EFFICIENT SIMULATION OF LARGE SYSTEMS - TRANSIENT BEHAVIOUR AND ACCURACY

Gábor Lencse
Department of Telecommunications
Technical University of Budapest
Sztoczek utca 2
H-1111 Budapest, Hungary
E-mail: Gabor.Lencse@hit.bme.hu

KEYWORDS

discrete-event simulation, communication networks, performance analysis, statistical synchronization

ABSTRACT

The transient behaviour and accuracy of the Statistical Synchronisation Method (SSM) is studied in a practically important case. An existing high-speed network (two interconnected FDDI Rings) is simulated accurately. The topology, cable lengths and the offered load is taken from the real system. The simulation is performed with and without SSM, and the results are compared.

We conclude that the simulation with SSM is a bit less accurate, but it opens up the possibility for an efficient parallel execution which can be easily implemented.

INTRODUCTION

Discrete event simulation is a powerful method in the performance analysis of communication networks, digital circuits and computer systems. The simulation of large and complex systems requires a large amount of memory and computing power that is often available only on a supercomputer. Efforts were made to use clusters of workstations or multiprocessor systems instead of supercomputers, as this would be much more cost effective. The conventional synchronisation methods for parallel simulation (e.g. conservative, optimistic (Fujimoto 1990)) use event-by-event synchronisation and they are unfortunately not applicable to all cases, or do not provide the desirable speedup. The conservative method is efficient only if certain strict conditions are met. The most popular optimistic method "Time Warp" (Jefferson et al. 1987) often produces excessive rollbacks and inter-processor communication.

The Statistical Synchronisation Method (SSM) (Pongor 1992) is a promising alternative to the conventional methods. Like with other parallel simulation methods, the model is divided into segments that typically execute on separate processors. But unlike other methods, SSM does not exchange individual messages between the segments but rather the statistical characteristics of the message flow. Actual messages are regenerated from the statistics at the receiving side. Further explanation will be given later.

SSM claims to be less sensitive to communication delay and it requires less network bandwidth than event-by-event methods. Nevertheless, it is not accurate in the sense that an event that occurred in one segment of the system does not have an immediate influence on another segment. For this reason, the method cannot be applied in some simulations, for example in the case of digital circuits, but remains feasible in other classes of simulation such as the performance estimation of communication systems.

This paper investigates the transient behaviour and accuracy of SSM in a practically important case, compared to the non-parallel discrete-event simulation method. These issues have not been studied in detail yet. The aim of our study is to examine the applicability of SSM and to compare the quality of the results produced with and without using SSM. We do not deal with the questions of actual parallelisation (e. g. what speedup can be achieved). Our aim is to demonstrate the transient behaviour and accuracy of SSM.

The simulated system is an FDDI network. First, a very accurate simulation of two interconnected FDDI rings is done without any parallelisation. One ring consists of several FDDI stations interconnected by wiring concentrators and the other one is a smaller ring of FDDI stations. The topology and the cable lengths are taken from a real system. The load is produced by measurements taken on the same system. Afterwards, SSM is used between the two rings and the results are compared with the case without using SSM.

This topic was identified as being of importance in the efficient parallelisation of event-driven discrete event simulation facilitating rapid and easy parallel implementation.

The remainder of this paper is organised as follows: first, a brief introduction to SSM is given, then the simulated system is described, next the simulation model is defined, and finally, the simulation results are presented and discussed.

THE STATISTICAL SYNCHRONISATION METHOD

A short summary of the Statistical Synchronisation Method is given here.

Similarly to other parallel discrete event simulation methods, the model to be simulated - which is more or less a precise representation of a real system - is divided into segments, where the segments usually describe the behaviour of functional units of the real system. The communication of the segments can be represented by sending and receiving various messages. For SSM, each segment is equipped with one ore more input and output interfaces. The messages generated in a given segment and to be processed in a different segment are not transmitted there, but the *output interfaces* (OIF) collect statistical data of them. The *input interfaces* (IIF) generate messages for the segments according to the statistical characteristics of the messages collected by the proper output interfaces. (see Fig. 1.)

