An Efficient Simulation Environment for 3rd Generation Cellular Networks

A. G. Valkó, A. Rácz
Ericsson Traffic Laboratory
[ethagv,tmparz]@lt.eth.ericsson.se

G. Fodor, L. Westberg
Ericsson Radio Systems
[erafodo,eralgw]@era-t.ericsson.se

Abstract

Together with measurements and analytical methods, the simulation-based evaluation of cellular systems will be increasingly important as the deployment of new mobile applications imposes new requirements both on the radio interface and on the fixed network infrastructure. Efficient allocation of the network's resources must be based on reliable and flexible performance evaluation techniques. In this paper we describe a simulation environment optimized for the performance analysis of wideband cellular networks. To handle the complexity of the system without losing low-level details due to a high-level abstraction, a hierarchical simulation structure is developed which is also largely based on the integration of analytical evaluations' results into the simulation. The resulting structure can surprisingly efficiently (both in terms of simulation run time and in terms of modeling flexibility and speed) simulate large and complex systems while the level of abstraction can be freely selected in a wide range by the user. For instance, in case studies we find that simulation times of ATM based cellular networks can be an order of a magnitude less than using most of the readily available simulators. Though the simulation environment described here is specific to ATM/AAL2 based mobile networks, the proposed concept is more widely applicable to accelerate simulations.

Keywords

hierarchical simulation, ATM, AAL2, mobile systems

1 INTRODUCTION

While the penetration of cellular mobile phones increases rapidly and may soon catch up with the ordinary telephone penetration, there is already a

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35353-1_28

D. Kouvatsos (ed.), *Performance Analysis of ATM Networks*

clear trend that goes beyond this quantitative change. The appearance of new applications with higher bit rate and diverse GoS and QoS requirements imposes new requirements both on the radio interface and on the fixed network infrastructure.

In this environment the efficient use of the network's resources will be increasingly important. While peak allocation and eventually overprovisioning may be adequate in today's single application mobile networks, in the future mobile operators will need to improve efficiency as high as possible. To do this, operators might need to apply variable bit rate coding of traffic, use ATM as transport network infrastructure (Eneroth 1997) and optimize resource management. The key to efficiently exploiting this more complex system might be to develop an accurate way to analyze its performance - if possible, before actually building it.

For any telecommunication system, performance analysis can rely on the following approaches, exhaustively described by (Kurose 1988):

- analytical methods,
- simulation,
- measurement and prototyping.

Measurements and prototyping usually provides the most precise and reliable information but the use of this technique is often very expensive, time-consuming and inflexible. *Analytical evaluation* methods give a larger freedom in varying the investigated system's parameters but their applicability is restricted by the need to find an analytically tractable model. With *simulation* techniques the level of abstraction can be freely determined though it affects largely the required processing capacity and the accuracy. As none of the three approaches provide an ideal solution in all situations, the analysis of a complex system must use a combination of these.

In this paper we describe an extension of the PLASMA ATM simulator, first described by (Haraszti 1995), which makes it capable to efficiently simulate ATM-based wideband cellular networks. We argue that in order to meet the above listed requirements a new simulation technique needs to be considered. The proposed hybrid hierarchical simulation environment is designed specifically for the performance analysis of systems where the complexity requires a combination of analytical techniques. The simulation is largely based on the integration of analytical results in the simulation which together with the hierarchical structure makes it capable of simulating a large and complex network without hiding the bit-level details or radio-related features behind a high-level abstraction.

After a brief introduction to the proposed concept, the applied model and the simulator's architecture will be described. Two simulation examples will also be provided to illustrate the simulator's capabilities. The examples are taken from the analysis of the new ATM Adaptation Layer No.2

where standardization activity was based on detailed performance evaluations (Eneroth 1997). The new adaptation layer provides high efficiency and low delay for cellular transport. In this investigation of the adaptation layer's performance the simulation environment described here played an important role.

2 SIMULATION TECHNIQUES

For large and complex systems a fully detailed simulation of the entire problem is often unrealistic. A byte-level simulation of a single ATM connection is so time-consuming that it is impractical in real investigations. While in simpler systems (PSTN or other constant bit-rate, single application communication systems) a higher level investigation may be appropriate, a more sophisticated system's characteristics such as bit error rate or delay can depend largely on lower level behaviour.

In the simulation of ATM based wide band cellular networks an additional difficulty arises from the fact that *events at various levels of abstraction* and *at various time scales* need to be modeled and simulated. For instance, low level changes in the quality of the radio interface may trigger a handover event at the connection level, which, in turn, may have cell level consequences inside the affected switches. We observe that this basic characteristic has two major general requirements for an efficient and practically useful simulator:

- the description, modeling and simulation of the system must be able to capture relevant events at whatever level of abstraction they happen;
- the description and modeling of the system must support the simulation of events at whatever time scale they happen.

(Note that the term *relevant* here refers to application specific modeling details.) We refer to these two basic requirements as *spatial* and *temporal* scalability of the simulator respectively.

Extending the classification of (Frost 1988) and (COST 1992) the various techniques for enhancing modeling and simulation efficiency of complex systems fall into the following broad categories:

- hybrid models increase the efficiency of the simulation by combining analytical models with simulation, see e.g. (O'Reilly 1984), (Lavenberg 1979) and (Frater 1989). Our method inherits the basic (rather general) idea of combining analytical and simulation techniques, as described in Section 3.
- variance reduction techniques improve computational efficiency by using statistical methods to obtain more accurate performance measures, as in (Shanmugan 1980), (Villén-Altamirano 1991), (Law 1991), (Fishman 1983), (Rubinstein 1985) and (Lavenberg 1982). We have found that finding a

good probability transform at various abstraction and time scales can be difficult. Even though these methods offer a considerable increase of simulation speed without requiring more processing capacity so far their applicability has only been shown for relatively simple examples and their extension for more realistic problems needs further research. For an overview of these and other special simulation techniques including hybrid and hierarchical simulation see (Frost 1988) and (COST 1992).

