STATISTICS COLLECTION FOR THE STATISTICAL SYNCHRONISATION METHOD

Gábor Lencse

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

Department of Telecommunications
Széchenyi István College
Hédervári u. 3.
H-9026 Gyôr, Hungary
E-mail: lencse@szif.hu

KEYWORDS

parallel discrete-event simulation, statistical synchronisation, statistics collection methods, nonparametric density estimation, accuracy.

ABSTRACT

The statistics collection methods applied to the Statistical Synchronisation Method (SSM) are studied. The distributions to be estimated are assumed to be time invariant. The accuracy and the resource requirements of the statistics collection is investigated both in general for some well-known distributions and in a practically important case: in a simulation of an FDDI network.

INTRODUCTION

The Statistical Synchronisation Method (SSM) (Pongor 1992) is a promising alternative to the conventional synchronisation methods for parallel discrete event simulation (e.g., conservative, optimistic) (Fujimoto 1990).

The conventional synchronisation methods use event-by-event synchronisation between the segments of the simulated system 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.

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.

The transient behavior and accuracy of SSM applied to an FDDI simulation were already demonstrated in (Lencse 1997), and the questions of parallelisation using SSM were discussed in (Lencse 1998). The latter paper shows that a very good speed up can be achieved in a PDES (parallel discrete-event simulation) applying SSM and loose synchronisation between the segments of the simulated system, because the processors executing the segments run independent in the vast majority of time.

The aim of this paper is to examine what statistics collection methods should be for used for SSM to be able to faithfully regenerate the statistical characteristics of the message flow. The distributions to be estimated are assumed to be time invariant. The accuracy and the resource requirements of the statistics collection, sending and regeneration are investigated both in general for some well-known distributions and in a practically important case: in a simulation of an FDDI network.

The applied distribution estimation methods are: count the relative frequency of the possible values of the random variable (for the discrete case only), the equidistant histogram, the Barron estimate and two types of the equiprobable bin histograms.

The investigated resource requirements are: computation used during the statistics collection and statistics regeneration, storage requirements during statistics collection and regeneration, communication requirement between the segments.

The remainder of this paper is organized as follows: first, a brief introduction to SSM is presented, then the reasons for the choice of the error criterion are given, next the statistics collection methods and their resource requirements are considered, afterwards the types of the distributions that can be met in discrete-event simulation (DES) are discussed, finally the accuracy of the estimation of the discrete, continuous and pseudo-continuous types of random variables are investigated both in general and in a special case.

This topic was identified as being of importance in the investigation of the accuracy of the Statistical Synchronisation Method.

THE STATISTICAL SYNCHRONISATION METHOD

For those not familiar with SSM, a short summary of the Statistical Synchronisation Method is given here. See (Pongor 1992) for more information.

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 behavior 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 or 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 Figure 1.)

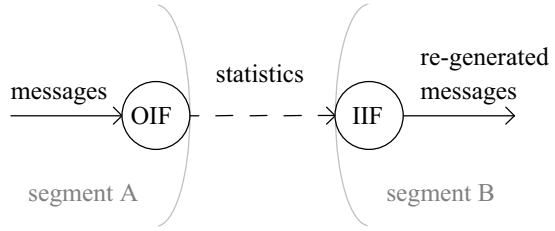

Figure 1. An OIF - IIF pair

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 SSM are not exact. The precision depends on the partitioning of the model, on the accuracy of statistics collection and regeneration, and on the frequency of the statistics exchange among the processors.

THE CHOICE OF THE ERROR CRITERION

Our task is to estimate the unknown distribution of a random variable X having the distribution function $F(x)=\Pr(X<x)$ and (in the

continuous case) probability density function $f(x)=F'(x)$. We estimate $f(x)$ by $g(x)$ and measure the error of the estimation by one of the error criteria:

$$L_p = \int_{-\infty}^{\infty} \left| f(x) - g(x) \right|^p dx, \quad p \geq 1$$

