Nuclear Instruments and Methods in Physics Research A278 (1989) 431-435
North-Holland, Amsterdam

DEAD TIME DUE TO TRIGGER PROCESSING IN A DATA ACQUISITION SYSTEM WITH MULTIPLE EVENT BUFFERING

G.P. HEATH

Department of Nuclear Physics, University of Oxford, UK

Received 14 June 1988 and in revised form 16 January 1989

A formalism is developed to allow the common treatment of dead time in multistage data acquisition systems under differing assumptions on the processing time distribution. The formulae derived are used to compare dead times assuming random and constant distributions of processing time.

1. Introduction

It seems clear that for the foreseeable future colliding-beam accelerators at higher and higher centre-of-mass energies will dominate the subject of high-energy particle physics. The experiments being developed for the accelerators presently under construction, such as LEP, HERA, SLC and the Tevatron Collider, and those proposed for possible future machines such as the SSC, are almost all large, general-purpose detectors with many thousands of electronics channels. The triggering and data acquisition (DAQ) problems associated with these large experiments are extremely complex, and careful design and modelling of these systems is essential if the desired performance is to be achieved at an affordable price [1]. Among the experiments presently under construction, the H1 [2] and ZEUS [3] experiments under construction for HERA have a particularly difficult problem in that the extremely short separation of 96 ns between bunches means that the data must be stored for several beam crossings until a first-level trigger (FLT) decision can be made. The ZEUS experiment, for example, proposes to deal with the large quantity of data output by the Central Tracking Detector by having significant processing power on the front-end electronics as well as at the higher levels of the system [4]. Further buffering is required at each level to allow sufficient time to process the events and bring the trigger rate down to a sufficiently low level to be recorded on tape.

It is clearly of vital importance to be able to calculate the dead-time losses at various points in multi-buffering, parallel-processing environments such as the above in order to optimise the processing power and number of buffers at each stage. Existing literature on dead-time estimation, in the context of dead times due to instrumental effects in counters and electronics [5,6], does not consider systems with buffering. However, it has long been realised in high-energy physics that statistical queueing theory [7] can be applied to the problem, and a "standard formula" for the dead time as a function of buffer capacity is widely known [1] (see section 4 below). It is often forgotten that this standard formula in fact depends on assumptions about the distributions of event arrival times and processing times in the system. In this paper we develop a notation which allows different assumptions on the statistical distribution of processing times to be treated within the same formalism. We have attempted to derive the results presented here from first principles, assuming no familiarity with the subject, and considering in detail only the cases which are of interest in high-energy physics. In particular results are presented for the case of constant processing time, which is likely to be more applicable to front-end processors, used for the compaction of large amounts of sparse data, than the usual assumption of a random processing-time distribution. In section 2 we develop the required notation, while in sections 3 and 4 it is applied to two particularly useful assumptions on the processing-time distribution. Section 5 examines some results from this formalism and contains the conclusions.

2. Development of the general formalism

We consider a two-level trigger scheme in which the first-level trigger accepts events with a rate R and the mean second-level processing time is τ. Buffer space for N events, in addition to the event currently being processed, is to be provided at the input to the second level. We assume that the time required for the second-level processors to process a particular event is unaffected by the operation of the first-level trigger and buffer system. We wish to calculate the dead-time fraction for this system as a function of N, which we denote by d_N. For $N=0$, i.e., where there is no multiple buffering, a dead time is associated with each FLT equal to the time required for processing at the second level. This clearly places a requirement on the second-level processing that the mean time required to process an event must be only a small fraction of the mean interval between incoming events in order to avoid large dead-time losses. We write $x = R\tau$ and the dead-time fraction associated with second-level processing is then