- extrapolative methods increase computational efficiency of a simulation by employing statistical methods to estimate the tail probability distribution outside the sample range (Jeruchim 1984), (Weinstein 1971), (Berberana 1990), (Dijk 1991).
- parallel and distributed methods attempt to increase the simulation time by employing more computer resources, see e.g. (Fujimoto 1994) and (Pham 1997) and the references therein. The performance of even advanced parallel simulation techniques, however, does not seem to justify the additional programming effort which is needed in the decomposition and synchronization tasks inherent in such techniques.
- co-simulation techniques aim at loosely interconnecting two or more independently running simulators of different abstraction levels by allowing them to exchange messages. This approach though attractive, often suffers from problems caused by timing and causability constraints (Coppola 1997). The challenge of efficient communication between the various levels in multiple time scale simulations is addressed in e.g. (Hines 1997), but the solution proposed there is not directly applicable to communication networks. Our approach is in fact a one directional co-simulation technique, also importing ideas from the hybrid approach. The main benefit of these changes is that the higher level simulator never needs to await results from the lower level counterpart. Instead, when needed, the higher level simulator uses predictions.

Recently a new and interesting approach, the *fluid-flow simulation technique* has been proposed in (Kesidis 1996) and (Gustafsson 1997) which adopts basic concepts from the fluid-flow analysis approach into traditional discrete event simulation techniques. This approach appears to be very efficient in deriving performance measures at the cell level in ATM networks, but seems to be difficult to extend to the network level and meet both our spatial and temporal requirement at the same time.

Also recently, some work has been started in simulation of wireless networks and services, (Mishra 1996) gives a classification of the proposed solutions with some references. An interesting simulation of voice over ATM can be found in (Iyer 1997) but this approach does not aim at scalability to large networks. Parallel simulation is proposed by (Liljenstam 1997). As in the case of broadband networks, the main problem is finding the proper balance between modelling complexity and simulation efficiency. Our solution leaves

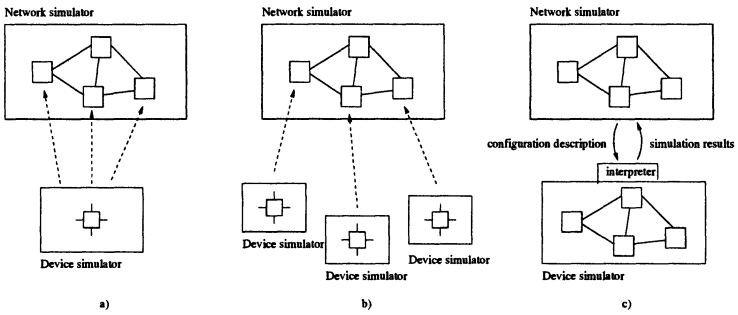

Figure 1 Hierarchical simulation: traditional approaches (a and b) and with flexible device-level simulator (c).

this choice to the user who can set the level of abstraction depending on the actual problem and on the required precision.

3 PROPOSED HIERARCHICAL SIMULATION APPROACH

3.1 Concept

In the case of fixed cellular networks the main focus of performance analysis is on the trade-off between network utilization and per-connection service quality parameters. Typically, the network's response to different connection control and routing strategies needs to be evaluated with service quality requirements as optimization constraints. This kind of investigation requires that the entire network be studied while the model is detailed enough to include the internal structure of network elements down to queues and processors. As this does not seem to be feasible in one simulator we propose a hierarchical decomposition of the problem.

As Figure 1 shows in the case of a hierarchical decomposition the lower level simulator(s) either provide characteristics about a number of identical or similar network entities (Figure 1a) or a dedicated lower level simulator must be assigned to each network element of interest (Figure 1b). While the former solution is based on the investigated system's specific inherent feature of having a number of identical network elements working in similar circumstances (which does not necessarily apply for cellular systems) the latter requires the use of a number of simulators in parallel which might come back to the problem of insufficient processing capacity with the additional problem of requiring a specific simulator for each network element of interest.

In our approach only one simulator is used at the lower level (Figure 1c) but that is designed in a flexible way which allows for the device-level simula-

tion of an almost arbitrary subset of the entire network. The simulation speed will, of course, depend on the size and complexity of the selected subset. This flexible device-level simulator is equipped with a communication interface using a configuration description language designed specifically for this purpose. Configuration descriptions arriving to this interface are interpreted inside the device-level simulator and a simulation session starts immediately. After the session, simulation results are available through the same interface.

The device-level simulator can, of course, simulate only one configuration at a time. But instead of defining this configuration in advance, it is *dynamically configured and re-configured* by the network-level simulator during simulation time. A single device-level simulator running orders of magnitude slower than the network-level simulator can not continuously provide information on the behaviour of each network element. This is, however, not necessary if the primary output of the investigation is the system's call-level behaviour, typically the load at certain network elements, call blocking ratio or network revenue as a function of different routing or admission control policies. While lower level, for instance Quality of Service parameters might be of equal importance, their exact value is usually of no interest as long as they satisfy certain system-specific bounds.

