ICL Technical Journal

Volume 3 Issue 3 May 1983

ICL Technical Journal

Contents

Volume 3 Issue 3

ICL Technical Journal

The ICL Technical Journal is published twice a year by Peter Peregrinus Limited on behalf of International Computers Limited

All correspondence and papers to be considered for publication should be addressed to the Editor

1983 subscription rates: annual subscription £13.00 UK, £15.50 ($35.00) overseas, airmail supplement £6.00 ($13.00), single copy£7.50 ($16.00). Cheques should be made out to 'Peter Peregrinus Ltd.', and sent to Peter Peregrinus Ltd., Station House, Nightingale Road, Hitchin, Herts. SG5 1SA, England, Telephone: Hitchin 53331 (s.t.d. 0462 53331).

Publisher
Peter Peregrinus Limited
PO Box 8, Southgate House, Stevenage, Herts SG1 1HQ, England

Printed by A.McLay & Co. Ltd., London and Cardiff ISSN 0142–1557

Editorial

Communications

This issue of the *ICL Technical Journal* – which happens to be the tenth since publication started in November 1978 – departs in style from previous editorial policy. In all issues to date the Editor has selected papers covering a range of subjects. In this issue all the papers, with one exception, are concerned with a single subject – Communications.

To all who are associated with computers, whether in manufacturing, in software houses, as users or in any other way, 'Communications' is associated with technicalities such as modems, networks, terminals, communications architecture, teleprocessing control programs and so on. These terms are themselves the keywords in a wider range of technical issues, all of which can be developed into technical discussions and disputations. The importance of the subject is evident from the variety of views that exist and the vigour with which these are promoted by their advocates.

The ICL position derives from some simple concepts that are not technical but rather are concerned with the practical needs of our customers. The physical separation of those who create or enter information into a system, or use information in a system, either from the system itself or from the management structure of their businesses, generates a need for communications capabilities. Distributed processing is not the isolation of information processing but its integration. Decision making depends not only on the processing of information, wherever that may be done, but also on the access to information that may be located elsewhere and which may have been collected from many points in the organisation. Thus there is an intrinsic need for a range of technical communications facilities, so that computer systems can match the physical dispersal of the organisations they serve.

Our conviction of the importance of this is so strong that we believe that the underlying structure of our systems must be based on communications capability. Hence the concept of the Networked Product Line, deriving from the belief that, by reflecting communications concepts in the products we develop, the products will better match the needs of the organisations they must serve.

Further, we recognise that our customers' needs will change and that ICL cannot provide every component of hardware and software needed for the solution of their problems. The only way that we can accommodate both change and range is to base our products on vendor-independent international standards, particularly those which relate to communications issues. This gives our customers security that they can build systems of technical longevity because future developments of new products will fit the architectural concepts. Thus these new components can be absorbed without destroying the investments in intellectual effort made in the design, implementation and operation of their systems.

We believe our approach is logical because it responds to customer needs as evident in business operations today, and allows for the inevitable changes that will occur both in those needs and in the technical facilities that can be offered as ways of meeting them. There is an additional factor, however, also concerned with 'Communications', that is vital to the success of the technical approach.

Communication is an act of imparting knowledge. We need to make visible the technical directions we are following so that our customers can make use of these directions in their own plans, both short term and long term, not just for computer systems but for the general development of their organisations. One of the factors in an effective organisation is the efficiency with which information moves, whether instructions on operations that must flow down the organisation or results from operations that must flow up and across. Our products are related to these tasks, but we must ensure that our customers appreciate the range of business possibilities and the contribution that these products can make to the adoption of effective styles of organisation and operation. Likewise we need to understand the needs of our customers in these terms, so that we can keep our product strategies aligned to real needs.

These are communication tasks that are just as important as the technical communication capabilities of the products. This Journal is one of the communication channels that allows for the exposure of the technical basis for some of our thinking and for the exploration of innovative applications of products and techniques, possibly going beyond the capabilities that we might support throughout the world.

ICL's commitment to 'Communications' is twofold. The concepts are embedded in our product plans, to the extent that these plans assume a communications-oriented system as being the basic need of our customers. They are embedded also in our operations, in that we want our customers to know our thinking and intentions so that they can exploit our products. And we want the communication flow to be in the other direction also, so that we know our customers' general intentions and can keep our product line alive to their consequent needs.

J.M. Watson
Technical Director, ICL Putney, London

Note on the papers

Of the seven papers in this issue on Communications, four form a related group dealing with ICL's information-processing architecture, IPA. The remaining three are separate from this group and from each other.

As a unifying concept a communications architecture is of the greatest importance both to the manufacturer and to the user; as Kemp and Reynolds said in their paper in the November 1980 issue of this Journal (Vol.2 no.2), IPA 'embodies ICL's strategy for information processing over at least the next 10-12 years'. The essential concept is simple; to quote Kemp and Reynolds again, 'in broadest terms its objective is the provision of standard methods for linking together computer systems and terminals via either public or private telecommunication lines or networks, extending to international linking'. But the practical realisation on the scale necessary if there are to be true benefits to the user is far from simple, largely because where communication between computers and similar devices is concerned so much has to be foreseen and planned for in the greatest detail – especially for the handling of faults and failures. The authors of the IPA papers wrote these in collaboration and, as I said, as a related group; there is a certain amount of intended overlap, particularly in the references, so that each paper is to some extent self-contained. We – meaning the Editor and the Editorial Board

– hope that these papers will convey the basic structure of IPA and enough of the details to give a reasonable picture of how it is being implemented.

The paper by Hughes of British Telecom is an overview of the types of telecommunication channel provided by the responsible national body: the vital importance of this provision to the whole field of information communication hardly needs stressing. Mapstone's paper on the formal specification of a communications protocol is included specifically to show how the properties of a communications system can be expressed and manipulated in a formal language, with the benefit of bringing mathematical and logical disciplines to bear on the processes of design and validation. Finally, Stevens's paper on ICL's MACROLAN networks describes the exploitation of state-of-the-art ideas and hardware in attacking the problems of internal communications within a computer system.

