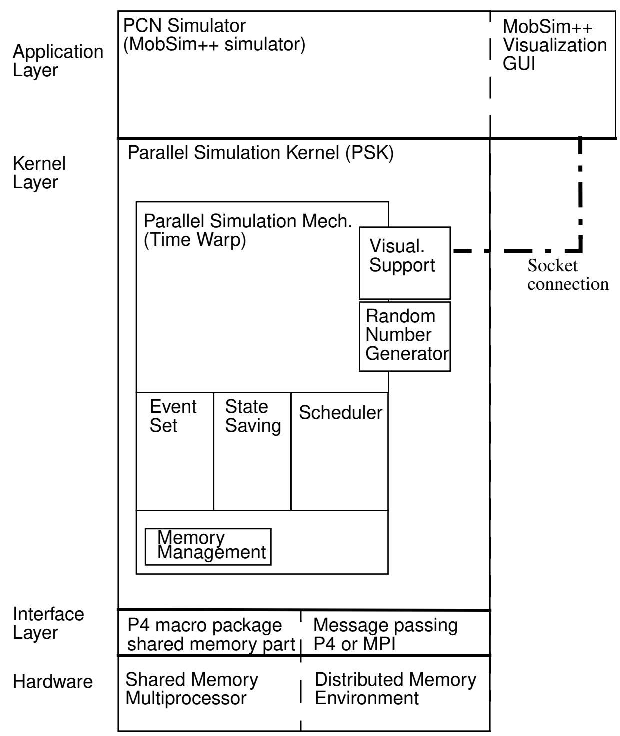

Figure 4. Structure of the Mobile Telecommunication Simulation Environment

7. Experimental Results

In its current shape it models an FCA system based on Frequency Division Multiple Access and/or Time Division Multiple Access (F/TDMA). For the purposes of studying partitioning and synchronization, it exhibits behavior similar to other schemes such as interference adaptive DCA.

In the experiments described here a hexagonal mesh BS layout (2 km cell diameter) with a cluster size of 9 was simulated. The average call length $1/\mu = 120\,s$ and unless stated otherwise the call arrival intensity to the system $\lambda = 1\ call/s$. The time interval between resource reallocations t_{rra} was set to 0.5 seconds to put it close to the interval between measurements in the GSM system which is 480 ms. The average mobile speed v_m was set to 90 km/h unless stated otherwise and the turning probability $p_{turn} = 0.01$. In this model we have used a resolution of 50 m in the gain matrix, a σ of 6 dB, and a D of 100 m. Furthermore, settings of $\gamma_{drop} = 8\ dB$, and $t_{outage} = 5$ seconds, and the run time was 2000 seconds of simulated time (unless otherwise indicated). The experiments were conducted on a SUN SparcCenter 2000 shared memory machine with 8 processors and 512 Megabytes of memory.

As has already been mentioned, partitioning is an important issue for this type of model to achieve good parallel performance. All stations in the system transmitting on a certain channel are included in the SIR calculations which are carried out in each RRA event. Thus, there will be frequent interaction between entities geographically spread throughout the system. However, since orthogonal channels are assumed (only stations transmitting on the same channel are included), interactions are local in the frequency spectrum. Figure 5 depicts a view of the state space of the model, spanning the geographical area and the frequency spectrum. (The change in cell shape is irrelevant.) Rather than decomposing the system geographically, as done in early studies, the system is decomposed along the frequency axis. This limits communication in the model since the frequently occurring SIR calculations are local in spectrum but not geographically. Thus, each radio channel constitutes a logical process (LP) and all stations that use the channel are part of the LP. (BSs are actually partitioned as they use multiple channels.) Hence, MTs are passed between the channel LPs as they change to a new radio channel but no message communication is necessary to perform the SIR calculations.