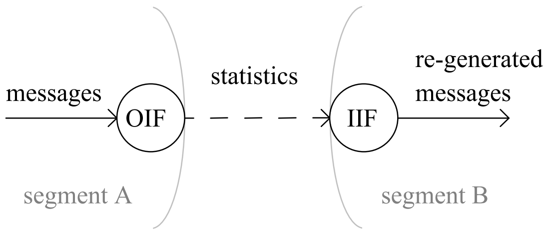

Fig. 1. An OIF - IIF pair

The segments with their input- and output interfaces can be simulated separately on separate processors, giving statistically correct results. The events in one segment have not the same effect in other segments as in the original model, so the results collected during the SSM are not exact. The precision depends on the segmentation, on the accuracy of statistics collection and regeneration, and on the frequency of the statistics exchange among the processors.

Advantages of SSM

SSM has the following advantages compared to the other PDES (Parallel Discrete Event Simulation) methods:

- requires less network bandwidth
- tolerates communication delay better
- can be easily implemented
- requires less support from the simulation kernel
- may produce better speedup

A feasible approach can be that the user implements his simulation as a uni-processor version first. After verification, he replaces the wires on the segment boundaries with statistical interface pairs. He may run the simulation fast on a cluster of workstations and produce results that are probably less accurate than those that can be achieved without SSM but they can be produced much faster and are probably applicable for tuning the model on the basis of them. The final results are to be verified with the usual DES verification methods.

THE SIMULATED SYSTEM

To examine the characteristics of the SSM in a practically important case, a widely used communication network: Fiber Distributed Data Interface was chosen. FDDI is a 100-Mbps fiber optic network standard. (ANSI X3.139 1987) It has a dual ring topology that can be extended by wiring concentrators. The so-called Timed-Token access protocol is applied for media access control purposes.

The aim of this simulation study was to examine SSM in a realistic simulation, so all possible efforts were made to use a simulation model that is very close to an existing network.

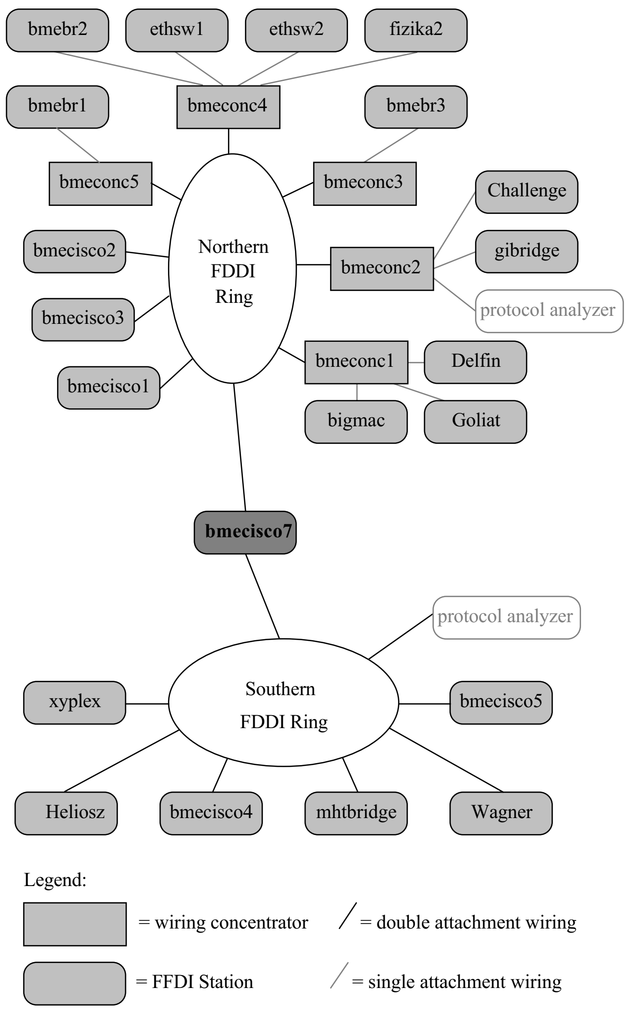

Fig. 2. FDDI backbone of the Technical University of Budapest

The FDDI backbone of the Technical University of Budapest was found to be appropriate. It consists of two rings: The Northern Ring is a university-wide network and consists of 15 FDDI stations interconnected by 5 wiring concentrators. The Southern Ring is the backbone of the