J. Howlett
Editor

IPA networking architecture

J.B. Brenner
ICL Network Systems Division, Kidsgrove, Staffordshire

Abstract

This paper is a general overview of ICL networking architecture. The nature and importance of networking and the ICL Networked Product Line are explained. The general requirements and strategic intent of the Networked Product Line's networking architecture, IPA, are described, with particular emphasis on open systems interconnection, OSI. The modular structure of the architecture is explained. The content of IPA is then summarised.

1 Introduction

The future business strategy of ICL is built around the concept of a Networked Product Line as the means of providing networked information systems. The main subject of this paper is the networking architecture for this product strategy.

1.1 Networking

We start by considering the nature of networking, and why networking is so important.

A networked information system is one which provides information services by interworking among physically distinct and interconnected units. The dispersion of its components may be local or widespread; their number may be small or large.

The need for networking is inherent in most information systems, whether computerised or not, because they generally involve people in separate places, information in separate places, and often separate organisations. The more sophisticated and information-rich a system is, the more it tends to involve interworking and integration between different producers, holders, processors and consumers of information. There is a fundamental symbiotic relationship between information and communication: information is nothing unless it is communicated, and the only substance of communication is information.

There is already a strong trend towards increased networking of computer systems. Technology is pushing in this direction; e.g. VLSI, digital telephony, PABX, local area networks. New public networking services are becoming available; eg. X25 packet switching, X21 circuit switching, teletex, videotex. The strong emergence of all kinds of common standards in the public domain now makes computer systems networking more readily achievable. The technological changes are strongly associated with the convergence by several industries (the electronics, data-processing, telecommunications, office products and consumer products industries) onto the same intensely competitive new market.

Networking is also a consequence of the growing use of Information Technology. The more computers there are, and the more people use them, and for more purposes, the more it will be that networking becomes an integral characteristic of information systems. The advent of the one-per-person computer (be it a hobby computer, or a professional work station) now opens up the opportunity and the need to communicate with other computer systems on an unprecedented scale. It is also the focal point for the integration of voice-data-image communication (its starting point is telephone + computer + TV). This new man-machine combination will be more inquisitive, communicative, ingenious, resourceful and prolific than anything that has gone before. The networking revolution is just beginning.

1.2 Networked Product Line

The Networked Product Line is a distinctive positioning of ICL products in the rapidly developing market for networked information systems.

The basic concept is that the potential value and volume of any Information Technology product is greater if it is able to network with others. Its functionality is augmented by that of the other products with which it can interconnect and interwork. It can be engineered to achieve its own particular purpose in an optimum way, as can the other products with their own particular specialities. This and the associated economies of scale provide a basis for much sought after '1+1=3' synergy; the added value is in the integration of the whole.

The Networked Product Line is also envisaged as a basis for ICL collaborations with specialist suppliers. The full spectrum of leading edge Information Technology capability cannot be provided from the resources of a single company. Nor can any one company achieve best cost and quality for every kind of Information Technology product. The subdivision of functionality which is inherent in the Networked Product Line greatly facilitates the integration of different products from different suppliers. This introduces the subject of networking standards, of which more later.

Fig. 1 illustrates Networked Product Line networking, and completes this summary of the heterogeneous networking environment in which ICL networking architecture is required to operate.

2 IPA

The ICL Information Processing Architecture, IPA, is the networking architecture of the ICL Networked Product Line. The scope of IPA includes all of the networking characteristics of the Networked Product Line, but IPA at this stage of its evolution is primarily concerned with communications. The current focus is *data communication* which includes data-encoded *text communication.* There is increasing integration with *voice communication* and *image communication*. This reflects the

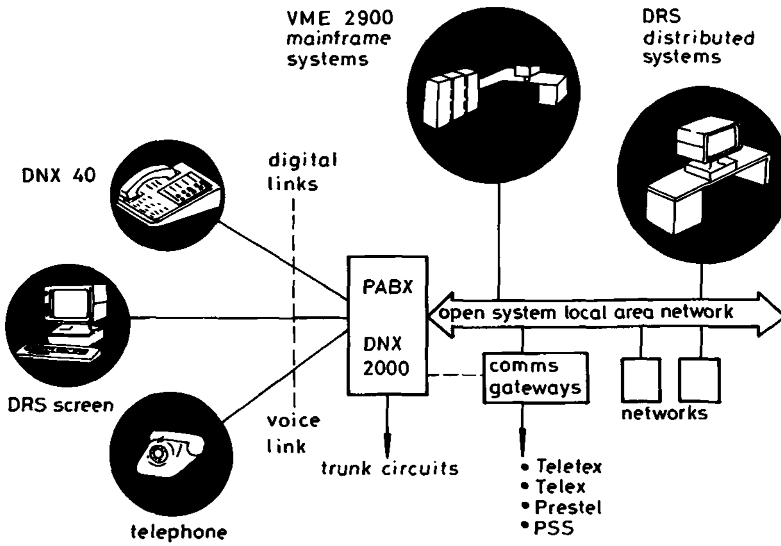


Fig. 1 An example from the Networked Product Line

technological and industrial convergence of these different types of communication, and the market needs for integrated voice-data-image communication, particularly for office systems.

2.1 Openness and Heterogeneity

A distinguishing characteristic of IPA is its commitment to use international public standards to achieve open networking.[1,2] The architecture is built around the ISO open systems interconnection reference model,[3] and uses protocol standards conforming with it. The aim is to use protocols which are internationally standardised and in the public domain, and only to use proprietary protocols where public standards are not available. However, most of the Information Technology market is currently based on proprietary standards, and it is therefore necessary for IPA to use proprietary standards in addition to public standards.

2.2 Evolution and Modularity

The strength of the IPA networking architecture is its ability to evolve rapidly across the virtually limitless field of networked information systems, taking advantages of new techniques, new standards and new market opportunities. All the while it must carry forward the existing capabilities and the user investment in them. The ability to change sufficiently rapidly and at acceptable cost and without undue disruption must be combined with the ability to maintain compatibility across a large and heterogeneous spread of protocols and products. This is reflected in the highly modular and flexible structure of IPA. The architectural modularity of IPA is one of the main themes of this paper.

The specification and content of IPA have evolved through a succession of evolutionary phases. The architecture described here is generally referred to as 'IPA phase 3'.

Table 1 summarises the phases to date.