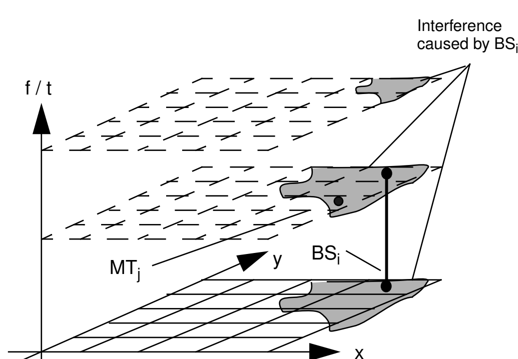

Figure 5. Simulation space for radio signal calculations, assuming placement of BSs according to quadratic cell layout. The x- and y-axes span the area covered by the system, and the f/t-axis spans the frequency/time spectrum.

Limiting the communication in the model is one important issue for succeeding with the parallelization. Another known issue is that of computation grain. There should be a sufficient amount of computation work in the events so that synchronization overheads do not become dominating. This is demonstrated in the first set of experiments, results shown in Figure 6. Here, speed-up (i.e. execution time reduction) compared to execution on a sequential simulation kernel is shown as the size of the simulated system is varied. The dominating work in the simulation is the SIR computations and the work in each such computation is proportional to the system size. For a small system of 36 cells (2 by 2 clusters) the computation grain is too small compared to the synchronization overheads to result in good performance. For a larger system of 81 cells (3 by 3 clusters) speedup improves as the computation grain increases. Beyond that size there appears to be of no additional benefit. Some explaination for this is given by Figure 7, where execution efficiency (i.e. the fraction of executed events that are committed) is plotted. This indicates that more events are rolled back for the largest system (144 cells) than for the 81 cell system. However, due to the larger event grain, performance is still better than for the smallest system size.

We can also note that sparse state saving (SSS) appear to perform somewhat better than incremental state saving (ISS) for the 81 cell case, but worse for the largest system size. This is not surprising as, in this case, the LP state size is proportional to the system size being simulated and each event will update only one call. Thus, at some point when increasing size, saving only the state modifications will be more efficient than saving the whole state.

In these experiments the load (actually call arrival rate) was scaled to keep it proportional to system size. Hence, λ was set to 1, 2.25, and 4 calls/s for 36, 81, and 144 cells respectively. It should also be noted that the results presented here are preliminary in the sense that they are from a small number of replications (2-3) occasionally resulting in large variances, as seen in the graphs.

Running an 81 cell system for 1000 seconds varying the system load produced the results shown in Figure 8. Rollback behavior improves with increasing message population (more available work) in the model up to a point and then declines. Efficiency is shown in Figure 9. Consequently, there appears to be an optimum load in terms of parallel performance. However, the load here ranges from very low to very high (blocking probability in the order of 50%) and the performance differences are relatively small.

Figure 10, lastly, shows speed-up as the average velocity of the MTs is varied. A slight decrase of performance with higher mobility appears to be indicated (although not statistically significant in these experiments). Higher

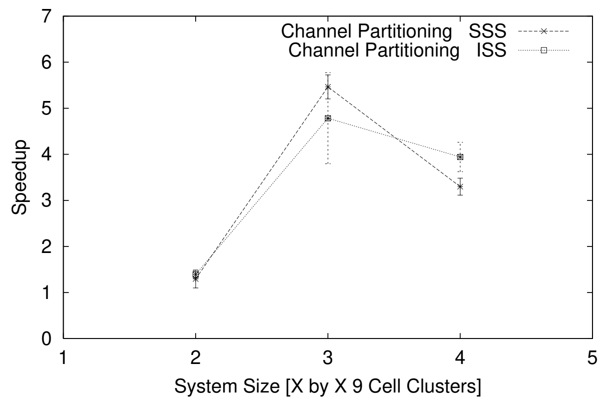

Figure 6. Speedup on 8 processors for varying system size (with 90% confidence intervals).