$$d_0 = \frac{\tau}{\frac{1}{R} + \tau},$$

or

$$d_0 = \frac{x}{1+x}.$$

We now consider a system which has multiple buffering, allowing incoming FLTs to be queued for later servicing while the second-level trigger (SLT) is processing previous FLTs. If the number of events which can be queued in this way is infinitely large, or the maximum allowed queue length infinite, the mean time for second-level processing can be as long as the mean interval between FLTs without incurring dead-time losses. If the processing time is longer the queue will grow without limit, and the FLT processing will eventually have to be switched off for some fraction of the time to allow the SLT to catch up. In terms of the parameter $x = R\tau$, the fractional dead time for infinite buffer capacity is, therefore,

$$d_\infty = \begin{cases} 0 & x \le 1, \\ \dfrac{x-1}{x} & x > 1. \end{cases}$$

Where the buffer capacity, or maximum event-queue length, is finite, however, additional dead time will occur when an event arrives which saturates the queue. In this case the FLT is not re-enabled until the completion of processing of the event currently being serviced by the SLT, which frees buffer space for the acceptance of a new event. Since in particle physics we are interested in maximising the allowed processing time for a given input trigger rate (i.e. in working close to $x = 1$), we would like to know what buffer capacity approximates sufficiently closely to the infinite case discussed above, so that we require an estimate of the fractional dead time due to second-level processing as a function of x and of buffer capacity N. As the process is clearly statistical in nature, its analysis will in general require consideration of the distributions both of arrival times of FLTs and of second-level processing times. The FLT arrival-time distribution will be random in most cases, but different distributions of processing times can be envisaged. In the following we assume that the FLT processing remains active, so that events will continue to arrive, so long as at least one buffer is available, and consider the behaviour of the second-level processors. When the processing of an event finishes and further events are queued, the next event is removed from the queue and input to the SLT. If the queue is empty, the second-level processors are idle until the next FLT event arrives at the input. Suppose that each time an event is input to the processors, the queue length m is recorded. The value of m will vary from 0 up to $N-1$ since, although the maximum length is N, one event will have been removed from the queue immediately beforehand. By the time the next event is input to the processors, the queue length will be between $m-1$ and $N-1$, depending on the number of events which have arrived from the FLT during the processing of this one event. We denote by a_j the probability for j events to arrive, which may be derived from the arrival-time and processing-time distributions. These probabilities therefore determine the probability for observing a particular sequence of values of the length m as successive events are input to the SLT. In the following, we will use this fact to relate the frequencies with which the different possible values of m are observed, and hence find the dead time fraction d_N.

The total rate of processing events in the SLT is the sum of the numbers of events per unit time where the queue length observed at the start of processing takes all possible values. Let the rate at which a particular length m is observed be r_m. For an event where the value m is observed, the queue length at the start of processing of the previous event must have been $m+1$ or less, giving for the different rates r_m the relationship

$$r_m = \sum_{j=0}^{m+1} a_j r_{m-j+1}, \quad m = 1, 2, \cdots, N-2. \tag{1}$$

The case of $m=0$ has to be considered separately. This value is observed following an idle period, as remarked above. The rate of occurrence of such idle periods is given by r_0 times the probability that no evident arrives from the FLT during processing, which gives rise to the last term in the expression

$$r_0 = a_0 r_1 + a_1 r_0 + a_0 r_0. \tag{2}$$

The fraction of time for which the processors are idle is $a_0 r_0 / R$; the set of N equations for the r_m is completed using the condition that the total time is made up of processing time and idle time,

$$\frac{a_0 r_0}{R} + \tau \sum_{m=0}^{N-1} r_m = 1,$$

where τ is the mean processing time per event as before, or

$$a_0 r_0 + x \sum_{m=0}^{N-1} r_m = R. \tag{3}$$

Since the total second-level processing rate must be equal to the input rate, after correction for dead time, we have

$$R(1-d_N)=\sum_{m=0}^{N-1} r_m. \tag{4}$$

Eqs. (3) and (4) give the dead time fraction d_N in terms of the rates r_m as:

$$d_N = 1-\frac{\sum_{m=0}^{N-1} r_m}{a_0 r_0 + x\sum_{m=0}^{N-1} r_m}. \tag{5}$$