Table 1 **IPA phases**

Phase 1	1970s. Consolidation of basic mode protocols into full extended basic mode protocols (FXBM)
Phase 2	Early 1980s. Phase 1 plus: preliminary layering of the architecture access to X25 networks file transfer facility (FTF) remote session access (RSA) distributed message router (DMR) distributed application facility (DAF) application data interchange (ADI) range remote job entry (RRJE) access to SNA networks.
Phase 3	Mid 1980s. Phase 2 plus: additions to the phase 2 facilities modular layered architecture fully in place ECMA, ISO, and CCITT standards in use access to value added networks (VANs) connection to LANs and PABX preliminary voice-data-image integration new terminal types

Current ICL products use phase 2 of IPA (see References 4 and 5) which provides the basic facilities for comprehensive IPA networking (FTF, RSA, DMR, DAF, ADI, RRJE). These facilities have been built over the existing ICL full extended basic mode (FXBM) protocols and X25. Products which are now beginning to reach the market place use phase 3 IPA, which extends the existing facilities to work over a fully layered and modular architecture, using public standards.

3 Strategy

The strategic intent of IPA is now considered further, in order to provide the background against which the architecture can be described. These strategy considerations explain much of the 'why' of IPA technical content.

3.1 Goals

The principal strategic goals of IPA are as follows:

(i) *Continuity:* Carry forward and enrich the existing IPA capabilities and the user investment in them.

(ii) *OSI:* Implement open systems interconnection standards as the native mode of IPA operation as soon as possible.

(iii) *VANs:* Support access to the public Value Added Networks provided by PTTs and others (e.g. teletex, videotex and electronic mail); and provide equivalent private facilities (e.g. private viewdata facilities, private electronic mail).

(iv) *SNA:* Support access to networks using the IBM architecture SNA.

(v) *Usability:* Provide the architectural structure for excellent networking quality (i.e. reliability, resilience, maintainability, usability, response time, throughput, efficiency).

(vi) *Technology:* Provide the architectural structure for ICL products to exploit new telecommunications services and technologies (e.g. fast circuit switching networks, high speed transmission systems, satellite communications, metropolitan area networks, cellular radio, ISDN etc).

(vii) *Cost:* Define the architectural basis for highly cost effective products, particularly by moving implementation of the architecture into silicon.

3.2 Intercept Strategy

To achieve the kind of technological and standards and market leadership envisaged for IPA, aggressive exploitation of new standards and new technology is being pursued. This is managed as a set of 'intercept strategies'.

The basis for an intercept strategy is the perception of a likely future business opportunity which can best be seized by starting lead-in activities well before the opportunity actually occurs. By successfully anticipating the outcome, one intercepts it. An intercept strategy is obviously risky, because of the uncertainties of the predictions on which it is based (e.g. the predicted opportunity may not occur). On the other hand, in a highly competitive and rapidly changing field of business, the alternative approach of wait-until-everything-is-certain is almost bound to fail by being too late.

A successfully executed intercept strategy is achieved by the following means:

- good predictions
- continual updating of predictions, followed by suitable course corrections
- careful management of the commitment/risk profile during progress towards the intercept
- gaining some degree of control over the train of events which are being intercepted.

This is analogous to picking winners, backing them to win, and helping them to win. There is a corresponding need for resoluteness and agility in cutting one's losses early when predictions go wrong.

The approach taken for IPA local area networks (LANs) is an example of a successfully executed technology and standards intercept strategy. It was foreseen that the interconnection technology most likely to become the early market leader was baseband CSMA/CD ('carrier sense multiple access with collision detection', which is more widely known by the name 'Ethernet'). It was also apparent that the general acceptance of CSMA/CD in the market place, and the vital availability of low-cost off-the-shelf silicon, were both dependent on early world-wide agreement on standards.

Therefore, ICL in collaboration with a number of other companies promoted the accelerated production of the ECMA CSMA/CD standards[6–8] in early 1982. This involved intensive liaison with the IEEE Project 802 standards body in the USA, which is setting what will become the definitive LAN standards world-wide. The IEEE work has been stabilised and accelerated, and the ECMA standards have provided accurate predictions of its outcome. This has enabled early commitment of silicon implementations.

Another example of an IPA intercept strategy is the accelerated development of the ECMA LAN (local area network) and network layer architecture,[9,10] and its early adoption as the main basis for IPA layer 1-4 architecture. This is now confidently expected to be the basis for early achievement of multi-vendor LAN networking, and more general standardisation via IEEE and ultimately ISO.

As may be apparent from the above examples, these intercept strategy activities involve intensive participation in national and international standards bodies, good liaison with government and user bodies, collaborations with other Information Technology suppliers (who are often competitors in the same markets), and close liaison with the component producing industry. These relationships strongly influence the technical content of IPA.

A final point should be made about intercept strategies. The one sure thing is that the predictions will always be wrong, even if only in some minor details. Therefore, the architecture and its implementations need to have a modular structure such that the effects of this uncertainty can be confined, and the risks controlled. The IPA investment is in architectural decoupling and structure, not in fallible forecasts.

3.3 Standardisation Problems

As already explained, IPA is targeted on networking standards which are generally referred to as open system interconnection standards, or OSI. Such standards are currently at various stages of development in standards bodies such as ISO, CCITT and ECMA. It will be many years before there are comprehensive, fully ratified, unambiguous and widely used OSI standards. IPA must overcome these temporary shortfalls.

It is likely that the long term outcome of OSI standardisation will not actually be a single complete set of standards, universally used. Some diversity (probably considerable diversity) is likely: there will somtimes be different standards for the same things, and there are likely to be local (typically national) variants. The standards will also be to some degree incomplete in their coverage, because the subject area is large and rapidly evolving, and the standardisation process is inevitably selective and relatively slow. IPA must overcome these more or less permanent difficulties.

These standardisation problems of diversity and change give further emphasis to the need for IPA to have a highly resilient and modular structure in order to achieve coherence and stability.

A further obstacle is that OSI carries a high risk of unsatisfactory performance. Standardisation is essentially a 'political' process of finding compromises between different technical proposals and different established interests, rather than a single minded quest for technical excellent. The standardisation is also fragmented, in that it is necessary for different bodies to develop standards for different parts of OSI in parallel, and inevitably they cannot always achieve fully co-ordinated results. These circumstances impact performance.