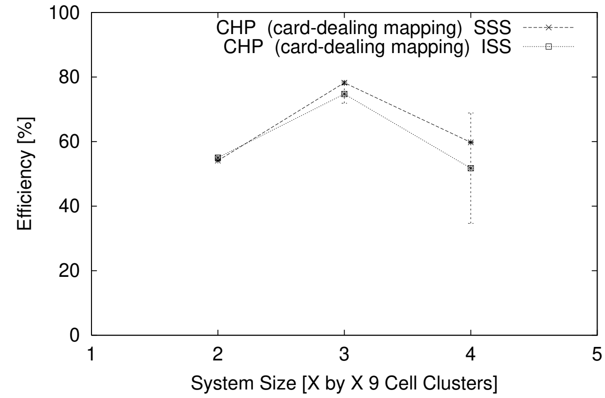

Figure 7. Execution efficiency, i.e. fraction of executed events that are eventually committed, for varying system size (with 90% confidence intervals).

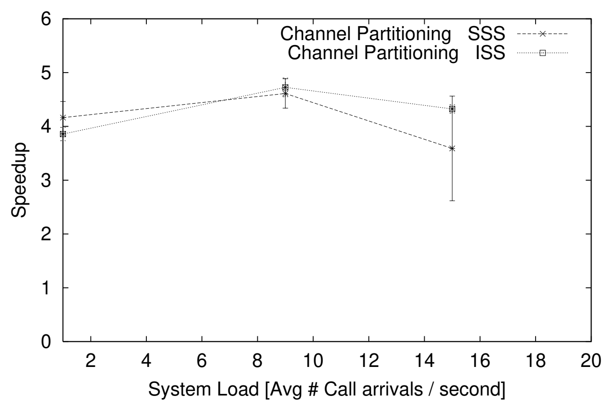

Figure 8. Speedup on 8 processors for varying system load (with 90% confidence intervals).

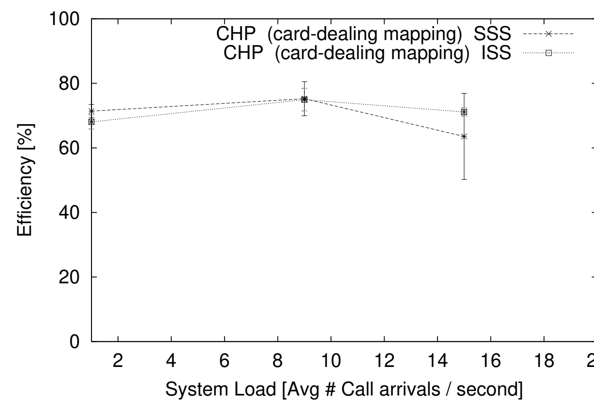

Figure 9. Speedup on 8 processors for varying system load (with 90% confidence intervals).

mobility will lead to more handovers between cells and thus more channel changes. Hence, message communication will increase somewhat with more mobility, leading to lower performance. However, again the performance difference is relatively small.

To summarize: Provided that there is sufficient event grain, i.e. that a large enough system is simulated, these results indicate the potential for significant speed-ups. For the parameters settings used in these experiments speed-ups ranged between 3.4 and 5.5, depending on system size, load, and user mobility.

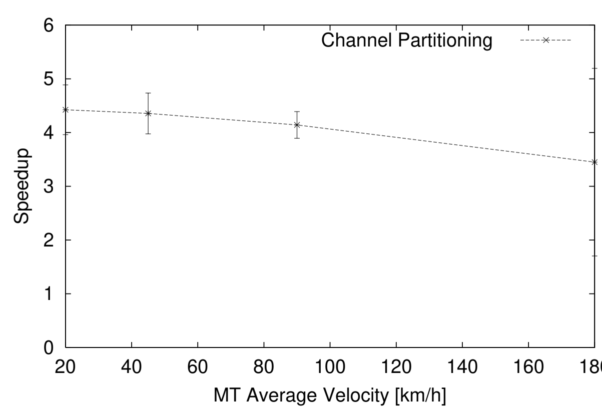

Figure 10. Speedup on 8 processors using ISS for varying mobility (mobile average velocity). (90% confidence intervals)

8. Conclusions

MobSim++, an experimental parallel simulation environment for the study of radio resource management in Personal Communication Networks has been presented. The simulator has an object-oriented design and is built on top of a portable simulation kernel that can run on both shared and distributed memory machines.