In this kind of studies a pure network level simulation is not sufficient as it does not give reliable information on the number of times the bounds are violated. However, a full cell-scale simulation is not only infeasible but also a waste of CPU time by providing, for example, cell delay values with millisecond precision for paths where delays are tens of milliseconds *below* the limit of tolerable delay. Our approach avoids this waste by concentrating device-level simulation power to points in time and to areas in the network when and where the violation of bounds is suspected to be frequent. Simulation is primarily performed on the upper level, allowing the user to focus on network-level behaviour. As not all network elements are simulated at the device level, the call-level simulator must be prepared for estimating device-level behaviour, typically based on the equivalent bandwidth approach. *Whatever precision this estimation gives will determine the network control and behaviour.* However, thanks to the device-level simulation sessions, the accuracy of the *information learned from the simulation* is not limited by the estimation. In brief, we will obtain accurate information, on both call and cell level, about a network controlled by inaccurate estimates. Despite its limitations, this method tends to model real networks that are typically *controlled* by inaccurate estimates but allow for *measurements* of "arbitrary" precision while improvement of the estimates is based upon feed-back from these measurements.

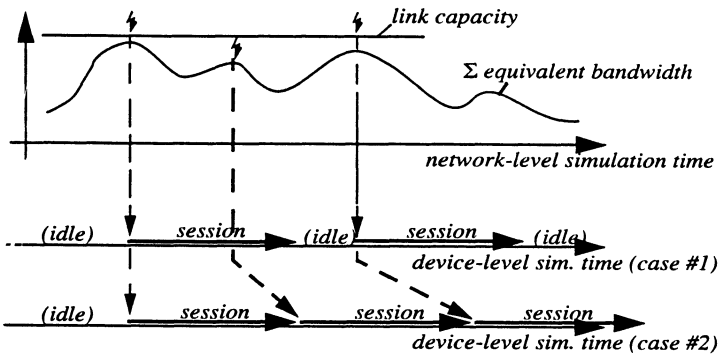

Figure 2 Simulation time scales.

3.2 Communication and synchronization

Figure 2 schematically illustrates the concept in operation for an elementary example: packet-switched connections being established and released on a single physical link. The network level simulator controls the process by calculating the connections' aggregated equivalent bandwidth and applying admission control. Looking at just the network level simulator's output gives sufficient information on the revenue subject the equivalent bandwidth estimate. It also gives the information that during "most of the time" the service quality must have been satisfactory since the estimated equivalent bandwidth was far below the link capacity.

Device-level simulation sessions triggered at critical time spots supplement this information with service quality parameters when cell loss was not negligible. At the end of the simulation shown in the figure, we will have information on revenue *and on service quality* subject the equivalent bandwidth estimate. At the same time device-level simulation results on service quality provide a cross-checking of the estimation accuracy and eventually give indications of its error.

In this basic example the gain compared to a full cell-scale simulation only comes from omitting, in the device-level simulator, periods other than the critical periods. However, as we will see in the examples of Section 5, for larger networks the "cut" can be made both in time and in complexity: at the device-level we only simulate the network elements "seriously affected" by the critical period. This further increases simulation performance, however, for the price of introducing device-level simulation inaccuracy due to neglected network areas.

The time scales shown in Figure 2 also show how simulation performance can be traded for accuracy by modifying the definition (threshold) of "critical

period", hence changing device-level simulation frequency. In case No.2 an additional point in time was considered critical compared to case No.1. This obviously increases confidence in results concerning service quality but it also increases the number of device-level simulation sessions hence decreases overall simulation speed. By further adjusting device-level session frequency, the precision can be set freely ranging from pure network-level to pure device-level simulations.

Note, that though it would be possible to actually use device-level simulation results in the network-level simulator, in our system this is not the case. While device-level simulation sessions are triggered and configured by the network simulator, their result is only fed back to the network simulator to make up a compact representation of the simulation results but are not used to automatically modify the estimation algorithm: that is still up to the human. The two simulators can therefore run independently; if for example they share the same CPU, the network-level simulator might but does not have to wait for the device-level simulation session. In our experimental implementation, unlike in Figure 2, several instances of the device-level simulation may run in parallel which allows the system make use of a network of computers.

3.3 Limitations and drawbacks

Despite its flexibility, the proposed concept has some inherent limitations. In its current form, it does *not* eliminate the need for equivalent bandwidth estimation in the network-level simulator. The hierarchical structure requires that each network element and each traffic source be assigned a model at both levels. Furthermore, as traffic is primarily simulated on call scale, the study of connection-less traffic is excluded. Due to the load-dependent simulation sessions, overall simulation time will depend on network load rather than on the amount of traffic.

The concept can be extended by letting device-level simulation results modify the estimation technique used in the network simulator when matching proved to be poor. Though this extension offers some significant advantages (primarily the possibility of "learning"), it also brings up new problems, in particular the appearance of a closed control loop and the degradation of accuracy due to the two simulations being repeatedly based on each other's results. This is, however, probably the most promising direction to extend the concept. Further extension possibilities include the application of dynamic device-level simulations and the introduction of a control level that allows for starting the simulation *without* an equivalent bandwidth estimation method and building one up gradually while device-level session frequency is decreased.

4 THE SIMULATION ENVIRONMENT

4.1 Model

The network-level simulator is an extended version of the PLASMA ATM simulator, inheriting most of its networking capabilities. For a detailed description of its functionalities and structure, the reader is referred to (Haraszti 1995), here we only list the most important features.

Traffic in the network simulator is modelled at connection-level where users are characterized by calling behaviour and mobility parameters. Users randomly initiate calls of different applications where each application is assigned a traffic description, a set of service quality requirements and a priority level. The traffic descriptions might also have some open parameters (e.g. peak rate) for which values are taken from a probability distribution at call initiation. The traffic descriptions, service quality requirements and priority level are used both by the network simulator for the equivalent bandwidth calculations and by the device-level simulator to build up the traffic generator and to handle traffic in the network.