At this point it is convenient to define a new sequence of terms S_m by

$$S_m=\frac{1}{a_0 r_0}\sum_{j=0}^{m-1} r_j. \tag{6}$$

Eqs. (1) and (2) may now be combined in the form

$$S_m=\sum_{j=0}^{m} a_j S_{m-j+1},\quad m=1,2,\cdots,N-1, \tag{7}$$

and rearranging, we obtain for S_m the recurrence relation

$$a_0 S_{m+1}=S_m-\sum_{j=1}^{m} a_j S_{m-j+1}. \tag{8}$$

In terms of the S_m, eq. (5) for d_N now becomes

$$d_N=1-\frac{S_N}{1+xS_N}. \tag{9}$$

From the definition (6) it is obvious that $S_0=1/a_0$, so that the recurrence relation (8) may be used to find S_N and hence d_N for any set of a_j. To apply the formalism for the trivial case of $N=0$ requires the value of S_0 which is so far undefined. However if we require eq. (7) to hold also for $m=0$, this gives $S_0=1$ and so

$$d_0=\frac{x}{1+x},$$

as before.

3. Random arrival time, constant processing time

We now proceed to use the above formalism to solve for S_m and therefore the fractional dead time using eq. (9) for the case of constant processing time. Since the analytic solution for this case is difficult to find in the literature, we give the derivation in some detail. The a_j distribution of the number of events arriving during processing is just the Poisson distribution

$$a_j^{\text{const}}=e^{-x}\frac{x^j}{j!}. \tag{10}$$

The first few solutions S_m may be derived by repeated application of the recurrence relation (8); $S_1=1/a_0=e^x$, $S_2=e^{2x}(1-x\,e^{-x})$, and so on. We have found by inspection that these solutions are described by the general form

$$S_m=e^{mx}P_m(y), \tag{11}$$

where P_m is a polynomial in $y=-x\,e^{-x}$,

$$P_m=\sum_{j=0}^{m-1}\frac{(m-j)^j y^j}{j!}. \tag{12}$$

The proof that eqs. (11) and (12) give the correct solution for all m proceeds by the method of induction. We substitute eqs. (10) and (11) into eq. (8) to give

$$e^{mx}P_{m+1}=e^{mx}P_m-\sum_{j=1}^{m}\frac{x^j}{j!}e^{(m-j)x}P_{m-j+1},$$

$$P_{m+1}=P_m-\sum_{j=1}^{m}\frac{(-y)^j}{j!}P_{m-j+1}. \tag{13}$$

Now we rewrite eq. (12) to give

$$P_{m-j+1}=\sum_{i=0}^{m-j}\frac{(m-j-i+1)^i y^i}{i!},$$

and substituting this into eq. (13) gives

$$P_{m+1}=P_m-\sum_{j=1}^{m}\sum_{i=0}^{m-j}\frac{(m-j-i+1)^i y^i(-y)^j}{i!j!}$$
$$=P_m-\sum_{j=1}^{m}\sum_{i=0}^{m-j}\frac{y^{i+j}}{i!j!}[m-(i+j)+1]^i(-1)^j.$$

Rearranging the double sum to group terms of the same $k=i+j$ leads to

$$P_{m+1}=P_m-\sum_{k=1}^{m}\frac{y^k}{k!}$$
$$\times\sum_{i=1}^{k}\frac{k!}{i!(k-i)!}(m-k+1)^i(-1)^{k-i}.$$