The most popular error criterion is the L_2 (squared error) because of its analytical simplicity. However, it is not adequate here, because it de-emphasizes the tails of a density by squaring the small density values there. L_1 is the criterion of choice, for the following reason: we are interested in that how well we can estimate the probabilities of events i.e. the integral of $f(x)$ for a given interval, rather than the function $f(x)$ itself. In other words how much differs the probability measure of our estimation $g(x)$ from the theoretical $f(x)$ for any Borel sets. The supremum of this difference can be expressed by the L_1 error, Scheffè's Theorem, see (Devroye and Györfi 1985. p. 2.):

$$\sup_{B \in \mathcal{B}} \left| \int_B f(x)dx - \int_B g(x)dx \right| = \frac{1}{2} \int_{-\infty}^{\infty} \left| f(x) - g(x) \right| dx = \frac{1}{2} L_1$$

Another useful property of the absolute error (L_1) is its invariance to monotone continuous changes of scale (Scott 1992. p. 41). L_1 is a dimensionless quantity and it is easy to see that $0 \leq L_1 \leq 2$.

For discrete random variables, L_1 is calculated by sum instead of integral. Let X take the values $x_1, x_2, \ldots x_k, \ldots$ with the probabilities $p_1, p_2, \ldots p_k, \ldots$ Let $q_1, q_2, \ldots q_k \ldots$ denote the measured relative frequencies of the values taken by X.

$$L_1 = \sum_k \left| p_k - q_k \right|$$

THE STUDIED STATISTICS COLLECTION METHODS

As simulation is usually applied to solve complex problems the analytical solution of which is not known, the distributions observed in simulation are not known in the vast majority of cases, so we consider only the nonparametric estimations. In the next subsections we examine the computation, storage and communication requirements for the studied statistics collection algorithms. We assume the reader is familiar with the elementary algorithm theory (Aho et al. 1975) and the C programming language. (C code segments will be set in `courier`.)

Relative Frequency

If a random variable has a discrete distribution and can take not too many values that are known in advance, the simplest way to collect statistics of the random variable is to count for its all possible values how many times they occur in the sample. It is called empirical distribution. Let $x_1, x_2, \ldots x_k, \ldots x_M$ denote the possible values of the discrete random variable X. If X takes its possible value x_k for n_k times in a sample of size N, its relative frequency is $q_k = n_k/N$. To collect statistics one needs an array of counters (that is integers) of size M. In the general case we need to store the x_k values ordered (requires another array of size M, usually an array of type "`double`" in the C programming language) and use binary search for all X_i (i = 1 to N) observations to find j: $X_i=x_j$ and then the j-th counter is to be incremented. In special cases, when we have some extra knowledge of the possible values of X, we may avoid the binary search and use a hash function or even direct mapping. (E.g.: Let X take the values 0, 10, 20, … 1000. Then X/10 can be used as the index in the array of counters.)

If we know and store the possible x_k values ordered in advance, the algorithmical complexity of the collection of a sample sized N is O(N*log(M)) in the general case and for statistics exchange, we need to transfer only the counters: M*`sizeof(int)`.

However, if we do not know or do not want to store the possible values of X in advance (perhaps because there are too many of them and some of them are too rare), we need to build the array of the x_k values during the statistics collection. The insertion of a new element may require to move some elements of the array, the cost of which can be linear with the array size, so the complexity of the statistics collection is: $O(N*\log(M')+(M')^2)$, where M' is the number of the different values of X in the actual N sized sample. Of course, the possible values of X (together with the counters) can be stored is a data structure that has a logarithmical insertion cost, for example AVL-tree, 2-3-tree, B-tree, etc. In this case we reduce the additional $O((M')^2)$ cost to O(M'*log(M')). Anyway, the communication costs will include the transfer of the M' elements too: M'*`sizeof(double)`.

At the place of the statistics regeneration, the storage requirement is also M*`sizeof(int)`. The generation of a random number on the basis of the collected statistics requires O(M) number of "+" operations if we use the trivial algorithm: Generate a random number R in the [0, n_k) interval according to uniform distribution. Then add the n_1, ... n_k, ...n_j values until the sum is greater than R. Return x_j as the result. A more sophisticated algorithm is the following one: calculate and store the sum(j) values for all possible j (requires M*`sizeof(int)` storage) and find the appropriate one by a binary search of O(log(M)) steps.

$$\text{sum(j)} = \sum_{i=1}^{j} x_j$$

Equidistant Histogram

If a random variable has continuous distribution or if it has discrete distribution but it may take extremely many values, and it is acceptable, one may use some kind of histogram for statistics collection.