Faculty of Electrical Engineering and Informatics, and being smaller ring of 7 FDDI stations. The topology of the network and the cable lengths were taken from the real system. *Fig. 2.* shows the topology of the FDDI network.

The two rings are interconnected by the *bmecisco7* router.

The load used in the simulation model came from measurements taken on the real rings. By using a protocol analyzer, all the packet heads were copied from the ring and the packet lengths, arrival times, as well as the addresses of the source and destination stations were stored.

The above data were used to produce a very detailed traffic matrix $T=[t_{ij}]$ where *i* and *j* stands for the source and destination stations, respectively and all the t_{ij} elements are two-dimensional distributions of the packets from $station_i$ to $station_j$, the dimensions being packet length and inter-arrival time.

THE SIMULATION MODEL

What is modelled from the FDDI network?

The parts of the FDDI standard that were found to be irrelevant concerning our simulation are not modelled. The simulated FDDI rings are always single rings built up of *Single Attachment Stations* (SAS) and *Single Attachment Concentrators* (SAC). The dual ring topology is not modelled, because the secondary rings are used only for backup purposes and are useful e.g. in case of cable break, but our aim is to simulate the normal operation of the ring. All the SAS's contain a *Media Access Control* (MAC) entity. Its normal operation (Timed-Token Ring Protocol) is simulated precisely, but all the ring initialisation and ring recovery mechanisms are omitted. At the beginning of the simulation, the *operative value* of the *Target Token Rotation Time* (*T_Opr*) is set in all the stations and a token is inserted into the ring by a chosen station. No errors are modelled during the operation as no errors were observed during the measurements. The exact fiber lengths between stations were taken from measurement logbooks. The value of *station latency* was taken from (MIL 3 1996). The value of T_Opr was queried from the FDDI stations. The effect of the wiring concentrators is modelled by a constant delay.

The implementation details of the model can be found in (Lencse and Varga 1997)

The simulation environment

The simulation was performed using the OMNeT++ discrete-event simulator. It was developed by András Varga at the Technical University of Budapest, and is freely available for academic purposes. Readers interested in the simulator should refer to the *OMNeT++ Home Page* (Varga 1997).

Load modelling

As it was mentioned before, the $T=[t_{ij}]$ traffic matrix was derived from the measurements on the FDDI rings. This matrix was observed at given points of the rings where the protocol analyzer was inserted. T is not identical with the $D=[d_{ij}]$ demand matrix, where d_{ij} expresses the two-dimensional (length and inter-arrival time) distribution of send requests arriving from the outside world at $station_i$ (as source station) to be sent to $station_j$. We need to use D for load generation but only T can be easily measured. Now we shall show why one can use T for load generation instead of D, though they are not identical matrices.

Fig. 3. shows the travelling of a packet from its source station through the network to its destination station. The horizontal axis shows the time and the vertical one shows the distance (cable length) and the stations. Let us trace the travelling of a packet. The transmitter of the source station starts transmitting at t_{Src}^{TrBeg} (the abbreviations mean: Source, Transmisson, Begin), and the transmission ends at t_{Src}^{TrEnd} . The receiver of the i-th station starts receiving the packet at t_i^{RecBeg} and it ends the receiving at t_i^{RecEnd}. Similarly, the transmitter of the i-th station starts and ends the transmission of the packet at t_i^{TrBeg} and t_i^{TrEnd}, respectively.

