Link systems for multi-computer control of a large process: Part 2—Fault detection

P. J. Comerford

Computing Laboratory, University of Bradford, Bradford BD7 1DP

High integrity in a hostile environment is an essential feature of a multi-computer process control complex. Part 2 of this paper considers fault conditions associated with the inter-computer link system of such a complex and describes methods of detecting these conditions. The methods described are embodied in the scheme for inter-computer link construction defined in Part 1 of this paper. Assessments of their effectiveness are based upon experience with a working inter-computer link system constructed under the above mentioned scheme at the University of Bradford.

It is concluded that reliable detection of a wide range of fault conditions in a hostile environment is possible and that this may be achieved with little sacrifice of speed of normal working of the link system.

(Received January 1973)

1. Fault conditions

A fault condition associated with an inter-computer link system may arise from one or more of the following causes:

1. A transmission error occurring during a data transfer between two system devices
2. Transfer blockage due to a device fault or due to a transmission fault
3. Other fault conditions arising from device hardware or software failures, including data block size disagreements
4. A link system hardware failure.

2. Fault detection mechanisms

This section describes the mechanisms by which the fault conditions listed in Section 1 are detected in link systems constructed under the author's scheme.

2.1. *Transmission errors*

Transmission error checks are made only on information passing over a serial link. Two classes of error are possible. They are:

1. Inversion of one or more data or control code digits during their passage over the link
2. Spurious '1' digits generated when no data or control code is being sent. ('0' digits are normally transmitted at this time and the appearance of a '1' signifies the start of a new code or data word transmission.)

To detect errors of the first class a feedback checking system has been adopted. All data and control code bits received over either channel of the fully duplex link are immediately returned over the other channel for checking. Transmissions are, therefore, effectively in half duplex mode.

Feedback checking is very reliable. The only transmission error condition that can deceive the checking circuits occurs when the erroneous inversion of a digit on its forward path is corrected by another erroneous inversion on its return path. In such an event two other lines of defence are available. The first uses waveform violation detectors on both link channels to signal departures of incoming line waveforms from a legal pattern. These detectors can sense most erroneous inversions and it is unlikely that compensating errors will escape them at both ends of a link. Violation detectors used in the Bradford system were a manufacturer's optional extra. Where such an extra is not available, the necessary circuit can easily be constructed from standard logical elements.

The second line of defence makes use of the fact that transmission errors due to electrical interference tend to occur in bursts. In the unlikely event of compensating errors escaping the violation detectors, it is probable that an adjacent error will raise an alarm. This will cause invalidation and resending in full of the transfer involved.

The safeguards described above provide high security against undetected inversion of a received control code or data word digit. A conservative calculation puts the probability of missing such an error at 1 in 2^{50}, provided, of course, checking hardware is functioning correctly.

As well as offering high security, feedback checking does not require the transmission of redundant check bits. Thus, under the author's scheme, data words can be transmitted as a consecutive block, adjacent words being separated by only two identifier digits. Individual confirmatory messages (confirming to the receiving end that no errors were detected at the sending end) are transmitted only for control codes.

The absence of individual confirmatory messages associated with data words permits efficient sending of data, but allows data words in error to be input to the receiving device. There is no security risk here provided no use is made of received data until terminating status for the transfer is raised. A transfer in error has to be resent, but, as noted earlier, this aids the reliability of transmission error checking.

The above arrangement is satisfactory where the transmission error rate is not high. Where a high transmission error rate is experienced, it will be necessary to arrange for some method of resending individual data words in error and this will involve the use of a confirmatory message for each data word sent. Such a message may take the form of a confirmatory character or may be the identifier of a later data word (for final data words, the terminating control code). The latter form of confirmatory message permits a data transmission efficiency comparable with that possible under the author's scheme, as well as providing faster recovery from a transmission error condition. It does this at the cost of additional data buffering and control circuitry and some loss of security.

In addition to its use for checking purposes, the return path of a data word transmission is also used to carry 'wait' messages. A 'wait' message is sent by simply inverting the returning digits of any data word which finds the receiving data buffer busy. The word is retransmitted automatically.

The feedback checking system does not offer protection against transmission errors of the second class, and, unless other precautions are taken, the appearance of one or more spurious '1' digits when no data or control code transmission is in progress may give rise to a phantom transmission or cause misinterpretation of the type or timing of a legitimate transmission closely following the spurious digits.

To avoid both these fault conditions, identifiers for control

codes and separated data words are longer than the minimum of two bits (start bit plus transmission type bit). The identifiers are checked for legality on receipt and on return by special circuits and detection of an illegal identifier or the appearance of a line waveform violation alarm accompanying an identifier digit will cause automatic temporary suspension of new transmissions from the end or ends of the link affected. This gives time for recovery from any out of phase condition created between the two ends of the link as a result of the fault. When data words follow each other immediately, sufficient security can be provided by a two bit identifier, with consequent benefit to the efficiency of data transmission.