An equal bin width histogram is characterized by its starting point t_0, the size (h) and number (M) of its bins. It requires a counter for all its bins, altogether about M*`sizeof(int)` storage and communication. The collection of the statistics is very simple if we use the algorithm: `counter_of_bin[int((X-t0)/h)]++;` the cost of the collection of N samples is only O(N).

For the generation of a random number on the basis of the collected statistics, one needs to choose a bin and generate a random number from that bin according to uniform distribution. The choice of the bin may be done by the same way and cost as we have seen it in the case of the statistics regeneration on the basis of the relative frequencies of the possible values of the random variable.

Note that this approach can only be used if we know a lower and an upper bound for the distribution. In some practical cases we now these bounds as "high probability bounds" only, that is, with a small probability the value of X may exceed these boundaries, in this case it is common to apply an underflow and/or an overflow bin for counting the extreme values.

Barron Estimate

Barron estimate (Barron et al. 1992) is a histogram-based density estimation method. It collects statistics according to the following steps: Transform the random samples by a fixed distribution function $G(x)$, that is related somehow to $F(x)$, the distribution function of the random variable X. Make an equidistant histogram in the [0,1] with M number of bins of width h from the transformed samples. The calculation of the approximated density function is in Figure 2.

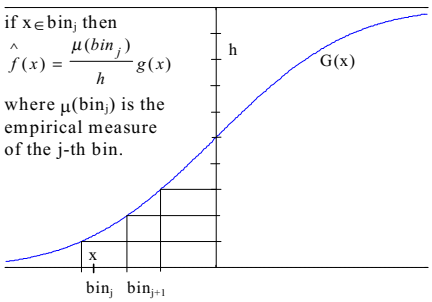

Figure 2. Barron estimate

Note that using $g(x_j)$ instead of g(x), where x_j is an appropriate fixed point in the j-th partition a nearly equiprobable bin histogram estimate can be made if G(x) is close to F(x).

The resource requirements can be derived from that of the equidistant histogram plus the computational costs of the transformation of the samples.

Equiprobable Bin Histograms

Knowing the distribution function *F(x)* one might set up the bin boundaries so that the bins are equi probable. The collection of the statistics requires to find the appropriate bin for the samples. This can be done by a binary search, and the algorithmical complexity of the collection of a sample sized N is O(N*log(M)), just like in the case of collecting the relative frequencies of the samples. The storage requirements include both the storage of the bin boundaries and the counters M*(`sizeof(double)+sizeof(int)`). The other resource requirements are similar to that of the equidistant histogram.

As *F(x)* is not known, this method will only be used as a reference for the following one.

Semi-equiprobable Bin Histograms

We set up the bin boundaries so that we gain statistically equivalent blocks. According to $F_N(x)$ (the empirical distribution function) the partition is equi probable. In this way the bin boundaries of the histogram are computed from the collected samples. We expect, that if the sample size N is large enough, the bin boundaries will be close to the optimal, and the L_1 error of the statistics will be close to that of the above mentioned reference method.

For this method we recommend the following algorithm: Store all the N observations of X. Then sort the collected samples and draw the boundaries of the M bins so that the same number of samples fall into all bins.

The cost of the sort is O(N*log(N)), the storage requirement is N*`sizeof(double)` for storing the observations. Theoretically only the bin boundaries have to be transferred between the segments (about M*`sizeof(double)` bytes), because the counter values are the same for all bins. However, the counters may differ for some reasons: the number of observations is not an exact multiple of the number of cells or if we use this method for collecting statistics of a quantized random variable then there may be equal ones among the observations and they have to be placed into the same bin. In this case the counters also have to be transmitted.

Other Possible Density Estimation Methods

The following two methods work without storing the observations: The P^2 method (Jain and Imrich 1985) calculates the quantiles of a density. The K-split method (Varga 1997) maintains the number of cells (i. e. bins) optimal for the distribution and the number of observations by doing cell splits.

Other nonparametric estimations are: frequency polygon, averaged shifted histogram and the different kernel methods. (Scott 1992)

These methods we do not study.

The Consideration of the Resource Requirements

When selecting the statistics collection method we must take care for the resource requirements described above not to slow down the simulation. The estimation of the number of inter-segment messages per simulation execution time seconds may be necessary. The accuracy of the estimation methods has primary importance, but it is also essential to check that the storage, computation and communication requirements of the algorithms can be satisfied without significant loss in the speed of the simulation.