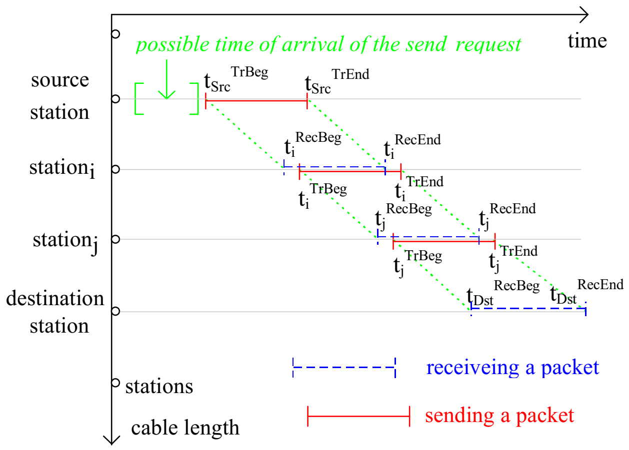

Fig. 3. Travelling of a packet in an FDDI ring

There is a so called *station latency* (SL) delay between the actions of the receiver and the actions of the transmitter of the same station. In this way:

$$t_i^{TrBeg} - t_i^{RecBeg} = t_i^{TrEnd} - t_i^{RecEnd} = SL.$$

If we know one of the above mentioned times, then all the other ones can be calculated in the knowledge of cable lengths, speed of the light in fiber, station latency and packet length. However, it is still unknown what time the packet arrived from the outside world to the source station. It must have happened some time before t_{Src}^{TrBeg}, but using only the information we gained by the protocol analyzer, it is undecidable when exactly. Fortunately, we may say: it arrived to the source station just before its transmission began at t_{Src}^{TrBeg}. This is undistinguishable from any other possible time of its arrival. In this way, we can make the following observations: The Δt_{ik} delay between t_i^{TrBeg} and t_k^{RecBeg} is always the same for all fixed *i* and *k*, because it is caused by multiple station latencies and cable delay. Let the protocol analyzer be $station_k$. The calculation of inter-arrival times of packets originated from $station_i$ destined to $station_j$ omits the identical Δt_{ik} delays and the inter-arrival time calculated on the basis of our observations at $station_k$ may be used as the inter-arrival time of the sending requests at $station_i$. We have shown that the observed T traffic matrix can be used as the D demand matrix.

Now we might calculate the t_{ij} two-dimensional distributions from the data gained by the protocol analyzer. However, the 2D distributions would not be appropriate for routing decisions (as explained in the next section: ”Traffic between the rings”). Two one-dimensional distributions are used: one for the packet length l_{ij} and one for the inter-arrival time τ_{ij}. These distributions are approximated by using histograms. The 1D histogram computation as well as the random number generation on the basis of the collected histogram is a part of the OMNeT++ simulator.

As the average utilisation of the observed FDDI network is below 5%, a factor called *Load Multiplier* was introduced to simulate higher load. The inter-arrival time is divided by this factor so the load becomes Load Multiplier times more. By changing the value of this factor, the load of the system can be easily modified, while the nature of the traffic remains the same.

Traffic between the rings

The two rings are interconnected by a router which has one port in both rings. In the simulation model called ”*wired*”, the two router-ports are interconnected by 2 wires. All the packets destined to the router in one ring will be captured by the router port and sent to the router port connected to the other ring. In the model, that port has to randomly select a destination station for the packet in the other ring, because the original routing information was not captured during the traffic measurement. The traffic measurement was done on MAC level with MAC addresses of the stations and routing is done in the network layer of the OSI seven-layered network architecture model. (Tanenbaum 1989) So when a packet arrives from the router port of ring #1 to the router port of ring #2, the router port of ring #2 has no direct information to which station of its ring the packet should be sent. (see *Fig. 4.*)

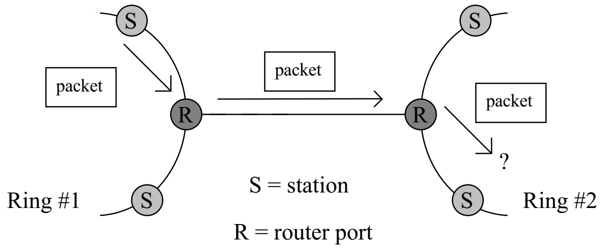

Fig. 4. The problem of routing

The packet length distributions from the router port in the destination ring to all the other stations in that ring (all the l_{ri}, $i \neq r$, r means the router port) are used for routing decisions. It is done the following way: the router is a normal FDDI station in both rings in the real system and its output statistics were measured and the packet length histograms were calculated. *Fig. 5.* shows an example.

