An analysis of real and simulated statistics for system design purposes

By R. Grimmond

Experimental observations of errors in data-transmission circuits are the basis for a statistical model which generates errors artificially by computer program. These errors are used in a variety of programs which simulate error-detection systems with a view to assessing their performance. This paper is based on a talk given to The British Computer Society in London on 1 February 1962.

1. Introduction

The behaviour of certain systems of engineering equipment can only be analysed in statistical terms. The particular system studied in this paper is that of data transmission over telephone lines, but the general principles are applicable to other systems as well. A major aspect of the design of a data-transmission system is the measures taken to combat errors. Generally the quantity that is of greatest interest is the overall effectiveness of a particular error-detection system in order that a comparison can be made with other possible schemes. A method of arriving at this measure of effectiveness will be described; it involves both real and simulated statistics about the errors which may be found on typical telephone connections. The processing of the statistics and the simulation of the protective systems are jobs for which the speed and flexibility of an electronic computer are eminently suitable. The computer used for these studies is Stantec Zebra. The programs for it have been written directly in machine code which lends itself well to this type of problem.

2. The Collection of Real Statistics

An experimental equipment has been used to record the errors in transmission on a variety of lines. A block schematic is shown in Fig. 1. It consists of a sending and a receiving terminal side by side. Information is transmitted from the sender both to a looped telephone line and to a variable delay, which is adjusted to suit the loop under test. The received information is compared element by element with that emerging from the variable delay. A discrepancy at the comparator is regarded as an error and is recorded as such. The error recording is a description not only of the line, but its effect upon the terminal modulating equipments (frequency-shift modulation). The variation of line delay with time is so small compared with the element time that its effect may be disregarded.

The errors are recorded on five-hole punched paper tape, suitable for computer input, by a high-speed punch, permitting transmission speeds up to 1,200 bauds provided all five holes are used for the recording. In order to conserve paper and to produce manageable records, the transmission is arranged in 2,000-bit cycles. After the transmission of every cycle the equipment makes available a serial number. If one or more errors are found during the cycle period, these are recorded as holes in the appropriate positions, and the tape continues to step on until the end of the cycle when the serial number is punched in binary-coded-decimal form inside a distinguishing frame of holes (see Fig. 2). By working back from the serial number on the tape the exact position of the errors in a cycle can be determined. The