THE TYPES OF THE DISTRIBUTIONS IN DES

In many cases random variables in a DES take discrete values. For example in a network simulation packet length, queue length, number of active stations etc. take not only discrete but even integer values. On the other hand the time of events is usually referred as continuous random variable. Let us introduce the quantized random variables for the following reasons.

- the resolution of the floating point variables of computers is finite
- a smallest unit of time may exist in the simulated system
- if the input data of the simulation come from measurements on a real system, they may be quantized
- in the simulation, there may be mixture distributions (that are a mixture of discrete and absolutely continuous distributions)

THE ACCURACY OF THE ESTIMATION

As it was mentioned before, the parametric estimation cannot be used, as the distributions to be estimated are unknown. However, we can test the different estimation methods for some frequently used distributions. It gives us an impression of the behaviour of the accuracy of the method, that help us to choose the method to be used in a particular simulation.

Exponential Distribution

The exponential distribution occurs many times in simulations as the distribution of the inter-arrival time of requests. It can be characterized by its probability density function:

$$f(x) = \lambda e^{-\lambda x}$$

The value of the λ parameter was chosen to be 1 in the experiments. What is the best L_1 error that can be achieved when the estimation is done by N number of observations with the optimal choice of the parameters of the studied statistics collection methods?

Equidistant Histogram

Besides the N number of the collected samples the L_1 error depends on the parameters of the histogram. The t_0 should be 0. Simulation experiments were executed with different values for the h_N size and the M_N number of bins. The N number of samples was chosen to be 1000 and the experiments were repeated for 1000 times to smooth the diagram.

Method	Computation	Storage	Communication
relative frequency	O(N*log(M)) or O(N*log(M')+(M')^2)	M*(`sizeof(int)+sizeof(double)`) or M'*(`sizeof(int)+sizeof(double)`)	M*`sizeof(int)` or M'*(`sizeof(int)+sizeof(double)`)
equidistant histogram	O(N)	M*`sizeof(int)`	M*`sizeof(int)`
Barron estimate	O(N)	M*`sizeof(int)`	M*`sizeof(int)`
equiprobable bin histogram	O(N*log(M))	M*(`sizeof(int)+sizeof(double)`)	M*(`sizeof(int)+sizeof(double)`)
semi-equiprobable bin histogram	O(N*log(N))	N*`sizeof(double)`+ M*(`sizeof(int)+sizeof(double)`)	M*`sizeof(double)` or M*(`sizeof(int)+sizeof(double)`)

Table 1. The resource requirements of the studied distribution estimation methods.

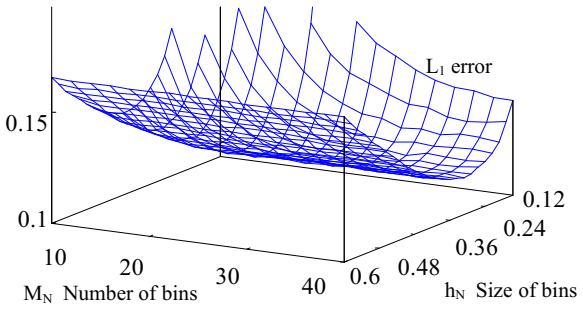

Figure 3. L_1 error of the equiprobable histogram estimation of 1000 samples from exponential distribution in the function of the h_N size and the M_N number of bins

Figure 3 shows that for a given M_N number of bins there is an h_{Nopt} optimal value for the bin size. If $h_N > h_{Nopt}$ then the resolution is too coarse that causes the error to grow as h_N increases. If $h_N < h_{Nopt}$ then the number of observations per bin is too small (there is too much fluctuation) that causes the error to grow as h_N decreases. To optimize the parameters the value of h_N was changed while the value M_N was always set so that the range of the histogram be large enough, that is:

$$M_N = \left\lceil \frac{7}{\lambda h} \right\rceil$$

The $7/\lambda$ range is enough, because in this case the measure of the tail of the exponential distribution is less than 0.001, which is negligible compared to the measured values of the L_1 error.

h_N	0.13	0.17	0.21	0.25	0.29	0.33	0.37
E(L1)	0.1393	0.1253	0.1194	0.1166	0.1173	0.1193	0.1242
σ(L1)	0.0178	0.0164	0.0154	0.137	0.0120	0.0105	0.0100

Table 2. The L_1 error in the function of the bin size (exponential distribution, 1000 samples, 1000 experiments)

As it can be seen from table 1, a good choice for h_N is 0.25. It also can be seen that a small change in h_N does not cause a significant change in the L_1 error. The number of bins (M_N) was chosen to be 20, because for fewer bins the L_1 error is significantly worse and for more bins the L_1 error is not significantly better.