The second sum now looks like a binomial expansion, except that it should extend from $i=0$ to k. We can therefore write

$$P_{m+1}=P_m-\sum_{k=1}^{m}\frac{y^k}{k!}\left[(m-k)^k-(m-k+1)^k\right],$$

and on substituting for P_m from eq. (12), most of the terms in $(m-k)^k$ cancel, leaving

$$P_{m+1}=\sum_{k=1}^{m}\frac{(m-k+1)^k y^k}{k!}+1-0,$$

where the last two terms are the remaining $(m-k)^k$ terms with $k=0$, m, respectively, or

$$P_{m+1}=\sum_{k=0}^{m}\frac{(m-k+1)^k y^k}{k!},$$

as required, which completes the proof. The solution for S_m is therefore

$$S_m^{\text{const}} = e^{mx} \sum_{j=0}^{m-1} \frac{(m-j)^j(-x\,e^{-x})^j}{j!}. \tag{14}$$

4. Random arrival time, random processing time

Another useful model for the processing times is to assume that they are randomly distributed (i.e. exponentially distributed with mean τ). To find the a_j in this case we convolute this Poisson distribution for the number of events arriving in a time τ', which has a mean of $x' = R\tau'$, with an exponential distribution for x':

$$a_j^{\text{ran}} = \frac{1}{x}\int_0^\infty \frac{x'^j}{j!} e^{-x'(1+x/x)}\,dx'$$
$$= \frac{1}{1+x}\left(\frac{x}{1+x}\right)^j.$$

The expression for S_m^{ran} can be shown to be

$$S_m^{\text{ran}} = \sum_{j=0}^{m} x^j = \frac{1-x^{m+1}}{1-x}, \tag{15}$$

by induction, giving the "standard" queuing theory expression [1]

$$d_N^{\text{ran}} = \frac{x^{N+1}(1-x)}{1-x^{N+2}}. \tag{16}$$

5. Results and conclusions

The variation of d_N^{ran} and the expression for constant processing time d_N^{const} with x is shown in figs. 1 and 2 respectively for x values from 0 to 2 and various N. The curves for d_0 and d_∞ are also plotted for comparison. The qualitative behaviour of the two functions is similar, although for constant processing time the ideal case of infinite buffer capacity is approached significantly faster with increasing N. The two expressions are compared directly for $N = 3$ in fig. 3, from which it will be seen that in the region of $x = 1$, d_3^{const} is around half of d_3^{ran}; the values for $x = 0.8$, for example, are $d_3^{\text{const}} = 5.9\%$, $d_3^{\text{ran}} = 12.2\%$. The generally accepted conclusion based on the expression for random processing time is that a modest buffer depth of 3 or 4 events is sufficient to allow processing to take up to 80–90% of the available time without introducing large dead-time losses. It can be seen from the above results that this is certainly also adequate for the case of processing times which are more nearly constant.

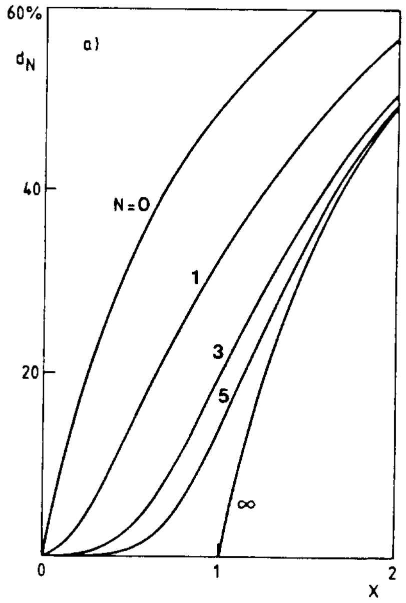

Fig. 1. Variation of fractional dead time d_N with rate parameter x for different values of N for a random processing-time distribution.