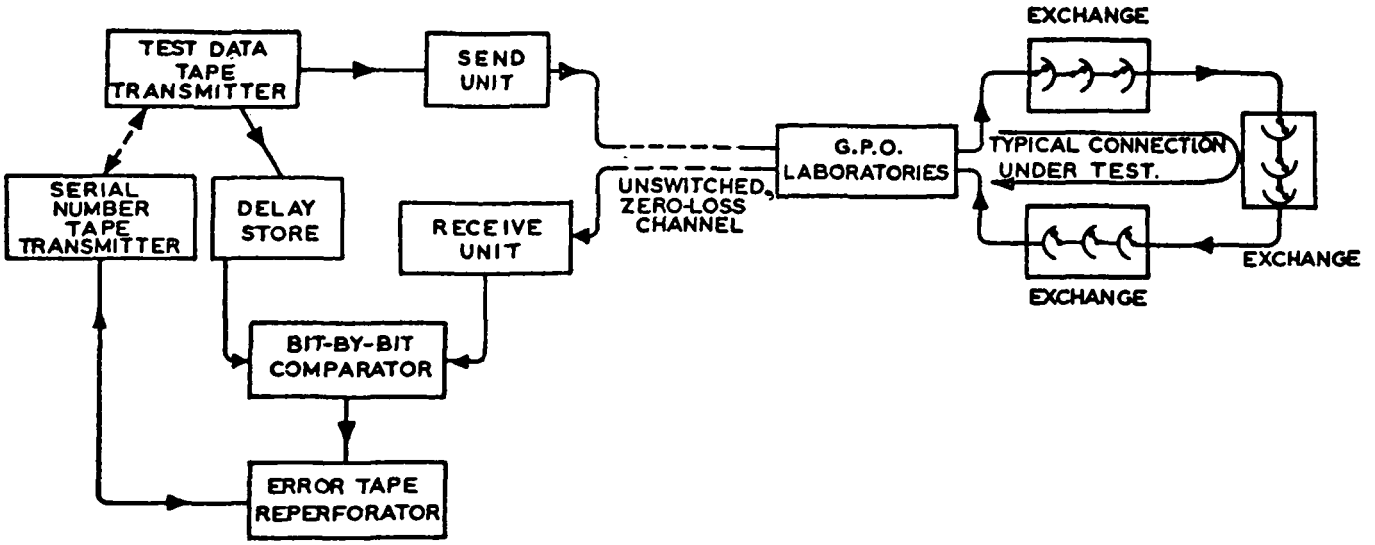

Fig. 1.—Schematic diagram of test installation

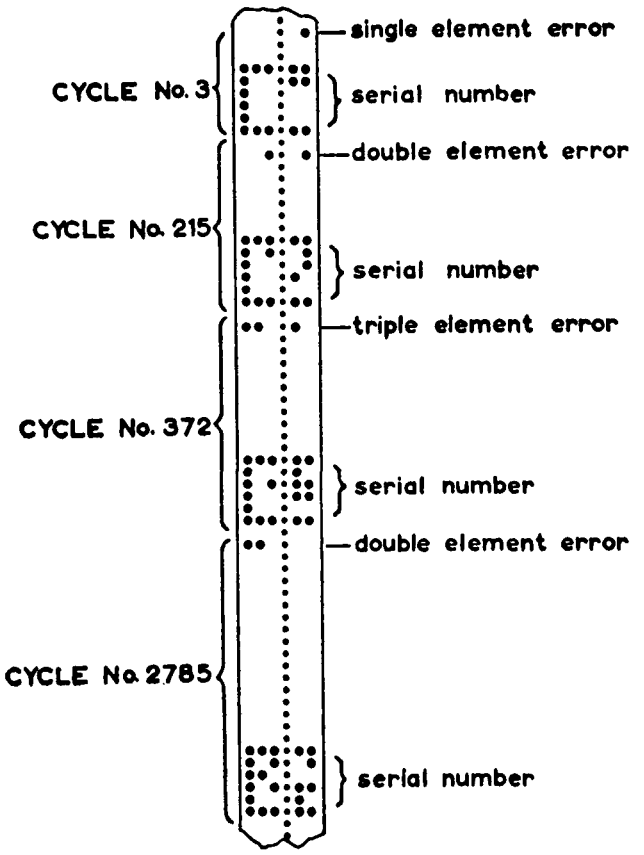

Fig. 2.—Typical error tape

chance that an error pattern may be interpreted as a serial number is very small, since the serial numbers represent definite *information*, in contrast with the error patterns which do not, except perhaps a repeating pattern due to regular disturbances.

A variety of lines have been tested at different speeds and different levels for a period extending over $1\frac{1}{2}$ years. The main classes of lines tested are:

1. Switched local circuits in the London area.
2. Switched toll circuits in the U.K.
3. Private circuits to the Continent.

In general, the tests show an increase in the error rate when the speed is increased or when the sending level is reduced, as would be expected. They also show a distinctly different error pattern for private and switched connections. In a typical switched circuit the errors are fairly evenly distributed in time, whereas in a typical private circuit long error-free periods are interrupted by sudden bursts of errors.

3. The Organization of the Source Material

Many hundred error tapes have been produced in the above tests, most of which represent a run of two hours. Although the error tapes are suitable for direct input to the computer, they have the disadvantage that the errors are not designated by numbers, which makes interpretation lengthy. In addition, an error tape is awkward to read visually, and no printed record can be made from it. It is therefore worth while to use the computer to convert every error tape once and for all into a *derived error* (DE) tape on which serial numbers and errors are represented by numbers in telegraph code, with suitable symbols (e.g. — and +) to distinguish them. In this way the source material is put into a form which is concentrated, easily printed and interpreted, and which can be used over and over again as data for subsequent programs without requiring to call in lengthy conversion routines.