The size and number of bins were determined in the case of some other number of samples collected. Table 3 shows the results. Note that the range of histogram collection ($M_N * h_N$) is in the order of $5/\lambda$.

N	500	1000	2000	4000	8000	16000	32000
M_N	17	20	24	33	41	56	67
h_N	0.30	0.25	0.20	0.15	0.12	0.10	0.08
E(L1)	0.1444	0.1166	0.0952	0.0765	0.0627	0.0485	0.0402
σ(L1)	0.0188	0.0136	0.0090	0.0067	0.0049	0.0035	0.0025

Table 3. The L_1 error of the equidistant histogram in the function of the no. of observations (exponential distribution, optimal bin no. and size)

Barron Estimate

Of course, if $G(x) \equiv F(x)$ is chosen, then with $M_N = 1$ cell the method produces zero L_1 error, so we must take care to be honest with the choice of the G(x) function. Let us consider the following three examples:

$$G_1(x) = F(\frac{x}{2}),\quad G_2(x) = \frac{2}{5\lambda}x - \frac{1}{25\,\lambda^2}x^2,\quad G_3(x) = \frac{2}{3\lambda}x - \frac{1}{9\,\lambda^2}x^2$$

The first one is also an exponential distribution, but with $\lambda/2$ parameter, the second and third ones are linear approximations concerning the density function. (See Figure 4.)

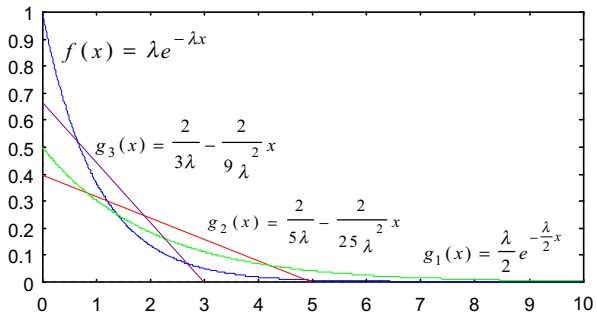

Figure 4. Density functions for Barron estimate

The optimal value for M is determined in the usual way, and the results for the L_1 error are shown in Table 4.

G	$G_1(x)$		$G_2(x)$		$G_3(x)$	
N	1000	8000	1000	8000	1000	8000
M_N	9	20	10	23	5	14
$E(L_1)$	0.0865	0.0438	0.1028	0.0588	0.1716	0.1402
$\sigma(L_1)$	0.0141	0.0062	0.0127	0.0052	0.0133	0.0053

Table 4. The L_1 error of the Barron estimate for different G(x) functions (exponential distribution, optimal bin number)

Barron estimate produced better results with both $G_1(x)$ and $G_2(x)$ than the equidistant histogram. However, if we do not have an appropriate a priori knowledge of the distribution then this method is not applicable, $G_3(x)$ is a negative example.

Equiprobable Bin Histogram

The $[0, 5/\lambda)$ interval is used for statistics collection and the bin boundaries are set according to the following: The [0,1) interval is divided into M_N equal intervals. The boundaries of the intervals are transformed by the modified inverse distribution function, so that the last bin boundary fall to $5/\lambda$:

$$F^{-1}(x) = \frac{-\ln(1 - xe^{-5})}{\lambda}$$

The M_N number of cells is optimized in the following way: M_N is increased until the further increment of M_N does not have a significant effect on the L_1 error. Table 5 shows the results for some values of N, the number of observations.