At connection setup Call Admission Control is performed for each hop on an equivalent bandwidth basis. In each node a destination-based fixed or fixed alternate routing decision is taken.

In accordance with the UMTS (Universal Mobile Telecommunication System) concept we have studied networks consisting of MSC (Mobile Switching Centre), RNC (Radio Network Controller) and BS (Base Station) types of nodes. The nodes are interconnected by ATM VCCs, voice traffic is carried in AAL2 connections according to the recent ITU recommendation (AAL2 1997). Switching is performed in MSCs and RNCs only, BSs originate and terminate connections. Though the simulation environment includes the modelling of the air interface, radio characteristics and mobility, these are outside the scope of this paper and can simply be considered as a modulation of user location and call behaviour and as a shaping of traffic arriving at the BS.

In the device-level simulator traffic is modelled at packet level. The basic unit of user information is the air frame that corresponds to the amount of data transmitted to/from the mobile terminal in one burst. Traffic sources generate air frames with stationary interarrival time and size distributions defined by the application-specific traffic descriptions. The mechanism is identical for upstream and downstream but data connections are not necessarily symmetrical or bidirectional.

Air frames belonging to one connection are transmitted between a BS and a MSC through a number of hops in dedicated AAL2 or ATM connections for voice and data sources, respectively. The establishment and release of these connections is not modelled in the device-level simulator. As defined in the recommendation, air frames from voice sources are packed for transmission in

AAL2 packets that in turn are packed into ATM cells as shown in (Eneroth 1997).

All AAL2 connections on a link are multiplexed in a single VCC and this "composite" voice trunk shares the link with other VCCs carrying traffic from one data source each. In order to allow mobility-related data processing and to support handovers, connections are demultiplexed and re-multiplexed in the RNCs. The multiplexing of AAL2 packets into ATM cells and of ATM cells onto the link is modelled as single-server finite-buffer queues where service time is dependent on ATM cell rate and on link capacity at the two levels, respectively.

4.2 Network to device-level mapping

To initiate a device-level simulation session first the area defined as "critical" needs to be determined, then both the configuration and the actual traffic situation must be mapped on the device-level model. In the experimental implementation the critical area can be defined with the granularity of one node, that is, each node of the network is assigned a model on device-level which is either entirely included in a session or is omitted. Hence to investigate an overloaded link, apart from the link itself the node generating traffic in the overloaded direction must be simulated at device level. With this limitation, once the selection of critical area is made, the configuration mapping is straightforward.

As statistical resource allocation in the ATM/AAL2 system is performed at both multiplexing levels, an equivalent bandwidth estimate is necessarily maintained at each AAL2 or ATM multiplexer. Correspondingly, thresholds of "critical load" can be specified for each level. We refer to these as the critical thresholds. A device-level simulation session is triggered for a multiplexer if the total equivalent bandwidth of its connections related to the multiplexer's output rate reaches a critical threshold.

Traffic is mapped on the device-level model per-connection: for each connection that contributes to the traffic in the critical area a traffic generator is built up using the application-specific traffic description and the connection-specific parameters. To take into account the shaping effect of previously passed swithes, the ingress ports of the overloaded switches are also modelled.

4.3 Simulation architecture

The hierarchical architecture of the simulation environment is illustrated in Figure 3. The figure also shows that the network-level simulator is actually built up of two interconnected simulators: a mobility and air interface sim-

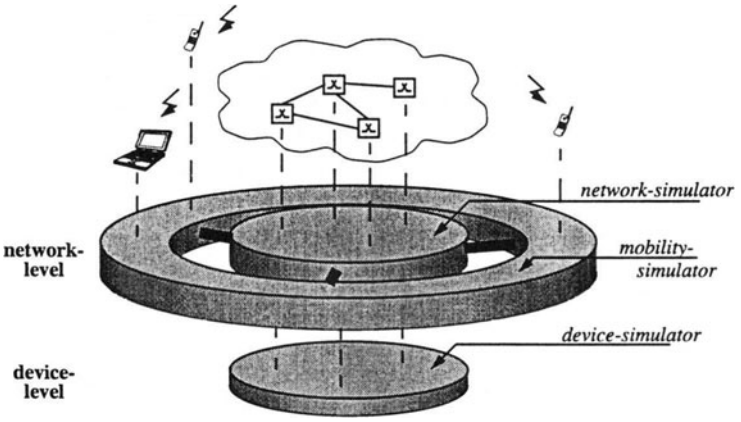

Figure 3 The simulation architecture.

ulator and the network simulator. As the modelling of mobility and radio characteristics are outside the scope of this paper, however, the former can simply be considered as a modulation of user behaviour and we can focus our attention on the latter.

The simulators are built in an object-oriented way around discrete event-driven kernels in the PLASMA simulation and management environment. (Haraszti 1995) The simulators communicate through a CORBA interface allowing the network simulator to "see" a device-level simulator as if it was one of its internal simulation objects. This not only makes communication simple but directly allows for multiple instances of device-level simulators making use of a network of computers.

4.4 Validation

The hierarchical structure of the proposed simulation environment implies that validation must be performed for both the upper and the lower level simulators and for the hybrid system. In this section we present one example from a set of test cases that we have used for validation purposes. Since the hybrid system is expected to approximate the results of the pure device-level simulation, its validation has been based on the comparison with device-level results and will be discussed together with the numerical examples in Section 5. The difficulty of validating the numerical results obtained with the hybrid simulator stems from the fact that the studied ATM/AAL2 technology is currently being standardized, and prototype systems are under construction.