4. Basic Statistics obtained from the Source Material

Both the printed records and the DE tapes have been used in the manual and computer production of basic statistics, the principle of which will be described.

4.1. *Element error rate*

This is simply the number of bits in error compared with the total number of bits transmitted for a particular class of line under specified working conditions. This figure has only a limited significance since, in practice, transmission will occur in definite block sizes which will be treated by the error-detecting equipment in their entirety. It therefore becomes important to know the statistical nature of the errors within such a block.

4.2. *Block error rate*

This is the number of blocks with errors compared with the total number of blocks transmitted. This figure increases more or less linearly with block size in the range 50 to 500 bits per block. The choice of block size is influenced by the block error rate to a certain extent; obviously for the smaller block sizes the percentage of retransmissions of blocks found in error by the protection equipment will be less. On the other hand, in comparison with large blocks, the smaller blocks need a higher percentage of check bits to give an equivalent degree of protection, and this redundancy represents a loss of transmission time as far as information is concerned. Another consideration which favours the use of the larger block is that the transmission time required to get back "request for transmission" signals as generated by the error-detection device at the receiving station should preferably not exceed the transmission time of an information block in the forward direction. Finally, considerations of cost and engineering set an upper limit to the block size. It has been found that, for an equal degree of protection, the range of block sizes, for which the percentage loss of transmission time for intelligence is close to the optimal minimum, is quite wide and extends from about 50 to 300 bits per block.

4.3. *Distribution of number of errors per block*

The source material refers to 2,000-bit cycles which may be subdivided into particular block sizes appropriate to some proposal for data transmission and error-detecting equipment. The number of errors per block

can then be counted and a frequency distribution of their occurrence determined. This task can be done quickly on the computer, using the DE tapes as data. It is found that this distribution depends on the class of line. In the switched circuits the contribution made by the low numbers of errors per block (1, 2, up to 10) covers the greater part of the distribution. On a private circuit there is a more even spread, with greater contributions made by the higher numbers of errors per block, as shown in Fig. 3.

4.4. *Distribution of separations between errors*

The separation between errors may be measured in bits, provided that it is not intended to deviate from the transmission speed used in the tests when considering the artificial generation of errors and their effect upon a simulated protective scheme, which will be described later. The distribution is obtained by feeding the DE tapes into the computer with a program which finds the separations by subtracting every error number on the tape from the next error number. All separations from 1, 2, 3, up to 200 bits are listed and counted. Beyond this point separations are listed as "over 200." The first error in each 2,000-bit line-test cycle has always an unspecified separation. This is so even when the previous cycle contains an error, because there is a dead interval between these cycles during which the serial number is being made available. These unspecified initial separations may be subdivided into two groups: those over 200 and those under 200. The "over 200" group can be combined with separations over 200 within the test cycle. The "under 200" group is, however, quite unspecified and is left out of the distribution altogether. A typical error separation distribution curve is shown in Fig. 4, in cumulative form.

4.5. *Distribution of error separations relative to previous error separation*

As will be shown later, it is necessary to know the distribution of error separations relative to the error separations preceding them. Strictly speaking, this would involve obtaining such distribution curves relative to every possible value of previous separation, viz. 1, 2, 3, etc., elements. Clearly this is awkward and unnecessarily precise. It has been found sufficient to use only 8 Relative to Previous (RTP) error separation distributions, each of which is relative to a particular *group* of previous separations. These groups of previous separations have been chosen so that there is not too large a divergence between the error separation distributions relative to them. A typical set of RTP distribution curves is shown in Fig. 5, drawn in cumulative form.