Experimental results indicate the potential for significant speed-ups compared to ordinary sequential execution. In the experiments presented here, speed-ups ranging between 3.4 and 5.5 (depending on model parameters) could be obtained provided that there is sufficient event grain.

Future work can be pursued along several different directions apart from further performance improvements: more functionality in the model such as DCA, power control etc., studies of traffic other than speech, and other propagation conditions (e.g. indoor environments). All of these are relevant to support the simulation of new system designs and determine their effect on the simulators parallel performance.

Acknowledgments

The author wishes to thank the Swedish Institute of Computer Science for kindly providing the use of the machine on which the experiments were performed. My gratitude naturally also to my project supervisors Rassul Ayani and Robert Rönngren without whom the project would not have come about, and to Johan Montagnat and Eva Frykevall who conributed to the PSK implementation.

References

[1] M. Andersin, M. Frodigh, K.-E. Sunell, "Distributed Radio Resource Allocation in Highway Microcellular Systems", *Fifth WINLAB Workshop on Third Generation Wireless Information Networks*, Rutgers University, New Jersey, 1995

[2] R. Butler, E. Lusk, "Monitors, messages, and clusters: the p4 parallel programming system", *Parallel Computing*, 20, April 1994

[3] C. Carothers, R. Fujimoto, Y.-B. Lin, "A Case Study in Simulating PCS Networks Using Time Warp", *9th Workshop on Parallel and Distributed Simulation,* Lake Placid, NY, -95

[4] R. Fujimoto, Parallel Discrete Event Simulation", *Communications of the ACM*, Vol. 33, No. 10, Oct. 1989, pp. 31 - 53.

[5] Fujimoto, R., "Parallel and Distributed Discrete Event Simulation: Algorithms and Applications", *in Proceedings of the Winter Simulation Conference*, 1993

[6] R. Fujimoto, D. Nicol, "State of the Art in Parallel Simulation", *in Proceedings of the Winter Simulation Conference*, 1992

[7] A. Greenberg, B. Lubachevsky, P. Wright, D. Nicol, "Efficient Massively Parallel Simulation of Dynamic Channel Assignment Schemes for Wireless Cellular Communications", *8th Workshop on Parallel and Distributed Simulation,* Edinburgh, Scotland, -94

[8] M. Gudmundson, "Correlation Model for Analyzing Handoff Algorithms", *IEEE Trans. Veh. Tech.*, vol. VT-42, pp. 351-356, Aug. -93

[9] D. Jefferson, "Virtual Time", *ACM Transactions on Programming Languages and Systems*, 1985, Vol. 7, No 3

[10] M. Liljenstam, R. Ayani, "A Model for Parallel Simulation of Mobile Telecommunication Systems", *in Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, San José, CA, Feb. 1996

[11] M. Liljenstam, R. Ayani, "MobSim++, An Environment for Parallel Simulation of PCNs", *in Proceedings of the World Congress on Systems Simulation (WCSS'97) (2nd Joint Conference of International Simulation Societies)*, Singapore, Sept 1997

[12] Malloy, B., Montroy, A., "A Parallel Distributed Simulation of a Large-Scale PCS Network: Keeping Secrets", *in Proceedings of the 1995 Winter Simulation Conference*, Arlington, VA

[13] Panchal, J., Kelly, O., Lai, J., Mandayam, N., Ogielski, A., Yates, R., "WiPPET, A Virtual Testbed for Parallel Simulations of Wireless Networks", *in Proceedings of the 12th Workshop on Parallel and Distributed Simulation*, Banff, Alberta, Canada, May 1998

[14] Perrone, F., Nicol, D., "Rapid Simulation of Wireless Systems", *in Proceedings of the 12th Workshop on Parallel and Distributed Simulation*, Banff, Alberta, Canada, May 1998

[15] J. Zander, "Radio Resource Management - An Overview", *in Proceedings of the IEEE Vehicular Technology Conference VTC 96*, Atlanta, April/May 1996.

[16] Message Passing Interface Forum, "MPI: A message passing interface standard", *Technical Report No. CS-94-230, University of Knoxville, Tennessee, , see also the International Journal of Supercomputing Applications, Volume 8, Number 3/4*, 1994.