Since IPA uses OSI standards for its native mode of operation, a careful selection and integration of the standards is necessary in order to achieve an efficient and resilient IPA design, with high throughput and fast response times. This is another major factor in charting the course of the standards intercept activities, and in deciding the structure and technical content of IPA.

3.4 Silicon

It is expected that well before the end of this decade, all the key networking standards will be available in low-cost off-the-shelf silicon. The world is busy moving the most popular and promising protocols into silicon. Once in silicon, a function tends to become low cost with high volume production. The particular implementation tends to become a stable and universally used industry standard, with multiple suppliers, and with specification and conformance under the control of a leading supplier. Note that this cracks the conformance problem which is currently of much concern to standards makers (because most equipment would use the same industry-standard silicon implementations of the international standards). The result is that protocols-in-silicon will achieve low cost, wide applicability, high quality, assured compatibility and probably market dominance. This is therefore judged likely to be the best route to commercially successful networking.

It is foreseen that the ultimate technical content of IPA (e.g. which communications protocols in what combinations) will be largely determined by silicon. Therefore, the technical content of the architecture and the intercept strategies are strongly oriented towards extracting the maximum possible benefit from collaborations with the semiconductor industry.

4 Modularity

The absolute necessity for the highly modular structure which IPA employs has now been established. The next step is to explain how IPA achieves this modularity.

4.1 Kernel and Affiliates

Probably the most important modularity within IPA is the distinction between Kernel IPA and Affiliate networking architectures. There is *one* Kernel IPA architecture, which is fully under ICL design control. There are *several* Affiliate networking architectures which are mostly outside ICL control. All these architectures are kept largely *separate* from one another (see Figure 2).

Kernel IPA is the native mode of IPA operation, and is the primary means of

Networked Product Line networking. It supports a rich functionality with high efficiency, and will be most widely available in ICL products.

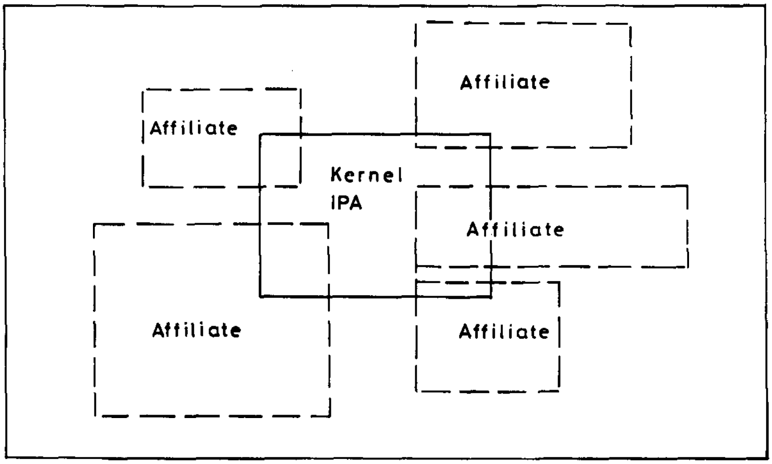


Fig. 2 Kernel and Affiliates

The Kernel IPA architecture is built around the ISO 7 layered model (see Reference 3). It is a specific implementation of the OSI architecture, using selected OSI standards, which are supplemented by use of ICL specifications where suitable standards either do not yet exist, or do not cover needs particular to ICL systems. Kernel IPA is designed to migrate with minimum disruption to full use of OSI standards when they emerge. An essential point is that the choice of Kernel IPA content, and the implementation of changes to it, are controlled by ICL, even though the detailed standards are mostly externally specified public standards. This provides a basis on which the Kernel IPA can be stable and highly tuned, even during the current immature stage of OSI standards development. The kernel architecture is naturally the focus of the IPA OSI standards intercept strategies.

The Affiliate networking architectures are a separate matter. Currently the following are the main affiliate architectures and protocols:

- *FXBM:* the IPA phase 2 full extended basic mode (FXBM) communications and device control protocols
- *Asynchronous device protocols:* 'industry standards' used for teletype, 'glass teletype', and various other device attachments
- *SNA:* the IBM systems network architecture, selected parts of which are used by IPA to provide access to IBM systems
- *CCITT VAN protocols:* teletex, videotex etc. for access to PTT-provided value added networks (VANs), and for equivalent private services
- *OSI affiliates:* OSI protocols which are supported by IPA, but are not currently included in Kernel IPA.

Communication between Kernel IPA and these affiliates is by means of the IPA boundary functions, described later. This inbuilt systematic ability to use and to interwork with other different networking architectures greatly extends the field of application of IPA, and is one of its most valuable features.

4.2 *Kernel IPA Structure*

The next most important IPA modularity is the layering of Kernel IPA. As already mentioned, the ISO layering[3] is used to achieve functional separation of different aspects of data communications. Particular selections and grouping of the ISO layer services have been chosen to maximise Kernel IPA stability, modularity and efficiency.

The *transport service* is the most stable and universal architecture boundary. In standard-making circles it is often referred to as the 'great divide'. Furthermore, it firmly separates the relatively stable layer 1-4 standards from the relatively unstable and inevitably heterogeneous higher layers.

In Kernel IPA, the functions below the transport service are referred to collectively as the IPA telecommunications function (TF). The Kernel IPA layer 5 and 6 functions above it are collectively referred to as the IPA data interchange function (DIF). Above the data interchange service (DIS) is a modular set of IPA interworking facilities. Above them are the information processing functions which are the users of IPA communications. It is foreseen that OSI standards will structure the application layer in similar ways. This modularity is illustrated in Figure 3.