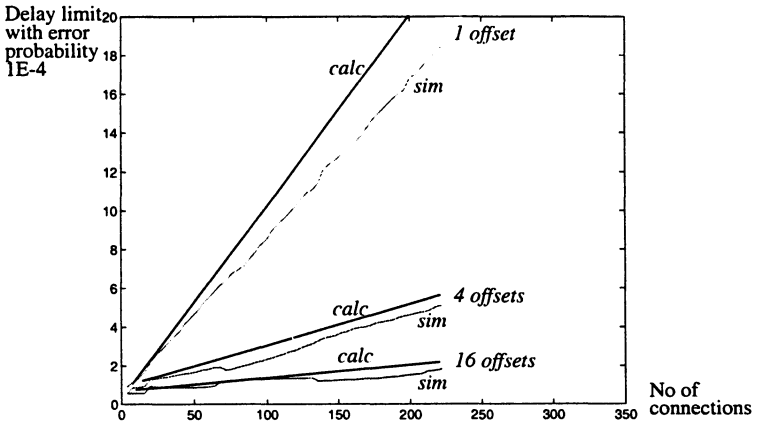

Figure 4 Simulated and calculated maximum number of connections in the $N \cdot D^{[x]}/D/1$ queueing model.

In order to validate the *network level* part of the simulator we have considered a number of cases detailed in (Lin 1978) and (Magi 1996). In (Magi 1996) a model for multirate circuit switched loss networks with non-zero call processing time is developed, which allowed us to compare simulation and analytical/approximative results in non-trivial cases.

To validate the *device level* part of the simulator, we have used two test cases where comparison with analytical/approximative techniques is feasible. First, we consider a single queue-single server system with batch arrivals as in (Bisdikian 1996). The $D^{[x]}/D/1$ queueing system was chosen because it plays an important role in the modelling of any system which carries compressed variable bit rate voice samples over ATM, most notably in the modelling of GSM/UMTS systems with AAL2 transport, see e.g. (Valko 1997).

Figure 4 is an example out of the series of test cases where we have compared analytical and simulation results on this queueing system. Specifically, we use the $N \cdot D^{[x]}/D/1$ version of this queueing system to study the maximum number of allowed AAL2 channels over an ATM VCC with a given QoS constraint. This queueing system operates as follows. Independent, identically distributed batches of random size x arrive at the queue from N independent sources at discrete, deterministic time intervals. If all N batches arrive at the same point in time, we refer to the system as one with a single offset. The batches are then queued and served in a FIFO manner. If on the other hand groups of batches arrive at different points in time, we refer to the system as one with multiple offsets. For instance, if the N independent sources are grouped into four groups, we say that the system is a four-offset system. This is the case in a cellular network, where all mobiles belonging to a base station

are synchronized such that there is a deterministic time offset between groups of mobiles, each communicating with the base station at periodic, discrete time intervals. Here, x corresponds to the number of code bits (and the size of the AAL2 packet) generated by the voice coder in the mobile terminal. Since the QoS constraints, such as the delay of the air frames have to be fulfilled even for the last arriving packet in any batch, it is clear that the multiple offset synchronization method allows for more connections on a link with fixed capacity, as it is shown in Figure 4. In the Figure we note that the simulator results are acceptable.

Building on these ideas and results from $N \cdot D^{[x]}/D/1$ type queueing networks, we have compared analytical and simulation results in some multi-node cases as well, and have found that the device level simulator performs well.

5 SIMULATION EXAMPLES AND RESULTS

5.1 Single-link example

In this example voice and data connections are established and released on a link of capacity $C = 1.5Mbps$. 50 voice and 20 data sources initiate calls according to Poissonian arrival processes with parameters $\lambda_v = 0.002$ and $\lambda_d = 0.001$, respectively and maintain the connections for exponentially distributed times with parameters $m_v = 500sec$ and $m_d = 1000sec$, respectively. Active voice sources generate packets with a constant inter-arrival time $T = 10ms$ where the packet size is determined by an embedded state machine of four states such that the mean rate is $9kbps$ and the peak rate is $20kbps$. The measurement-based four-state model is extensively described in (Valko 1997). Active data sources are of on-off behaviour with exponentially distributed "on" and "off" period lengths with parameters $\alpha_{on}/\alpha_{off} = 0.23$ and rate $r = 64kbps$ in the "on" state. Traffic sources are all independent. Both applications tolerate a maximum packet loss probability of 10^{-3}.

Active voice sources are assigned an AAL2 connection each and are all statistically multiplexed in a single ATM VCC. This VCC is furthermore statistically multiplexed with and is prioritized over the VCCs assigned to the active data sources, one each. Such scenarios are expected in ATM based cellular networks, see e.g. (Eneroth 1997). Figure 5 shows the equivalent bandwidth estimation maintained by the network-level simulator during a 1000 minute simulation. Arrows show the points in simulation time where a critical threshold was reached and a device-level session was triggered. The labels attached to these points show the results obtained by the sessions.

Looking at the device-level simulation triggered at the 90% threshold we observe that the equivalent bandwidth was underestimated since at the traffic peaks service quality was poorer than required. However, from these results only we are unable to accurately estimate the per-connection service quality

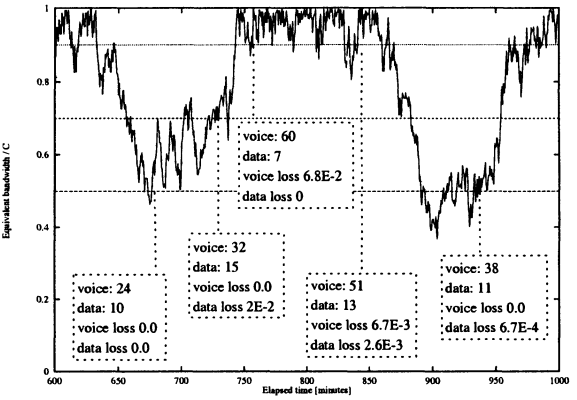

Figure 5 Simulation results for the single-link case.