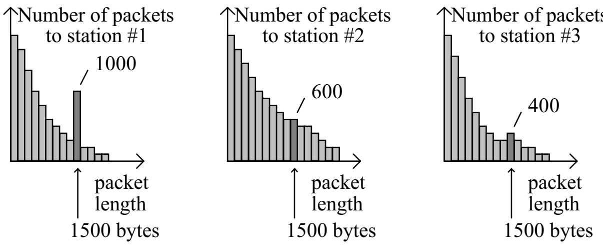

Fig. 5. Packet length distribution histograms from the router port to different stations in ring #2, - for illustration only

These histograms are not used for traffic generation because the traffic of the router port to the stations of one ring always comes from the other ring. They are used only for routing. In the simulation model, the router checks the length of the packet arrived from the other ring. Let it be 1500 bytes in our example. The router port retrieves the heights of the packet length distribution histograms in that position. They are 1000, 600 and 400 in our example. Then it randomly selects the destination station for the packet so that the probability of the selection of $station_i$ is proportional to the height of the length distribution histogram from the router to $station_i$ (l_{ri}) at the position of the length of the current packet. In our example the probability of the selection of $station_1$, $station_2$ and $station_3$ is 50%, 30% and 20%, respectively.

In the simulation model called ”*SSM*”, the two router ports operate in the same way, but they are interconnected through statistical interfaces.

Statistical interfaces

The statistical interfaces of the SSM consist of output interface (OIF) and input interface (IIF) pairs. An OIF captures the messages, collects statistics about them, and if some conditions are met it sends the statistics to the corresponding IIF. The IIF generates messages on the basis of the statistics it received from the corresponding OIF.

The trigger condition of the OIF to send the collected statistics, should be chosen by the user. In this article, the transmission of the statistics is controlled by the so called *update threshold* (UT) parameter. The OIF counts the captured messages and if the counter reaches UT, the OIF sends the collected statistics to the appropriate IIF, and also restarts the statistics collection. In this way, when the OIF sends its statistics they are already based on enough observations to achieve the required accuracy, but this method may produce long transient if the rate of message arrival to the OIF is low. Another approach could be that the OIF sends its statistics after a certain simulation time has elapsed. Using this second method, the length of the transient caused by SSM is bounded, but the accuracy of the statistics is not ensured. Of course, it is possible that OIF does not delete its statistics at the time of sending, but it must somehow delete the old observations, otherwise the changes in the statistical characteristics could not go through the OIF-IIF pair.

In this article, the length and the inter-arrival time of the messages are observed by the OIF's and the collected statistics (histograms) are sent to the appropriate IIF's. The IIF's re-generate the traffic on the basis of the statistics.

SIMULATION RESULTS

Initial transient caused by the SSM

The aim of our first experiment was to test the operation of the OIF - IIF pairs. To set up a reference, the two rings were interconnected with wires first. The *Load Multiplier* factor of the Northern Ring (NR) and the Southern Ring (SR) was set to 4 and 1, respectively. In this way the majority of the load of the Southern Ring came from the Northern Ring. The utilisation of the Southern Ring was measured. Then, the wires were replaced by statistical interface pairs. The simulation was performed with the values 100 and 1000 for the UT (update threshold) parameter of the OIF.