N	500	1000	2000	4000	8000	16000	32000
M_N	9	12	15	19	24	30	40
$E(L_1)$	0.1871	0.1495	0.1218	0.0989	0.0803	0.0656	0.0529
$\sigma(L_1)$	0.0201	0.0157	0.0105	0.0073	0.0054	0.0039	0.0028

Table 5. The L_1 error of the equiprobable bin histogram in the function of the number of observations (exponential distr., optimal bin number)

The idea of the equiprobable histogram gave the hope of a good solution for the bins with too few observations and therefore much uncertainty and error, but our results show that the value of the L_1 error is higher for the equiprobable bin histogram than for the equidistant histogram. This is quite surprising and unexpected. To check our simulation results, let us compare the L_1 error of the ideal equidistant and equiprobable bin histograms. Here, "ideal" means that the histograms are constructed on the basis of the probability density function of the exponential distribution instead of on the basis of finite number of observations. The histograms are constructed for the $[0, 5/\lambda)$ range, the L_1 error contains the $e^{-5} \approx 0.0067$ error caused by the loss of the $[5/\lambda, \infty)$ tail of the distribution. Table 6 shows the results, that justify the intuition: the equiprobable bin histogram has less L_1 error, but the difference becomes less and less as the number of bins (M) increases.

M	2	4	8	16	32	64
L_1 of EqD	0.5801	0.3106	0.1611	0.0842	0.0455	0.0261
L_1 of EqP	0.5282	0.2846	0.1523	0.0818	0.0450	0.0260

Table 6. The computed L_1 error of the ideal equidistant (EqD) and equiprobable bin (EqP) histograms in the function of the number of bins (M) in the case of exponential distribution.

The results of the simulation experiments for the same number of bins and for N=8000 number of observations are shown in Table 7. The experiments were executed 1000 times; average and standard deviation were calculated. Results show that in the case of the histograms that are built upon finite number of observations, the equiprobable bin histogram produces better results than the equidistant one for very little number of bins only and for larger number of bins the equidistant one performs better. The question "why" we leave open for mathematicians.

M	2	4	8	16	32	64
L_1 of EqD	0.5802	0.3109	0.1625	0.0898	0.0634	0.0676
$\sigma(L_1)$	0.0002	0.0003	0.0008	0.0019	0.0042	0.0062
L_1 of EqP	0.5283	0.2853	0.1557	0.0948	0.0769	0.0841
$\sigma(L_1)$	0.0014	0.0013	0.0022	0.0043	0.0060	0.0063

Table 7. The L_1 error of the equidistant (EqD) and equiprobable bin (EqP) histograms in the function of the number of bins (M) built by collecting N=8000 observations and repeating the experiments 1000 times in the case of exponential distribution.

Semi-equiprobable Bin Histogram

The number of bin used were not optimised but they were taken from the equiprobable bin histogram. The results for some values of N are shown in Table 8. This method gives approximately the same L_1 error as the "more perfect" equiprobable bin histogram. (Of course the L_1 error is not less, see the value of standard deviation!)

N	500	1000	2000	4000	8000	16000	32000
M_N	9	12	15	19	24	30	40
$E(L_1)$	0.1845	0.1482	0.1227	0.1019	0.0853	0.0716	0.0588
$\sigma(L_1)$	0.0216	0.0164	0.0113	0.0083	0.0059	0.0042	0.0030

Table 8. The L_1 error of the semi-equiprobable bin histogram in the function of the number of observations (exponential distribution, bin numbers are equal with that of the equiprobable bin histogram)

Gamma Distribution

The sum of n number of exponential distributions with parameter λ is a gamma distribution with parameters (n, λ). Its probability density function is:

$$f(x;\lambda,n) = \begin{cases} \dfrac{\lambda^n}{(n-1)!} x^{n-1} e^{-\lambda x}, & x > 0 \\ 0, & x \le 0 \end{cases}$$

For our experiments, let us choose n=4 and λ=1 The random numbers of the gamma distribution are generated according to the definition: the sum of n=4 random numbers of exponential distribution with λ=1 parameter. To avoid the two dimensional optimisation, the range of the statistics collection is the [0, 11) interval omitting the [11, ∞) tail of the distribution, the measure of which is less than 0.005.

Equidistant Histogram

The optimal bin number (M_N) was determined for all the values of N used. The bin size is $h_N = M_N/11$. The experiments were repeated 1000 times. Table 9 shows the results.

Because of space limitations we have to omit the Barron estimate and the equiprobable bin histogram. They are not the methods of choice in the practical case as we do not know the distributions to be estimated.