Thus, the transmission error checking circuits offer high security against both classes of transmission error, with little sacrifice of data rate under fault-free conditions.

If the time taken by the transmission of control codes is neglected, a ratio of transfer time spent transmitting useful data bits to overall transfer time of 92·3 per cent is possible (see also Part 1, Section 5).

Admittedly the scheme requires fully duplex links to provide a half-duplex transmission facility but, within the environs of an industrial plant, the cost of such links is moderate and partially offset by the low cost of the simple checking hardware of a feedback system. In addition, it can be argued that the advantages in this situation of a fully duplex transmission facility are few and outweighed by the organisational difficulties of such an arrangement.

More conventional error detecting code schemes can use half-duplex links but may introduce a high level of redundancy. Individual checking of 24 bit words using cyclic error detecting codes would need about seven check bits per word for moderate security (99.2 per cent probability of detection of a random error pattern), reducing the best ratio of data transfer time to overall transfer time to 72·7 per cent. If data can be organised into long words better redundancy ratios are possible, e.g. 50 check bits for 1,000 information bits, but generating and checking hardware requirements are considerable.

2.2. *Transfer blockage*

A data transfer is initiated by the data sending device. The data receiving device must accept or reject the transfer and failure to do so will result in a transfer blockage. While a transfer is in progress, failure of either device to supply or receive data will have the same effect. A third cause of transfer blockage is the persistent failure of control code or data transmissions.

Solidly blocked transfers are terminated by a forced release mechanism controlled by timers fitted to all system device adaptors. Whenever a transfer exceeds a pre-determined maximum duration an exchange of fault status terminating control codes is forced. If the codes cannot be exchanged successfully the transfer is abandoned.

Abandoned transfers may be re-attempted as many times as required, or transfers over other paths, possibly using link system modules previously involved in the abandoned transfer, may be attempted. The forced release mechanism ensures at all times that healthy modules are freed for such attempts.

2.3. *Other fault conditions due to device failures*

To protect its store against unwanted incoming transfers erroneously or otherwise initiated by other devices, every computer attached to a link system is given a set of 'accept' commands. Data cannot enter the store of a computer unless the 'accept' command relating to the sending device has been issued by the computer.

Store access requests are inhibited when the number of data words specified by a computer has entered its store. The arrival of further data words will raise a data block size disagreement alarm. The arrival of fewer data words than expected will raise the same alarm.

Transfer initiating commands issued by a device will not be obeyed while an earlier transfer involving the device is in progress. Before such commands can be obeyed the earlier transfer must have terminated and its terminating status have been read by the device (See Part 1, Section 3).

2.4. *A link system hardware failure*

Circuits are installed to signal excessive variations of link system logic power supplies. No other special precautions are taken with respect to link system hardware failures.

3. Recovery from a fault condition

The link system recovers automatically after the clearance of any of the faults described in sub-sections 2.1 to 2.3.

4. Communication of fault alarms between devices

At the termination of a data transfer between two devices it is important to ensure that:

1. A fault that could affect the accuracy of received data is signalled at both sending and receiving devices.
2. A disagreement over the amount of data involved in the transfer is signalled at both devices.
3. A fault that does not affect the accuracy of received data or its amount is signalled at both devices or at neither.

The first and second of these requirements are essential to the integrity of the system. The third ensures that ambiguous transfer terminating conditions are avoided.

In many cases, a fault alarm appearing during a transfer between two devices is raised at the device adaptor of one device only. Under normal circumstances, the alarm is communicated to the other adaptor by the terminating control codes of the transfer and there is no difficulty in meeting the fault communicating requirements listed above.

It is more difficult to meet these requirements when the terminating control codes themselves are affected by transmission errors which cause their persistent failure. A potentially ambiguous condition occurs when a successfully received terminating code transmission appears in error at its sending device adaptor due to a transmission error affecting the return of the confirmatory message for checking.

Special precautions have been taken to deal with this and other difficult conditions and the fault communicating requirements listed above are met under all the fault conditions defined in Section 1, with the exception of certain link system hardware failures.

Further details of all fault detection and communication facilities are given in Comerford (1971).

5. Performance

Extensive tests of the Bradford link system under real and simulated fault conditions have shown it capable of dealing adequately with transmission errors and device failures of every expected type.

It proved impossible to generate an undetected transmission error during tests. In use, the transmission error rate has been low but no data corruption or system malfunction due to undetected errors has been observed. The transfer blockage and data block size disagreement detection facilities have been put to good use during program development and have performed reliably. No fault signalling ambiguities between sending and receiving devices have been observed.

The principal shortcoming of the fault detection facilities lies in their insensitivity to some fault conditions in link system hardware. This disadvantage is shared with many computers and their peripheral interfaces.

6. Conclusions

The results achieved show that it is possible to design a link system to reliably detect a wide range of fault conditions in a hostile environment, and to do this with little sacrifice of speed of normal working.