The utilisation of the Southern Ring was calculated in each 25 ms (simulation time) wide window and plotted in the function of time. *Fig. 6.* shows the average of 20 simulation runs with different seeds for the random number generator. The averaging of the results of multiple runs was necessary because there were too much fluctuations - the 25 ms window size could not be increased because the initial transient would have disappeared. The curve titled "wired" serves as a reference. The curves titled "SSM UT=100" and "SSM UT=1000" were produced by using SSM with UT of 100 and 1000, respectively. "SSM UT=100" has a short transient at the beginning and "SSM UT=1000" has a much longer transient period. This is explained by the fact that in the second case 10 times as many messages have to arrive to the OIF of the Northern Ring before the first statistics are sent to the IIF of the Southern Ring.

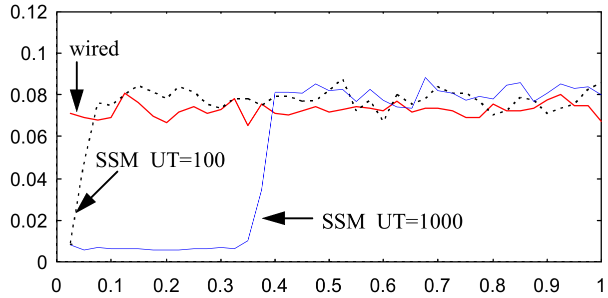

Fig. 6. Utilisation of the Southern FDDI Ring

Transient in the simulated system

We have already seen that SSM causes an initial transient even if the simulated system shows no transient behaviour. Let us examine what happens if there is a transient in the simulated system.

The following simulation experiment was performed with and without using SSM. With the help of the Load Multiplier factor, the offered load of the Northern Ring was raised by a factor of 2 at t_1=1 and it was set back to its original value at t_2=2s. We measured the utilisation in both rings. It was averaged in a 25 ms window for 20 simulation runs and depicted on *Fig. 7*.

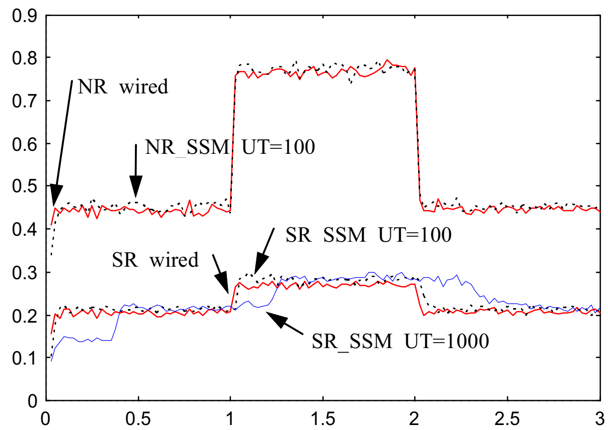

Fig. 7. Transient behaviour: the load changes

The "wired" curves of both rings are regarded as references. The curves "NR_SSM UT=100" and "SR_SSM UT=100" are produced with SSM using update treshold of 100 in case of both rings. The NR_SSM rises with NR_wired, because the load from the Southern Ring to the Northern Ring does not change. After t_1 the IIF of the Southern Ring still genates packets according to the old statistics, so SR_SSM does not raise until the IIF gets new statistics. As this happens fast with UT=100, the transient of the "SR_SSM UT=100" is hard to see, that's why we have also plotted "SR_SSM UT=1000". The situation is similar after t_2, but the transient is longer, because the OIF of the Northern Ring sends the statistics after the arrival of every UTth packets and after t_2 the load in the Northern Ring is significantly less then in the

(t_1 , t_2) interval. In this way, the length of the transient depends on the rate of packets arriving to the OIF.

Accuracy of the results

In many cases, the aim of the simulation is to study the steady state behaviour of the system. Let us examine how accurate results can be produced by using SSM.

Now, we would like to ignore the effect of the initial transient, and make longer simulation runs. *Fig. 8.* shows the results of 10 simulation runs averaged in a 50ms wide window. This figure shows that the results produced by SSM are very close to the results of the wired case.