5. The Generation of Simulated Statistics

5.1. *The Monte Carlo method*

This method is used to generate error separations artificially from the *cumulative* frequency distribution of error separation (C), which in effect is the integral of

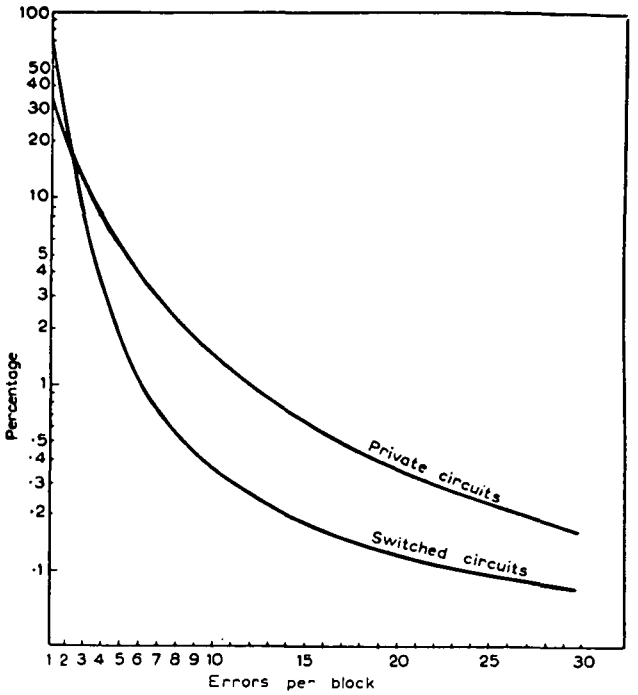

Fig. 3.—Percentage of faulty blocks having different numbers of errors. Block size = 50 bits

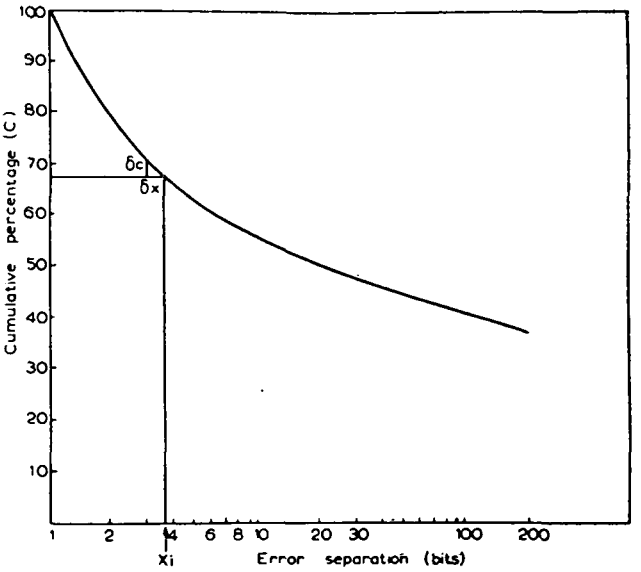

Fig. 4.—Distribution of error separations

the error separation distribution (see Fig. 4). A random-number generating process is used to produce values of C between the limits of probability 0 and 1. The frequency of getting x_i associated with a particular small interval δc will be given by:

$$\frac{\delta c}{1} = \frac{dc}{dx} \cdot \delta x$$

Now dc/dx represents the value of the frequency of x_i in the error separation distribution curve. Thus, by the

use of the cumulative distribution and a source of random numbers, it is possible to generate a set of error separations the frequency of which conforms to the experimentally obtained distribution, provided sufficient tries are made and the values of C are sufficiently random.

5.2. *Pseudo-random number generator*

This is a subroutine which generates random numbers by Lehmer's method. In this a new random number U_{n+1} is made by multiplying a previous number U_n by a suitable constant k, according to the formula $U_{n+1} = (k \,.\, U_n)$ modulo M, where $M = 2^{33} + 1$. The power of 2 used depends on the word length of the computer; here it has been chosen appropriate to Stantec Zebra, where the modulo value is obtained simply by subtracting the head from the tail of the product standing in the double-length accumulator as the result of multiplying by k. The action of k may be regarded as a "paddle" which stirs up the number; it also determines the cycle length after which the series becomes repetitive.