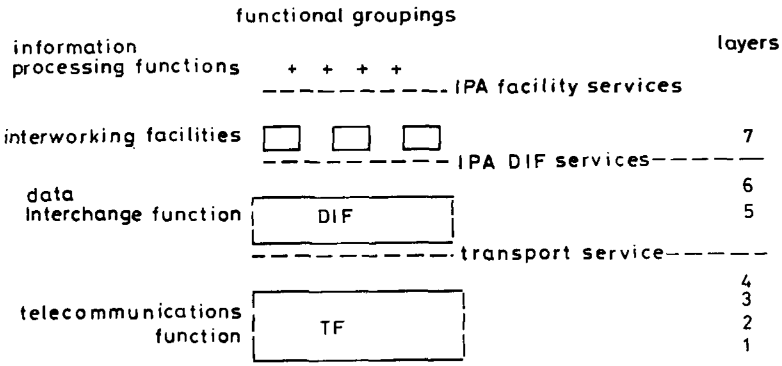


Fig. 3 Kernel IPA layering

The Kernel IPA *telecommunications function* (TF) is designed to use a wide range of network types, both local area (LAN) and wide area (WAN), and to provide communication across the boundaries of multiple networks in series. It is particularly oriented towards efficient and highly resilient communication within distributed systems which are built around clusters of LANs, with WAN interconnection between sites. For this purpose, it uses an OSI class 4 transport protocol and, where needed, an internet datagram. This is the ECMA TR13 and TR14 architecture (see References 9 and 10), which was previously referred to in the discussion of intercept strategy. The TF architecture can use any kind of network; the acutal choices are product decisions. Two of the main types supported are CSMA/CD and X25.

The initial service provided by the Kernel IPA *data interchange function* (DIF) is restricted to what is judged to be a stable core likely to emerge from ISO, while being sufficiently comprehensive for general use. It is based on a selection of the services defined in the ECMA session and presentation layer standards, ECMA75, ECMA84 and ECMA86 (see References 11, 12 and 13), which are judged to be a good prediction of what is likely to emerge from ISO. The services provided at this level are classified into four groups:

- connection control
- environment control
- dialogue control
- data transfer.

The DIF connection control services are those derived from the underlying presentation, session and transport layers. The DIF environment control services are derived from the presentation layer. They provide a means to select, negotiate and envelope any transfer syntax (i.e. the data representation as visible in the protocol) The DIF dialogue control services are derived from the session layer. They provide synchronisations and performance optimisations to handle the uncertainty and transit delay effects inherent in communication. These dialogue controls are also used when gatewaying the communications control procedures of affiliate architectures via the IPA boundary functions, as described later. The DIF data transfer services are derived from the transport layer, augmented by session layer controls where appropriate.

Details of the *interworking facilities* are given later, under the heading of Interworking.

Kernel IPA has a further dimension of modularity, which is the systematic use of standard mechanisms to select dynamically a particular protocol where there is a choice of separate protocols. This modularity and selectivity is inherent in the service access point addressing of the ISO model,[3] and in OSI protocol negotiation techniques.

Note that the modularity of Kernel IPA is based on the stability of its inter-layer *service* specifications, not its protocols. The chosen initial set of TF and DIF services are unlikely to be disrupted by changes in the international standards developments. Some of the protocols are more exposed to such variability, and they can be changed when needed, without disrupting the stability of Kernel IPA.

4.3 IPA Boundary Functions

Communication between Kernel IPA and the affiliates is via what are called IPA boundary functions. These handle the affiliates in such a way that there is a minimum intrusion of their particular characteristics into the core design and implementation of IPA.

The IPA boundary functions use standard techniques for gatewaying between different protocols. These characteristics are latent in the ISO layering (application layer, presentation layer and session layer concepts). They are explicitly supported in Kernel IPA. The basic concept is that if one analyses *any* data-communications protocol (e.g. any IPA affiliate protocol), no matter what its actual design or tangible structure, it will always have a 'deep structure' of four sets of characteristics. These are:

- application semantics
- data image characteristics (i.e. the nature of the data objects)
- transfer syntax capabilities
- end-to-end communications structures.

These logical characteristics are necessarily exactly the same at both end points. The actual detailed means of communicating them need not be the same at both end points, because they can generally be masked and converted invisibly by an intervening gateway without affecting the logical characteristics of the interaction, and without disturbing its logical correctness and consistency.

The Kernel IPA DIF service provides a completely general mechanism for communicating the logical characteristics of any protocol. Its environment control services can carry any transfer syntax mechanism. Its dialogue control and data transfer services can map all of the end-to-end communication structures which are in general use. Any residue of unmappable data transfer characteristic may be encoded into the chosen transfer syntax.

Therefore, an IPA boundary function has the possibility of gatewaying *any* affiliate IPA protocol into the Kernel IPA environment, by mapping its end-to-end data communications characteristics onto the DIF service, and choosing (via the DIF syntax control service) a suitable transfer syntax for use in Kernel IPA presentation layer. Kernel IPA includes sets of transfer syntax which match the characteristics of its various affiliates. Their encoding is as close as possible to that of the affiliate protocol, so that protocol conversion overheads in the IPA boundary function can be kept to a minimum. Figure 4 illustrates this IPA boundary function architecture.

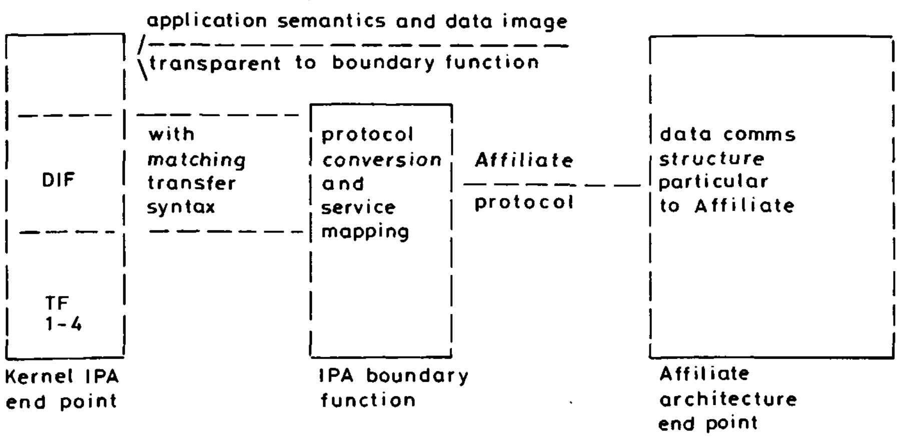


Fig. 4 IPA boundary function, high gateway

The resultant gateway in the IPA boundary function is referred to as a high gateway, because it masks the detailed structure of the affiliate protocol right up to a high level. This minimises the intrusion of the affiliate architecture characteristics into IPA. The boundary function also includes low gateways, which can allow as much as is wanted of the structure of an affiliate protocol to be carried across the Kernel IPA TF or DIF service. This reduces gatewaying overheads to a minimum, but requires a complete 'pillar' of affiliate architecture to be implemented in the IPA end point.