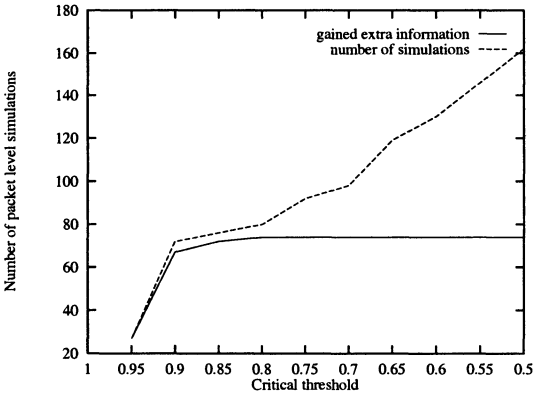

Figure 6 Performance penalty of increased accuracy in the single-link case.

since device-level results are available for the highest traffic peaks only. By lowering the critical threshold first to 70% then to 50% we trigger more frequent device level sessions. From the figure we see that this gives more accurate information on QoS parameters. By further lowering the critical threshold, the pure device level simulation can be approached.

Figure 6 shows the performance penalty of the increased triggering frequency. On the horizontal axis the critical threshold varies from 100% (pure network level simulation) to 50% (practically a pure device level simulation). The curves show the accuracy of the equivalent bandwidth and the required run time respectively. The accuracy is expressed in terms of the number of occurrences when the QoS constraints are violated, which corresponds to the

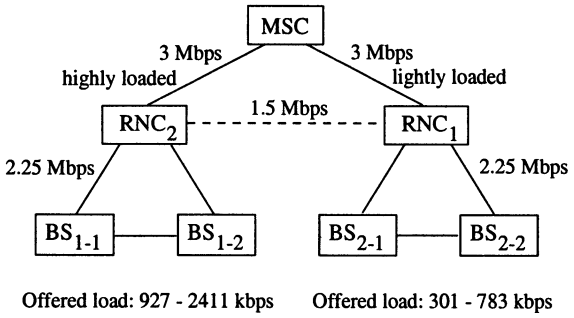

Figure 7 Network example - configuration.

"extra" information gained from the hierarchical system compared to the pure network level simulator. We observe that approaching the pure device level simulation, the run time drastically increases while the accuracy saturates which justifies using the hybrid approach.

5.2 Network example

As a network example we have used the configuration shown in Figure 7. This corresponds to two base station sub-systems each consisting of two BSs and one RNC connected in a ring. The two sub-systems are connected to an MSC node. Mobile users generate voice and data traffic with traffic parameters and QoS requirements as in the previous example. In the first sub-system (left side) the offered traffic is significantly higher than the engineered traffic which results in an overload on the corresponding RNC-MSC link causing high call blocking probability.

In this example we are primarily interested in the decrease of call blocking probability if we apply load sharing to make use of a direct RNC-RNC connection. In Figure 8 this improvement is shown while the total offered traffic is varied on the horizontal axis. In accordance with expectations the blocking could be decreased by applying load sharing between the two RNCs. However, these changes also affect QoS parameters that are not shown by a pure network-level simulation while the cell-level simulation of this network is infeasible.

By exploiting hierarchical simulation we can monitor the packet level QoS without an unacceptable simulation time. In Figure 9 and Figure 10 the total time of QoS-violation out of a 500-minute simulation is shown for the original configuration and for the one with load sharing on the RNC-RNC link, respectively. We can observe that without load sharing QoS is often violated on the overloaded RNC-MSC link and never on the other one. We note on

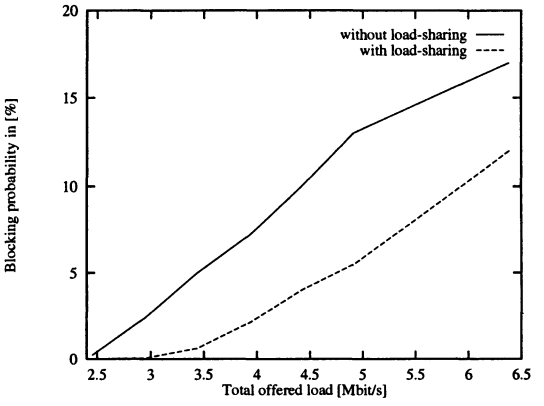

Figure 8 Blocking probabilities with and without load sharing.

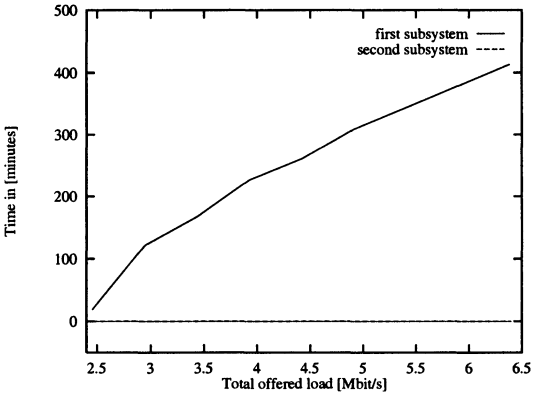

Figure 9 QoS violation without load sharing [min].

Figure 10 that applying load sharing results in similar service quality in the two subsystems.

For these simulations the critical threshold was set to 95% for each multiplexer and the network to device-level mapping was configured such that only an overloaded multiplexer and its outgoing link was simulated in each device-level simulation. With this setting a 500-minute simulation took 300 to 700 minutes depending on total offered load while a complete de- ice-level simulation would take approximately 120 minutes per simulated minute.