5.3. *The error generating process*

It is quite straightforward to produce a set of artificial errors by adding together the error separations, produced by the Monte Carlo process, to form a series of error numbers. The cumulative error separation distribution is stored in the computer such that the address of a location is a measure of C, while the content of a location contains the separation. Although this arrangement is wasteful of storage space, it is faster to look up a quantity than to compute it. Moreover, the range of separations is not very large, so that several values of separation can be packed into one word. The random-number generator then acts as an address selector, and the values of separation are found immediately.

Although the set of error numbers produced in this manner will exhibit the *overall* statistics of an actual channel, it will not be precise in detail because each separation is generated without reference to what happened before. In other words, the generating process has been assumed to be purely unrelated. In actual fact it is known that it also depends on previous separations. For simplicity it will be assumed to depend on only the immediately previous separation, and that contributions by separations further removed have no effect (a Markov process). One way in which this process may be regarded is that it applies a correction (either positive or negative) to the separation generated from the overall error separation cumulative distribution, as shown in Fig. 5. This correction is in fact a kind of correlation function between present and previous separations.

The generation of separations according to the above process by a computer program is done as follows. The eight RTP distributions described in Section 4.5 are put into cumulative form and are packed into a storage section of the machine in a manner similar to that used

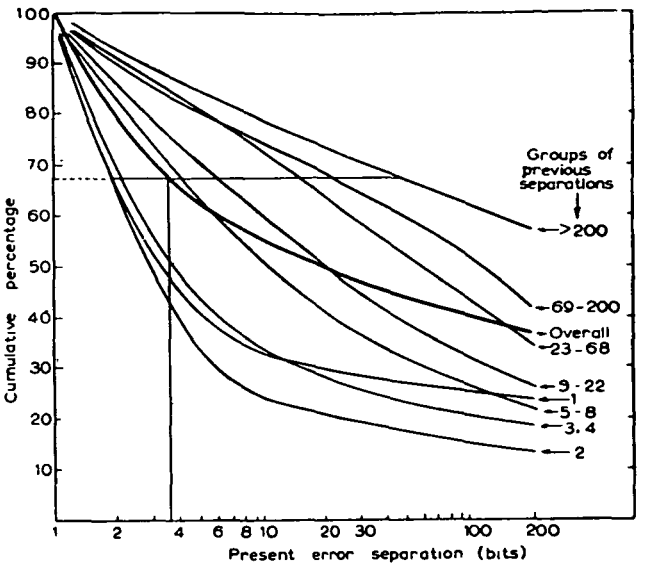

Fig. 5.—Distribution of error separations relative to groups of immediately previous separations

for the overall distribution. A random number is generated and, with a knowledge of the group in which the previous separation is situated, the appropriate separation is looked up. Separations produced in this way are found to correspond much more closely to those found in reality, and may be regarded as a sufficiently close approach to make them a reliable source of artificially generated errors.

5.4. *Arranging the errors into block sizes*

A set of error position numbers is produced by adding each separation to the previous error position number. As soon as an error position number exceeds a specified block size, it is reduced to modulo block-size value. In this way sets of error numbers consistent with block size are made, without upsetting their actual positions of occurrence. If a separation of "over 200" is looked up, there is uncertainty about the starting position in the next block. It is easiest to make the new starting position the same as the position reached before the "over 200" separation occurred. It is probably better to determine the new position by throwing a "die" having as many sides as there are bits in the block. This is quite easy to do with a random number generator at the sacrifice of some speed.

It is arranged that the error generating program will make available only those numbers of errors per block that are of interest to the particular error-detecting scheme which is being simulated concurrently in the computer. This means that if blocks with, say, up to ten errors are called for, such blocks may only be released until the process has been taken into the next block, in order to guard against the possibility of there being errors beyond a total of ten. A facility for releasing blocks with only an even number of errors is provided for certain types of simulation.

6. The Simulation of the System

A proposal for an error-detecting scheme can generally be simulated without difficulty on a digital computer. In general, an actual protective system consists of an arrangement of logical interconnections between sending and storage devices. The functions performed by it are very similar to those available in the majority of computers. The simulation process will therefore have to make judicious use of the computer store, its index registers, and its arithmetical functions such as addition, shifting, and collation. Some of the programs simulating proposals for error-detection schemes will be described. They are arranged to operate either on real errors (from the DE tapes) or on artificially generated errors.