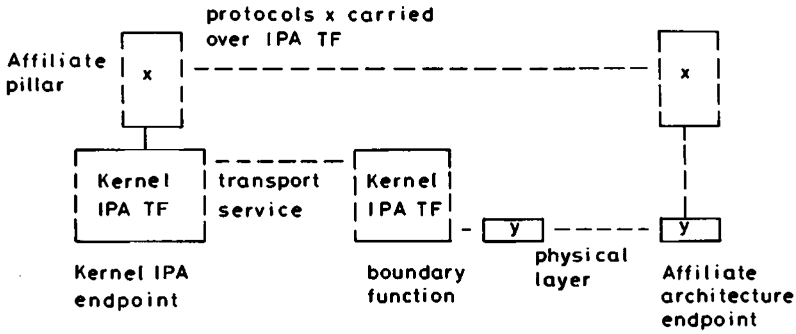


Fig. 5 IPA boundary function, low gateway (extreme case)

Figure 5 illustrates an extreme case where most of the affiliate protocol structure x is carried over the Kernel TF transport service to a complete affiliate pillar in the Kernel IPA endpoint. More usually the Affiliate is layered in such a way that only its upper levels of protocol need be exposed: e.g. layer 5-7 protocols would be end-to-end between a layered Affiliate endpoint and the corresponding affiliate pillar in the Kernel IPA endpoint.

In both cases, the effect is that endpoints within Kernel IPA gain distributed access to the affiliate network, by using the Kernel IPA communications system and network media to communicate with the boundary function. This is closely matched to the concepts of distributed end systems, end system gateways and generic implementation models, which are under development in ECMA at the time of writing.

The choice of whether to use high or low gateways in a particular product instance of the IPA boundary function is largely tactical. It depends on performance trade-offs, the current state of external standards, availability of silicon, etc. In the long term, the availability and modularity of off-the-shelf silicon implementations of protocols is likely to decide choices of gateway structure. For the present it is necessary to keep the options open.

It should be noted that communication with an Affiliate endpoint does not necessarily involve a boundary function gateway. A product may implement the affiliate protocols directly.

4.4 Network Servers

Another kind of modularity is provided by using the network server concept. This is particularly important in Kernel IPA.

In a networked information system there are various commonly used services which are accessed across the network. Standard classifications of these are emerging, together with standard definitions of their services, and standard Network Server entity types. Some examples are:

- print server
- file server
- directory server
- mail server.

Within each type there may be a range of subtypes (e.g. character printer, image printer), with appropriate standard protocols for remote access to the server entity (the provider of the service) by client entities (for users of the service).

The networking architecture defines the server entity types, their services and the server access protocols. This does not constrain the implementation of a server entity, which may be in a free-standing dedicated unit, or part of some larger machine with multiple roles. The server access protocols are invariant to these implementation circumstances, and therefore provide long-term architectural stability of IPA implementations.

5 Architecture

This is a brief overview of the architectural content of IPA from the viewpoints of interconnection, interworking and network management.

5.1 Interconnection

The connectivity provided by the IPA interconnection architecture is data communication, end-to-end between concurrently active endpoints. Within this there is a degree of voice-data-image integration, in that certain communications media are shared for these different uses.

An IPA interconnection network, such as that illustrated previously in Figure 1, is described in terms of nodes, which are interconnected via communications media. In architectural terms, a node is an abstract object type with certain defined data communication characteristics. The tangible embodiment of a node is in a product whose data communication provisions conform to a node type. Products will often implement more than one node type, and thereby perform a composite role.

There is a classification of connectivity attributes, by which generic node types are distinguished. Some of the main node types are as follows:

(i) *IPA Kernel node:* This type supports standard Kernel IPA protocols.
(ii) *Affiliate node:* This type supports protocols different from the Kernel protocols, but for which there is generally IPA boundary function gate-waying.

(iii) *IPA boundary function node:* This type supports Kernel IPA protocols and a set of affiliate protocols, and provides means of gatewaying between them.
(iv) *Internet node:* This type provides a means of communicating between separate access-networks. Different subtypes operate on protocols up to layer 1 or 2 or 3, and protocol above this is passed through transparently.

Within this broad classification of node types there is subtyping to classify different variants. These node types also have various generic interworking and network management characteristics, which are defined in those other parts of the architecture.

The mainstream of Kernel IPA connectivity is between Kernel nodes. If multiple networks are traversed, connectivity is via the intermediary of one or more Internet nodes. Kernel IPA connectivity and its layering are described more full in Reference 14.

Connectivity between Kernel IPA nodes and affiliate nodes is via the appropriate boundary function node type, as illustrated previously in Figures 4 and 5.

Connectivity within the Affiliate networking architectures is provided in ways specific to them. There is a product choice whether to communicate directly with affiliate networks by implementing a complete Affiliate node within a product, or to go via the intermediary of a boundary function node in another product.

5.2 Interworking

The IPA interworking architecture provides a set of data communication facilities, end-to-end between concurrently active endpoints. There is also provision for voice and image communications encoded as data, and for stored message communications.

In Kernel IPA, the interworking architecture is mainly concerned with layer 7, and is therefore distinct from the Kernel IPA interconnection architecture (DIF and TF), which is concerned with layers 1-6. In various Affiliate architectures these distinctions are less clear cut.

In Kernel IPA, there is a set of generic interworking facilities. The main facility types are as follows:

- distributed filestore facility
- job transfer facility
- message transfer facility
- direct data interchange facility
- remote service selection facility
- distributed transaction facility
- terminal access facility.

Some of these Kernel IPA interworking facilities are also supported in various of the Affiliate IPA architectures; most importantly their phase 2 predecessors are fully supported in the FXBM transitional IPA architecture, and the corresponding

IPA boundary functions support interworking with them.

The detailed content of the IPA interworking architecture is described in Reference 5.