N	500	1000	2000	4000	8000	16000	32000
M_N	13	15	19	24	28	37	45
$E(L_1)$	0.1012	0.0839	0.0678	0.0551	0.0459	0.0369	0.0308
$\sigma(L_1)$	0.0174	0.0122	0.0085	0.0060	0.0042	0.0031	0.0022

Table 9. The L_1 error of the equidistant histogram in the function of the number of observations (gamma distribution, optimal bin number)

Semi-equiprobable Bin Histogram

The optimal bin number (M_N) was determined for all the values of N used. Table 9 shows the results. Comparing them with the results of the equidistant one, we can see that the equidistant one produces about 20-25% less L_1 error. If the range of the possible values is not known in advance and storage allows, it may be worth storing the observations. Then we can set up the range of the histogram on the basis of the observations, collect the equidistant histogram and send it to the appropriate segment. There is no justification to use the semi-equiprobable bin histogram.

N	500	1000	2000	4000	8000	16000	32000
M_N	12	16	20	23	28	37	45
$E(L_1)$	0.1279	0.1060	0.0888	0.0758	0.0649	0.0535	0.0449
$\sigma(L_1)$	0.0221	0.0168	0.0098	0.0080	0.0053	0.0039	0.0030

Table 10. The L_1 error of the semi-equiprobable histogram in the function of the number of observations (gamma distribution, optimal bin number)

Packet Length Distribution in an FDDI Backbone

To examine a practical case, let us consider the distribution of the length of the packets in a physically existing network. The selected network is the FDDI backbone of the Technical University of Budapest.

The data were acquired from the so-called Northern Ring by a protocol analyser (Lencse 1997). Figure 5 shows a rough overview of the distribution. This is a discrete distribution of integer values in the range [61-1521], the number of the possible values is 1461.

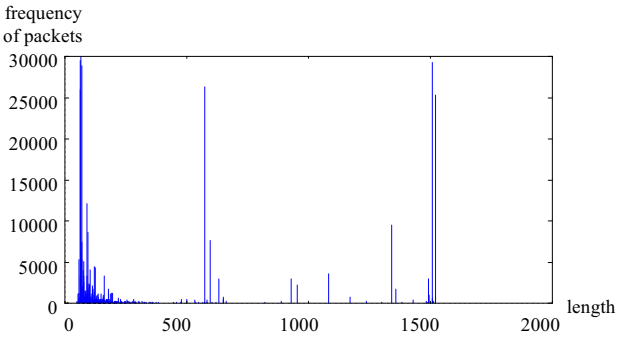

Figure 5. The distribution of the packet length in an FDDI backbone. (The column at length=67 is truncated, its height is about 130000.)

As we do not know the distribution itself, the relative frequencies of the values in the N=500..32000 size samples are compared to the relative frequencies of the values in the whole sample (500000 observations). The N sized samples are chosen randomly. Table 11 shows the results.

N	500	1000	2000	4000	8000	16000	32000
$E(L_1)$	0.3874	0.2978	0.2242	0.1691	0.1246	0.0904	0.0654
$\sigma(L_1)$	0.0230	0.0189	0.0137	0.0093	0.0062	0.0048	0.0026

Table 11. The L_1 error of the relative frequency estimation (packet length distribution in an FDDI backbone)

In the case of this distribution, the value of the L_1 error is quite high for the smaller values of N, but the convergence speed in nearly $c_1/N^{1/2}$. This is faster than the convergence speed experienced in the case of the other distributions that produce the rate of $c_2/N^{1/3}$, which is the guaranteed one for histograms.

Inter-Arrival Time Distribution in an FDDI Backbone

The inter arrival time of the packets in the before mentioned network was also observed. Unfortunately, the protocol analyser recorded the time data with 0.00001s accuracy only. As FDDI is a 100Mbit/s network, this 10μs is the time of 1000 bits. In this way the inter-arrival time is a quantized random variable, so we use relative frequency estimation.