6.1. *Group parity error detection*

In this the block is split into a number of groups of, say, five bits each. Parity bits are associated with the 1st, 2nd, 3rd, etc., bits of each group. All odd numbers of errors per block will be detectable. The protection given by this scheme can be very powerful if several systems of such group parities are used together, e.g. three systems in which the group lengths are 3, 5, and 7 bits. A simple arithmetical method may be used to find if a pattern of errors is detectable or not. Each error number is expressed in its modulo value appropriate to the group size, e.g. 3. Whenever a particular modulo value is encountered, i.e. 0, 1, 2, a count is made in a location associated with it. After all errors in the block have been dealt with, these counts are inspected and if any one contains an odd number, the block is detectable. Instead of counting and inspecting, use can be made of addition without carry, which will be described in the next Section.

6.2. *Binary parity error detection*

Here the elements of the block are labelled with binary numbers, having, for example, three bits. The labelling scheme is then as follows:

ELEMENT NO.	BINARY LABEL
1	001
2	010
3	011
4	100
5	101
6	110
7	111
8	001
9	010
10	011
11	100
12	101
13	110
14	111
15	001
etc.	etc.

In an actual system, whenever a one occurs in the intelligence, the binary label will be added, according to the rules of addition without carry, into a parity register which is at zero at the start of the block. This arrangement is more powerful in dealing with even numbers of errors in close proximity, but it will not always detect odd numbers of errors as is the case with group parity.

A mutilated block may be regarded as intelligence plus errors. The error pattern on its own will tend to drive the parity register from its initial zero state. Those patterns which do not move it from zero are undetectable. In the simulation program every error number is first converted to modulo form (this would be 7 in the example given). If the modulo value is zero it is converted into the radix, e.g. 7. Each modulo value is then added without carry into a parity register location, which is tested for zero at the end of the block.

Addition of two numbers without carry consists of adding corresponding digits and disregarding carries, e.g.

$$\begin{array}{r} 0011 \\ 1010 \\ \hline 1001 \end{array}$$

It may be programmed according to the rule:

Sum without carry = arithmetic sum − left-shifted collation.

The operations on the right-hand side can be performed directly by the arithmetical hardware in few instructions in a machine like Zebra.

6.3. *Shift register parity*

This has a labelling scheme similar to Binary Parity, but the labels do not follow a regular sequence of increments of one. This form of parity is sometimes known as "scrambled" binary. Odd numbers of errors per block may all be made detectable for certain configurations of this system. But the system's main power lies in its scrambled nature, which is particularly suitable for detecting short sequences of errors which are so characteristic of actual line conditions. In an actual equipment the parity bits are generated in a shift register, into which the elements of the intelligence are added without carry at a number of points as shown in Fig. 6. As in the binary parity scheme, the errors considered alone will drive the parity register from zero unless they are undetectable. The parity register is represented in the simulation program by the accumulator and its shifting facilities. The input points are specified by a mask, and the addition without carry is done by the method already described.

The cycle of repetition of the register on its own has an important bearing upon the effectiveness of the system. A number of subsidiary programs has been written to determine the cycle length and the cycle structure when the configuration of input points is varied.