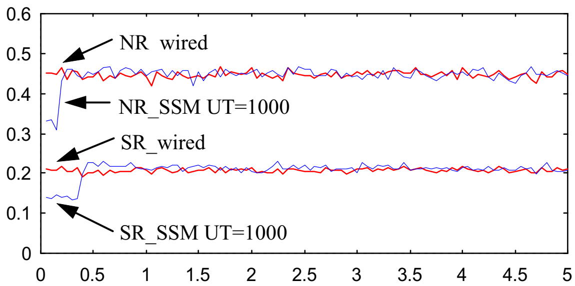

Fig. 8. Utilisation of the FDDI rings with and without SSM

As the load and therefore the utilisation fluctuates, it is not easy to see the average. The average utilisation in the (1s, 5s) interval was calculated for the Northern Ring. It is 0.4470 and 0.4492 for the wired case and for using SSM, respectively. The difference is below 0.5% of the reference value (wired case). The average values for the Southern Ring are 0.2076 and 0.2127 for the wired case and for using SSM, respectively. The difference is below 2.5% of the reference value (wired case).

If one is interested in the steady state behaviour of the system, then he detects the end of the initial transient period of the system and ignores the results collected during the transient period. Methods exist for the detection and elimination of the initial transient of the system from result calculations. They may be used to eliminate the transient caused by the statistical interfaces too.

CONCLUSION

The transient behaviour and accuracy of the Statistical Synchronisation Method was tested and compared to the traditional event-by-event synchronisation. The selected simulated system was an FDDI network. It was found that SSM produced similar results than the traditional simulation used as reference. The length of the transient caused by SSM depends on the frequency of statistics exchange and the packet arrival rate to the statistical interface pair. The difference between the results produced by using SSM and the results of the reference simulation was below 2.5% of the reference value.

We conclude that SSM makes it possible to implement efficient parallel simulation of large systems on clusters of workstations or multiprocessor systems.

The direction of the further research is the parallel execution of the segments of the simulated system on interconnected computers using SSM between the segments.

ACKNOWLEDGEMENTS

Thanks to András Varga, the developer of the OMNeT++ simulator for his technical support and to the staff of the Center of Information Systems of the Technical University of Budapest especially to Ferenc Pál and Béla Győri for their help in taking traffic measurements on the FDDI rings and for providing exact data about the topology and cable lengths.

REFERENCES

ANSI X3.139. 1987. *Fiber Distributed Data Interface (FDDI) Token Ring Media Access Control (MAC)*

Fujimoto, R. M. 1990. "Parallel Discrete Event Simulation". *Communications of the ACM* 33, no 10, 31-53

Jefferson, D; B. Beckman; F. Wieland; L. Blume; M. DiLoreto; P. Hontalas; P. Laroche; K. Sturdevant; J. Tupman; V. Warren; J. Vedel; H. Younger and S. Bellenot. 1987. "Distributed Simulation and the Time Warp Operating System". *Proceedings of the 12th SIGOPS - Symposium on Operating System Principles*, pp. 73-93.

Lencse, G; A. Varga. 1997. *OMNET++ Discrete Event Simulation System - Examples Manual.* ftp://ftp.hit.bme.hu//sys/anonftp/lencse/omnetpp/exman.zip

MIL 3. 1996. *OPNET Example Models Manual*, Release 3. (Chapter FDDI) MIL 3, Inc.

Pongor, Gy. 1992. "Statistical Synchronization: a Different Approach of Parallel Discrete Event Simulation". *Proceedings of the 1992 European Simulation Symposium (ESS 92)* (Nov. 5-8, 1992, The Blockhaus, Dresden, Germany.) pp. 125-129.

Tanenbaum, A. S. 1989. *Computer Networks.* Second Edition, Prentice Hall Inc.

Varga, A. 1997. *The OMNeT++ Home Page* http://www.hit.bme.hu/phd/vargaa/omnetpp.htm

BIOGRAPHY

Gábor Lencse was born in Győr, Hungary, in 1970. He received his M.S. in electrical engineering and computer systems from the Technical University of Budapest in 1994. He is currently pursuing his Ph. D. at the same university. The area of his research is computer architectures and parallel processing. He is interested in discrete event simulation.