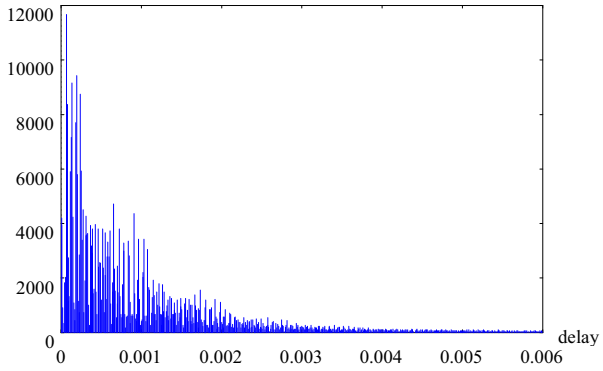

Figure 6. The distribution of the inter-arrival time ("delay", measured in seconds) in an FDDI backbone

Figure 6 shows an empirical picture of a part the distribution only, the distribution is not limited to the [0, 0.006] interval. The envelope of the distribution reminds us the exponential distribution. The range of the statistics collection is [0, 0.01] and the measure of the omitted tail is 0.012, which is a significant part of the L_1 error for the samples of size greater than 2000. Results are shown in Table 12.

N	500	1000	2000	4000	8000	16000	32000
$E(L_1)$	0.1911	0.1478	0.1160	0.0918	0.0702	0.0552	0.0438
$\sigma(L_1)$	0.0243	0.0171	0.0113	0.0084	0.0061	0.0049	0.0034

Table 12. The L_1 error of the relative frequency estimation (packet length distribution in an FDDI backbone)

CONCLUSION

A number of statistics collection methods were compared and applied in the case of different distributions in order to determine what estimation methods should be used for statistics collection for the Statistical Synchrinisation Method. Both their resource requirements and accuracy were examined. Their resource requirements must be taken into consideration, not to slow down the simulation, but their accuracy is even more important to safeguard the accuracy of the simulation results.

The L_1 error criteria was chosen for measuring the error of the estimation methods.

Barron estimate may produce the smallest L_1 error among all the distribution estimation methods, but it requires an a-priori information of the distribution that is not available in the general case so the method cannot be used in general.

Theoretically, the equiprobable bin histogram should produce less L_1 error than the equidistant one, and it is also the experience in the case when very small number of bins are used, but if there are enough bins, the equidistant histogram is better. For continuous distributions, the equidistant histogram is the method of choice.

The relative frequency method produced acceptable results for the examined real life discrete or quantized distributions.

ACKNOWLEDGEMENT

The statistics collection tools of the OMNeT++ discrete-event simulator were used for statistics collection. See its home page for more information (Varga 1998).

REFERENCES

Aho, A. V.; J. E. Hopcroft; J. D. Ullman. 1975. *The Design and Analysis of Computer Algorithms*, Addison-Wesley

Barron, A. R.; L. Gyôrfi; E. C. Meulen. 1992. "Distribution Estimation Consistent in Total Variation and in Two types of Information Divergence" *IEEE Transactions on Information Theory* Vol 38, No. 5, September 1992, pp. 1437-1454.

Devroye, L.; L. Gyôrfi. 1985. *Nonparametric Density Estimation: The L1 view*. John Wiley, New York

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

Jain, R.; C. Imrich. 1985. "The P^2 Algorithm for Dynamic Calculation of Quantiles and Histograms Without Storing Observations". *Communications of the ACM* 28, no. 10 (Oct.) pp. 1076-1085.

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. 1997. "Efficient Simulation of Large Systems - Transient Behaviour and Accuracy" *Proceedings of the 1997 European Simulation Symposium (ESS'97)* (Passau, Germany, Oct. 19-23). SCS Europe, pp. 660-665.

Lencse, G. 1998. "Efficient Parallel Simulation with the Statistical Synchronization Method" *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation (CNDS'98)* (San Diego, CA. Jan. 11-14). SCS International, pp. 3-8.

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.) SCS Europe, pp. 125-129.

Scott, D. W. 1992. *Multivariate Density Estimation* John Wiley & Sons, Inc.

Varga, A; B. Fakhamzadeh. 1997. "The K-Split Algorithm for the PDF Approximation of Multi-Dimensional Empirical Distributions without Storing Observations". *Proceedings of the 9th European Simulation Symposium (ESS'97)* (Oct. 19-22 1997, Passau, Germany.) SCS Europe, pp.94-98.

Varga, A. 1998. "OMNeT++ Discrete Event Simulation System" `http://www.hit.bme.hu/phd/vargaa/omnetpp.htm`

BIOGRAPHY

Gábor Lencse was born in Gyôr, 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 (parallel) discrete event simulation. Since 1997, he works for the Széchenyi István College in Gyôr. He teaches computer networks. He is a member of the Society for Computer Simulation International.