Future work should be aimed at improving the self-detection facilities of system hardware, particularly where failures could affect the accuracy of transmitted data.

References

COMERFORD, P. J. (1971). *A High Speed, Self Checking, Inter-Computer Link System*, University of Bradford, Ph.D. Thesis, December 1971.

AUSTIN, J. S., BARNES, R. C. M., and FERGUS, P. J. B. (1967). *An 840 Kilobit/sec Data Transmission System for Computer-to-Computer Communication*, HM Stationery Office, (AERE-R 5529).

ROSSER, W. J. (1970). High Integrity Data Transmission System, *Design Electronics*, p. 50.

Book reviews

Le Langage ALGOL W: Initiation aux Algorithmes, by J. S. Chion et E. F. Cleemann, 1973; 292 pages. (*Presses Universitaires de Grenoble*, 30F)

Of the many dialects of ALGOL 60 in use at present, ALGOL W has a considerable following partly because of the efficiency of its compiler, partly because of the useful set of debugging facilities provided but also because of the range of types of data structure which can be represented in the language. The availability of string handling operations, records and references has allowed the authors considerable flexibility in choosing exercises and examples. As a result this book goes considerably past the stage commonly reached in most introductory textbooks. Such topics as converting expressions to postfix notation, evaluating postfix expressions and the use of recursion in the description of trees are included in the text. Anyone working through this book would thus not only have a working knowledge of ALGOL, but would also acquire an understanding of some of the data structures which are essential in the theory of programming languages. The first chapter of the book, however, has a pragmatic flavour in describing the nature of algorithms, and this is followed by a detailed description of the language, with careful explanation of each feature. There is a useful set of problems at the end of the book, which illustrates many of the features of the language described in earlier chapters.

In summary, this book is strongly recommended as background reading for Computer Science undergraduates, and for inclusion in departmental libraries.

M. H. ROGERS (Bristol)

Theory of Linear and Non-Linear Programming, by S. Vajda, 1974; 118 pages. (*Longman*, £2·95)

This brief book (110 pages of text) may be briefly reviewed. Like everything Professor Vajda writes it is excellent of its kind. It is a book about mathematical programming written for mathematicians. It contains no account of applications and so is not a work for anyone intent on learning how to formulate problems in terms of programming and then get an answer.

Professor Vajda adopts a definition of mathematical programming which means minimising or maximising a function of several variables subject to inequality constraints. He excludes integer, geometric, dynamic and stochastic programming from consideration.

He begins with a succinct account of convex sets, then develops the theories of alternatives and of duality, discussing, inter alia, von Neumann's minimax theorem. There follows a chapter on linear and quadratic programming. Evidence, if needed, that this is a mathematician's book is provided in two throw-away footnotes 'this is the rationale underlying the Simplex method' and 'this is the basis of the primal-dual algorithm'.

For non-linear programming he has a chapter on theorems of sufficiency and theorems of necessity in which he carefully and with considerable skill selects the essential mathematics from the Kuhn-Tucker theory and its later developments. He concludes with a short chapter on duality in relation to non-linear programming.

The book is written with extreme economy of style—I doubt if there is a superfluous word. The amount of knowledge condensed into it is quite remarkable. I highly recommend it.

A. YOUNG (Coleraine)

Computer-Aided Information Systems Analysis and Design, edited by J. Bubenko Jr., B. Langefors, and A. Sølvberg, 1972; 207 pages. (*Lund: Studentlitteratur; Copenhagen: Akademisk Forlag; London: Auerbach*, £4·00)

Although this is a belated review of an English translation of the proceedings of a conference which took place in April 1971, the subject matter is still of interest.

The conference was of the Scandinavian Information Processing Projects (SCIP) which is made up of a collection of co-operating research groups working in the area of information systems design. The term 'computer aided design' is better known in the Engineering field but is applied in this case to the automation of the design of Information Systems. Several, but not a majority, of the papers in the book place emphasis on computer aids to design. The point that is being made is that an information system (database) can be used to control the process of designing information systems (databases). Langefors goes as far as defining a new word 'cad' for a 'system for computer aided design'. However it is the formalising of the multi-stage design process and the relational approach adopted by the project being referred to (CADIS) which is of greater interest than the automation. Other authors are more down to earth and present their systems in terms such as 'Computer Based Documentation System'.

The conference gave a good coverage to the principles, tools and even the politics involved in the design of information systems, with case studies included to balance the theoretical papers. If it were not for the time that has elapsed since the conference this book would be recommended reading for those considering the problems of database design and the development of data dictionary systems. However it must be expected that the projects involved in SCIP have continued and that later papers have been published.

J. S. KNOWLES (Aberdeen)

Short notice

A comprehensive catalogue of Software available in 1974 *for exchange among members of the* (*American*) *Joint User Group*, 1975; 806 pages (*Collier Macmillan*, £12·50)

Programs are listed by manufacturer user group and there is a comprehensive subject/category index of 5,180 items. The volume is available in the BCS Library.