5.3 Community Management

The network management facilities of IPA are built around the concept of community management. An IPA community is a grouping of systems which interwork and are managed in a coherent way.

Community management is a set of facilities and mechanisms which help in those behind-the-scenes functions necessary to plan and install networked systems, maintain them in working order, and ensure their smooth day-to-day running.

The main subject areas covered are as follows:

- community access control
- community accounting
- configuration management
- software distribution and updating
- directory management
- telecommuniations management
- statistics services
- fault and error management
- community modelling, capacity sizing, reliability sizing.

The community management architecture is built around a set of abstract object types which it manages. These object types have standard attributes and standard inter-relationships, which are to a considerable degree independent of the variability of their real instances. Some examples of these object types are:

- user
- account
- service
- location
- system
- hardware unit.

In Kernel IPA, most of the management characteristics are supported by application layer protocols, built on top of the standard IPA interworking facilities. Certain aspects are supported by specialised low-level protocols (e.g. for primitive aspects of LAN station management).

The nature of the above Kernel IPA management protocols is such that they are largely network independent, and to this degree they are also applicable in Affiliate networks. Some Affiliate architectures also have their own in-built management facilities, as do some of the layer 1-3 network types.

As yet, there are no OSI standards for management protocols. Therefore, proprietary designs are used, and these are based on designs being developed by standards bodies whenever possible.

A wider view of the IPA Community Management architecture is given in Reference 15.

6 Conclusion

This paper has given an outline of the market and technological environment in which ICL has developed its networking philosophy. This philosophy has been given practical expression in the ICL Networked Product Line and the IPA architecture. The paper has examined the strategic goals of IPA, and described the architectural approach which has been evolved to achieve these goals.

The resultant architecture can be viewed as a piece of practical engineering to move open systems interconnection standards into a coherent implementation framework, which also provides comprehensive interworking with other coexisting network architectures.

7 Acknowledgments

I would like to acknowledge the contributions made by many ICL colleagues over many years to develop ICL networking architecture. I would also like to acknowledge the enormous talent, enthusiasm and dedication of standards colleagues from many different organisations and countries. Without their work in bodies such as BSI, ECMA, CCITT and ISO, Open System Interconnection would be impossible.

References

1 HOULDSWORTH, J.: 'Standards for open network operation', *ICL Tech. J.*, 1978, **1**, 50-56
2 BRENNER J.: 'Using open system interconnection standards', *ICL Tech. J.*, 1980, **2**, 106-116
3 Draft international standard, DIS 7498: 'Data processing – open systems interconnection – basic reference model'
4 KEMP, J., and REYNOLDS, R.: 'The ICL information processing architecture IPA', *ICL Tech. J.*, 1980, 2, 119-131
5 LLOYD, R.: 'IPA data interchange and networking facilities', *ICL Tech. J.*, 1983, 3 (3), 250-264
6 Standard ECMA 80, Local Area Networks (CSMA/CD baseband) Cable System
7 Standard ECMA 81, Local Area Networks (CSMA/CD baseband) Physical Layer
8 Standard ECMA 82, Local Area Networks (CSMA/CD baseband) Link Layer
9 Technical Report, ECMA TR13, Network Layer Principles
10 Technical Report, ECMA TR14, Local Area Networks, Layers 1-4 Architecture and Protocols
11 Standard ECMA 75, Session Layer Protocol
12 Standard ECMA 84, Data Presentation Protocol
13 Standard ECMA 86, Generic Presentation Services and Protocol
14 TURNER, K.: 'The IPA telecommunications function', *ICL Tech. J.*, 1983, 3 (3), 265-277
15 GOSS, S.: 'IPA community management', *ICL Tech. J.*, 1983, 3 (3), 278-288

IPA data interchange and networking facilities

R.V.S. Lloyd

ICL Network Systems Division, Kidsgrove, Staffordshire

Abstract

The paper 'IPA networking architecture' by J.B. Brenner gives a general picture of the overall principles behind the design of IPA and outlines the modular layered architecture – telecommunications, data interchange and facilities. This paper takes as its starting point the transport service described by Turner in his paper 'IPA telecommunications function' and indicates the nature of the IPA facilities built on it for interworking between IPA nodes. The paper distinguishes between those facilities which are an integral part of IPA, the Kernel facilities, and those which belong to Affiliate architectures, supported via boundary functions.

1 IPA interconnection and interworking

The objective of IPA communications is to permit application processes to interwork irrespective of the nature of the data transfer path between them. The majority of the standardisation inherent in IPA is therefore directed towards the provision of IPA protocols for the purposes of interconnection between systems. IPA's architecture is structured in such a way that it can evolve to support international and industry protocol standards with the maximum of efficiency and the minimum of user awareness of its detail.

However, there are aspects of interworking which are common to many separate application areas, and which are generally supplied to the user as part of the operating system or programming environments. IPA identifies a number of such aspects and provides definitions of IPA interworking facilities in terms of:

– the service they provide to their users
– the protocols by which the interworking is carried out, and
– the nature of the interconnection service they use.

This approach conforms to the architecture defined by ISO in the basic reference model for open systems interconnection (OSI)[1] and to the work in progress for production of international standards in the application layer of the model. However, it will be some time before ISO standards are ratified either for application layer protocols or for the presentation and session services they will use, and the definition of an inner architecture for the ISO application layer services is still in the early stages of discussion. IPA therefore adopts a practical approach towards providing users of the ICL Networked Product Line with the basic capabilities they need for interworking, while anticipating the eventual advent of standards for OSI.