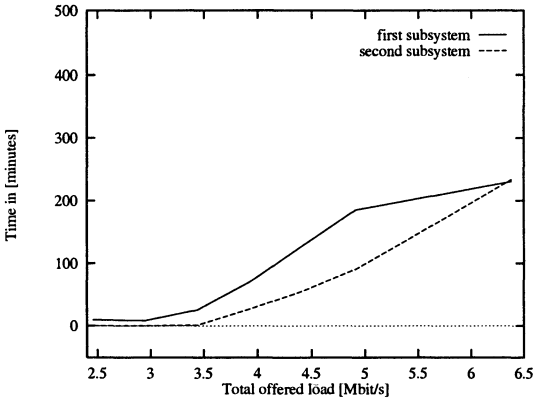

Figure 10 QoS violation when load sharing is applied [min].

6 CONCLUSIONS

In this paper we described a simulation environment optimized for the performance analysis of wideband cellular systems. Due to the large complexity of these systems we saw the need to develop a new hierarchical simulation concept which combines the advantages of fast network-scale simulators and detailed device-scale simulators.

Using the proposed concept a fast network level simulation can be performed while conformance to low level quality requirements is also monitored. A sub-set of a large network simulated on call-scale can be "magnified" and simulated on device-scale if increased traffic or a failure situation deserves a closer look. Last but not least, a simulation study can be placed at an arbitrary point of the "simulation speed versus model precision and confidence" trade-off depending on the requirements of the actual analysis.

These features are made possible by the following main characteristics:

- the simulation environment is built up in a hierarchical structure where the lower level simulator is a generic device-level simulator in which the simulated configuration can dynamically be updated (triggered from the upper level simulator),
- the upper level simulator performs estimates on low-level parameters and uses the lower level simulator to check these estimates,
- low-level characteristics such as bit error rate and cell transfer delay are not the main output of the simulator: we can specify bounds on these and are only interested in the probability of these bounds being violated while our main interest is on network-level behaviour.

7 ACKNOWLEDGMENTS

The authors would like to thank Miklós Boda for his continuous support during this work. We are also grateful to Michael Liljenstam at the Royal Institute of Technology for helping us in the literature review and to the anonymous reviewers for their constructive criticism of the paper.

REFERENCES

Ast, L., T. Cinkler, G. Fodor, S. Racz and S. Blaabjerg (1997) Blocking Probability Approximations and Revenue Optimization in Multirate Loss Networks. *SCS Simulation Journal*, special issue on Modeling and Simulation of Computer Systems and Networks.

B-ISDN ATM Adaptation Layer Type 2 Specification. *Draft ITU-T Recommendation* I.363.2

Berberana, I. (1990) A Method for Estimating Loss Probabilities in Networks of Queues: the Extreme Value Theory Approach. *3rd IEEE Intern. Workshop on CAMAD of Networks and Links*, Toronto, Paper No. 2.2.

Bisdikian, C., J.S. Lew and A.N. Tantawi (1996) The Generalized $D^{[X]}/D/1$ queue: A flexible computer communications model. *Telecommunications Systems*, Vol. 6, No. 2.

Brady, P.T. (1969) A Model for Generating On-Off Speech Patterns in Two-Way Conversations. *Bell System Technology Journal*, Vol. 48, pp. 2445-2472.

Coppola, M. (1997) A Cosimulation Environment with Opnet and VHDL. *13th UK Workshop on Performance Engineering of Computer and Telecom. Systems* (UKPEW '97), Ilkley, UK.

COST 224 (1992) Performance evaluation and design of multiservice networks. *COST 224*

Dijk, V., E. Aanen, H. van der Berg and J.M. van Noortwijk (1991) Extrapolating ATM Simulation Results using Extreme Value Theory. *Proc. 13th ITC, Queueing, Performance and Control in ATM.*

Eneroth, G. and M. Johnsson (1997) ATM Transport in Cellular Networks. *Proc. International Switching Symposium* (ISS '97), Toronto.

Fishman, G.S. (1983) Accelerated Accuracy in the Simulation of Markov Chains. *Operat. Res.*, Vol. 31.

Frater, M., B. Bitmead, R. Kennedy and B. Anderson (1989) Fast Simulation of Rare Events Using Reverse-Time Models. *ITC Specialist Seminar*, Adelaide, Paper No. 9.1.

Frost, V.S., W.W. Larue, K.S. Shanmugam (1988) Efficient Techniques for the Simulation of Computer Communication Networks. *IEEE J. Select. Areas Commun.*, Vol. 6, No. 1, pp. 146-157.

Fujimoto, R.M. and D. Nicol (1994) Parallel Simulation Today. *Annales of*

Operations Research: Simulation and Modeling (ed. O. Balci), Vol. 53.

Gustafsson, E. and R. Ronngren (1997) Fluid Traffic Modelling in Simulation of a CAC Scheme for ATM Networks. *Proc. IEEE MASCOTS '97*, Haifa, Israel.

Haraszti Zs., I. Dahlquist, A. Farago and T. Henk (1995) PLASMA - An Integrated Tool for ATM Network Operation. *Proc. International Switching Symposium* (ISS '95).

Hines, K. and G. Borriello (1997) Dynamic Communication Models in Embedded System Co-Simulation. *34th ACM Design Automation Conference*, Anaheim, CA, USA.

Iyer, J., R. Jain, S. Munir and S. Dixit (1997) Performance of VBR Voiice over ATM: Effect of Scheduling and Drop Policies. *ATM Forum*, 97/0608.