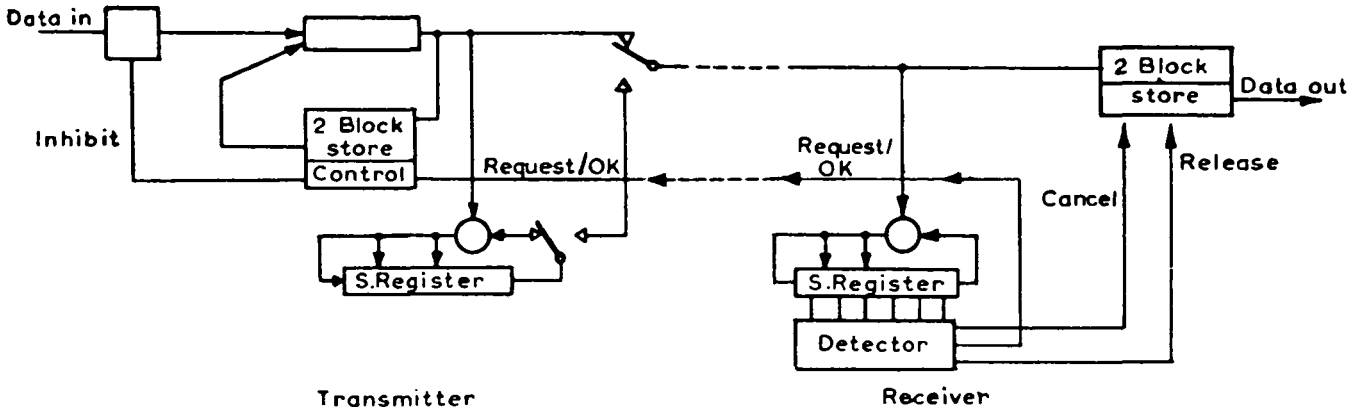

Fig. 6.—Shift register parity error detection system

7. Evaluating Overall Improvement

7.1. *Error reduction factor*

The effect of the protective scheme will be to reduce or eliminate the acceptance of blocks having a particular number of errors. The error reduction factor is defined for a particular number of errors per block as:

Number of undetectable blocks/Number of faulty blocks.

The simulation process gives these factors directly, provided a sufficiently large sample is taken.

7.2. *Overall improvement factor*

This is a measure of the overall effectiveness of the error-detecting measures and is given by:

Number of faulty blocks/Undetectable blocks

considering all possible numbers of errors per block. To obtain this factor the distribution of number of errors per block must be known (see Section 4.3 and Fig. 3). The percentage incidence of a particular number of errors per block is multiplied by its error reduction factor, which may of course be zero. The sum of these percentages taken over all possible numbers of errors per block gives the percentage of undetectable blocks out of all blocks in error. The overall improvement factor is obtained by dividing this percentage into 100. Obviously this factor will depend on the type of line because of the variation in the distribution curve as shown in Fig. 3. Some typical values of overall improvement factors for different protective systems are shown below:

Group Parity	
3 bits	10
5 bits	23
7 bits	56
5 and 7 bits	670
3, 5, and 7 bits	4,900
Binary Parity	
3 bits	7
5 bits	25
7 bits	50
5 and 7 bits same starting point	250
3, 5, and 7 bits same starting point	700
Shift Register Parity	
12 bits	40,000

8. Conclusions

The use of a computer with a Monte Carlo method to generate artificial statistics together with a simulation of a system, is a powerful way of assessing the effectiveness of the system without having to go to the trouble of building it. Once a sufficiently large amount of real statistics has been collected the computer can quickly produce sets of performance figures for a variety of conditions and design parameters.

Before this study of error statistics of data transmission was undertaken, there was considerable uncertainty about the relative merit and degree of confidence that could be given to various protective schemes. In particular, it was not known to what extent the redundancy (parity bits) per block could be reduced without an appreciable loss of security. It is felt that simulation studies can be relied upon to give an accurate prediction of what may be obtained with a practical system under typical working conditions.

9. Acknowledgements

Acknowledgements are due to the management of Standard Telecommunication Laboratories Limited, Harlow, for permission to publish this paper, and to Mr. E. P. G. Wright who initiated this work and contributed many ideas. The author would like to thank Mr. A. D. Marr for his help with the statistics and Messrs. D. G. N. Hunter and M. Lawn for their advice and help with the programming of Stantec Zebra.

10. References

WRIGHT, E. P. G. (1961). "Data Collection and Transmission," *The Computer Journal*, Vol. 4, p. 103.
PETERSON, W. W., and BROWN, D. T. (1961). "Cyclic Codes for Error Detection," *Proc. I.R.E.*, Vol. 49, No. 1, p. 228.