Some general conclusions on the design of triggering and DAQ systems can be drawn from the results presented here. The formalism developed allows a realistic treatment of dead times in multibuffered systems for a variety of assumptions on the distribution of processing times. Indeed the dead time equations can in principle

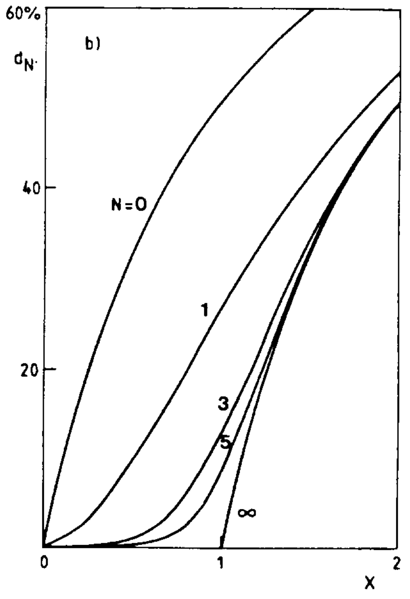

Fig. 2. Variation of fractional dead time d_N with rate parameter x for different values of N for a constant processing-time distribution.

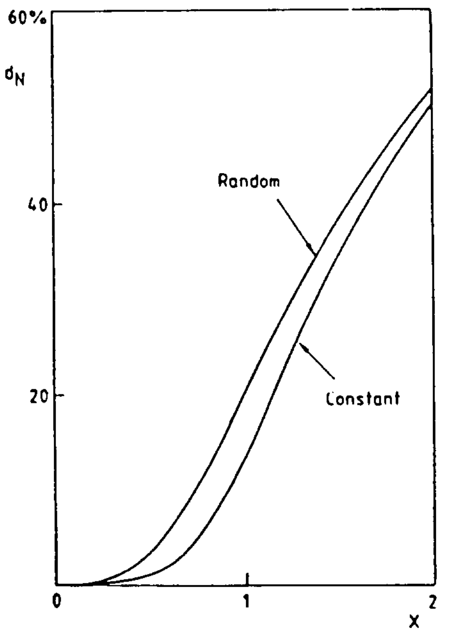

Fig. 3. Comparison of d_3 for the two processing-time distributions.

be solved, either analytically or if necessary by numerical methods, for any form of the processing-time distribution. This treatment gives a means of estimating for a given DAQ configuration the necessary buffer depth or the best way to trade off buffers against processing power. A general and well known conclusion which can be made irrespective of the distribution of processing times assumed is that if the input rate to any level is significantly in excess of the value assumed in optimising the buffering, very large dead-time losses occur. Thus it is essential that the trigger reduction factors quoted for each level in a DAQ system are realistic.

In the next generation of hadron–hadron colliders, such as SSC or LHC, the beam crossing rates and luminosities are extremely large, leading to very large interaction and trigger rates as well as enormous quantities of data [8]. Analytic results such as those derived here should facilitate the design of trigger and DAQ systems for experiments at these machines by allowing quick checks of dead times for various configurations. However, an important proviso is that although the formulae derived here can be used to estimate the fractional dead time for each processing stage treated in isolation, the details of interactions among the different stages may lead to modifications and correlations of the arrival-time distributions at these stages. It is extremely difficult to treat these correlations in an analytic form. In order to obtain reliable dead-time estimation in a multi-stage DAQ system, a realistic Monte Carlo simulation should be set up, as for some of the LEP experiments, to model the full DAQ system architecture.

References

[1] See for example Ph. Gavillet, Nucl. Instr. and Meth. A235 (1985) 363.
[2] H1 Collaboration, Technical Proposal (March 1986) unpublished.
[3] ZEUS Collaboration, Technical Proposal (March 1986) unpublished.
[4] B. Foster, University of Bristol Preprint BRP/HEP 87-1, to be published in Proc. 10th Warsaw Symp. on Elementary Particle Physics, Kazimierz, Poland (1987).
[5] A. Handbook of Radioactivity Measurements Procedures, NCRP Report no. 58, Washington (1978).
[6] J.W. Müller, Nucl. Instr. and Meth. 112 (1973) 47.
[7] A.O. Allen, Probability, Statistics and Queueing Theory (Academic Press, London, 1978).
[8] J.R. Hansen et al., Proc. Workshop on Physics at future accelerators, La Thuile, CERN 87-07 (1987) p. 274.