Jeruchim, M.C. (1984) Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems. *IEEE J. Select. Areas Commun.*, Vol. SAC-2, No. 1, pp. 153-170.

Kesidis, G., A. Singh, D. Cheung and W.W. Kwok (1996) Feasibility of Fluid Event-Driven Simulation for ATM Networks. *IEEE Global Telecommunications Conference* (Globecom '96), pp. 2013-2017.

Kurose, J.F. and H.T. Mouftah (1988) Computer-Aided Modeling, Analysis and Design of Communication Networks. *IEEE J. Select. Areas Commun.*, Vol. 6, No. 1, pp. 130-145.

Lavenberg, S.S. and P.D. Welch (1979) Using Conditional Expectation to Reduce Variance in Discrete Event Simulation. *Proc. 1979 Winter Simulation Conference*

Lavenberg, S.S., T.L. Moeller and P.D. Welch (1982) Statistical Results on Control Variables with Application to Queueing Network Simulation. *Operat. Res.*, Vol. 30.

Law, A.M. and W.D. Kelton (1991) Simulation, Modeling and Analysis. *McGraw-Hill*, New York.

Liljenstam M. and R. Ayani (1997) Partitioning PCS for Parallel Simulation. *Proc. IEEE MASCOTS '97*, Haifa, Israel.

Lin, P.M., B.J. Leon, C.R. Stewart (1978) Analysis of Circuit-Switched Networks Employing Originating Office Control with Spill. *IEEE Trans. Commun.*, Vol. COM-26, No. 6, pp. 754-765.

Magi, A. (1996) Performance Evaluation Algorithms for Telephone Networks Taking into Account the Holding Time of Unsuccessful Calls. *MSc thesis*, Technical Univ. Budapest, Dept. Telecom. Telematics, Budapest.

Mishra, P., M. Srivastava, P. Agrawal and G. Nguyen (1996) Ethersim: A Simulator for Wireless and Mobile Networks. *IEEE Global Telecommunications Conference* (Globecom '96), pp. 2018-2027.

O'Reilly, P.J.P. and J.L. Hammond (1984) An Efficient Simulation Technique for Performance Studies of CSMA/CD Local Networks. *IEEE J. Select. Areas Commun.*, Vol. SAC-2, No. 1.

Pham, C.D. and S. Fdida (1997) Perspectives in Performance Evaluation of

Large ATM Networks. *Proc. 5th IFIP Workshop on Performance Modeling and Evaluation of ATM Networks*, Ilkley, UK.

Roberts, J.W. and J.T. Virtamo (1991) The Superposition of Periodic Cell Arrival Streams in an ATM Multiplexer. *IEEE Trans. Commun.*, Vol. 39, No. 2, pp. 298-303.

Rubinstein, R.Y. and R. Marcus (1985) Efficiency of Multivariate Control Variates in Monte Carlo Simulation. *Operat. Res.*, Vol. 33.

Shanmugan, K.S. and P. Balaban (1980) A modified Monte Carlo simulation technique for the evaluation of error rate in digital communication systems. *IEEE Trans. Commun.*, Vol. COM-28.

Short, J., R. Bagrodia and L. Kleinrock (1995) Mobile Wireless Network System Simulation. *First Annual Intern. Conference on Mobile Computing and Networking* (MobiCom '95).

Tran-Gia, P. and H. Ahmadi (1988) Analysis of a Discrete-Time $G^{[X]}/D/1-s$ Queueing System with Applications in Packet Switching Systems. *Proc. IEEE Infocom'88*, New Orleans, LA, USA.

Valko, A., A. Racz, G. Fodor (1997) Voice QoS in 3rd Generation Mobile Systems. submitted to *IEEE J. Select. Areas Commun.*, special issue on Future Voice Technologies, available on request.

Villén-Altamirano, M. and J. Villén-Altamirano (1991) RESTART: A Method for Accelerating Rare Event Simulation. *Proc. 13th ITC, Queueing, Performance and Control in ATM*, pp. 71-76.

Weinstein, S.B. (1971) Estimation of Small Probabilities by Linearization of the Tail of a Probability Distribution Function. *IEEE Trans. on Commun. Technology*, Vol. COM-19.

8 BIOGRAPHY

András G. Valkó received the M.Sc. degree in telecommunications engineering from the Technical University of Budapest in 1994. In '94 he joined the High Speed Networks Laboratory at the Technical University of Budapest. Since '96 he is with Ericsson Traffic Analysis and Network Performance Laboratory. His main interest includes performance analysis and traffic control in mobile systems.

András Rácz received the M.Sc. degree in Technical Informatics from the Technical University of Budapest in 1997. Currently he is a Ph.D. student at the High Speed Networks Laboratory at the Technical University of Budapest and also a part time employee at the Traffic Analysis and Network Performance Laboratory at Ericsson in Budapest. His interest is focused on teletraffic theory and simulation techniques.

Gábor Fodor received the M.Sc. degree in telecommunications engineering from the Technical University of Budapest in 1988. Until '93 he was with the Department of Process Control at ABB Atom, Sweden. In '93 he joined the High Speed Networks Laboratory at the Technical University of Budapest

as a Ph.D. student. Since '97 he is with the Mobile Networks and Systems Research Department at Ericsson Radio Systems, Stockholm, Sweden. His interest includes modeling, simulation and performance analysis of telecommunication systems.

Lars Westberg received Lic. of Techn. degree at the Royal Institute of Technology in Stocholm, Sweden. Since that he has been working with ATM and system performance questions at Ellemtel and at Ericsson Radio Systems. Currently he is with the Mobile Networks and Systems Research Department at Ericsson Radio Systems, Stockholm, Sweden.