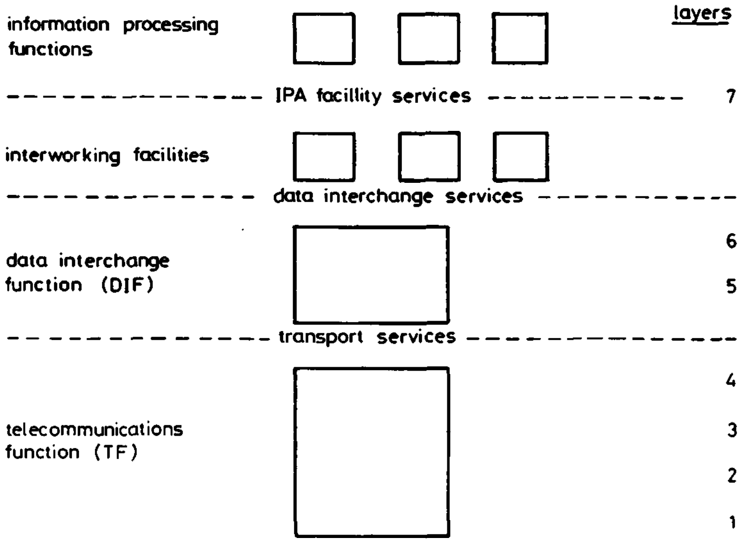


Fig. 1 Kernel IPA layered structure

IPA interconnection falls into two main sections – the telecommunications function, defined in the paper 'IPA telecommunications function'[3], and the data interchange function, which is described in this paper. These correspond to ISO layers 1-4 and 5-6, respectively.

IPA interworking is defined as a number of separate functional facilities, some of which will have their analogues in ISO layer 7 services, and some which provide applications with relatively transparent access to IPA interconnection functions.

2 IPA data interchange – an evolving service

As with IPA in general, so with the data interchange function (DIF): its target is to conform to standards for OSI. Its functionality is made available to its users in the form of a data interchange service (DIS) just as the telecommunications function is made available as the transport service (see Fig. 2). In OSI terms, the DIS is seen by the application layer as a presentation service, which transfers data between application entities.

An application entity is the set of functions within an application process which is involved in providing intercommunication with another such process. This is an architectural abstraction of the ISO-OSI reference model, and may comprise a number of separate layers of protocol. It is represented in practice by those aspects of a terminal, or of a user or system program, which are aware of the existence of another. This is illustrated in Fig. 2.

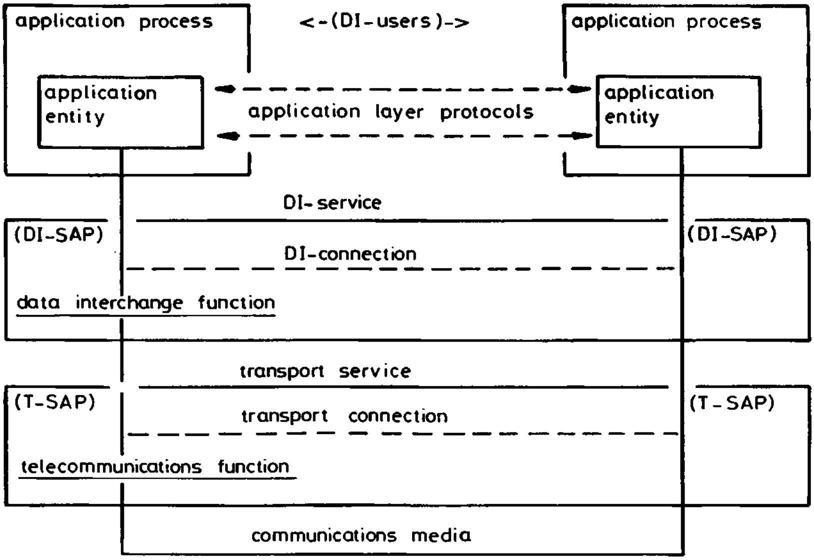


Fig. 2 IPA interconnection architecture

2.1 *Kernel IPA DI-service characteristics*

- *Connection establishment.* A DIS-user requests the service to establish a data path to another known DIS-user which is located via a DI-service access point (DI-SAP). A DI-SAP is referenced by its name (i.e. a DI-address). The DI-user has the ability to select or negotiate certain operational characteristics (the *DI-environment*) of that path with the remote DIS-user. There is no visibility to the DI-users of the actual transmission paths used, or of the nature of the protocols within the DI-service which will be used.

- *Connection termination.* An established connection will exist until it is terminated either by explicit request from one of the DIS-users or by the service itself. Termination can be either *orderly* (at a point in time when both DIS-users are ready for the event, with no data in transit between them) or *disruptive* as a result of unilateral action by one DIS-user or of an error condition with the service.

- *Transfer environment selection.* Data transfer over an established connection takes place in the context of a set of control parameters known as a *DI-Environment.* The main feature of such an environment is its *transfer syntax* – the way in which the meaningful data being exchanged between application entities is structured and encoded during its passage over the connection. A number of such syntaxes is defined by IPA and supported by appropriate translation and mapping functions in the Networked Product Line implementations. Examples of such transfer syntaxes include the data and format control of a video terminal's screen, a simple character string for message transfer, complex structured data being exchanged between word processors, and so on. The environment selected during connection also includes a negotiated choice of available DIS functional options.

- *Data transfer.* The actual process of transferring data in either direction according to the currently established DI-environment.

- *Dialogue control.* Functions are provided to enable the communicating entities to synchronise their use of the connection and to control the right of one of the entities to use a particular DI-service function at a particular time (token management).

Provision of some or all of this set of facilities for data interchange is by means of one or more layers of protocol, which in turn are mapped onto the transport service provided by the telecommunications function described in Turner[3]. The OSI reference model defines two such layers, presentation and session, which are functionally distinguished as described below.

The DI connection and disconnection services are provided by the presentation and session layers, and mapped onto the underlying transport services. The DI-environment selection services are derived from the session layer and the DI data transfer services are derived from the transport layer, augmented by session layer controls where appropriate.

2.2 IPA DIF intercept strategy

IPA, which of necessity exists before the formal definition of ISO standards for session and presentation layers, adopts an intercept strategy for defining early solutions to the provision of required functions[4]. In Phase 2, its currently implemented form, the interworking facilities of IPA use the Full XBM Affiliate architecture[16] as the basis for the definition of device access protocols and transfer syntaxes (collectively known as 'access levels'), suitable for interactive display data or bulk printer data and for controlling dialogues and resynchronising data transfer connections.

The next stage (Phase 3) redefines the DI-environment inherent in the FXBM protocols in terms of a DI-service operating over the transport service, and provides FXBM-based transfer syntaxes and device access protocols. It also paves the way for the introduction of new classes of transfer syntax to support application requirements not handled by FXBM, and for the adoption of OSI standards for session and presentation protocols.

Just as the IPA telecommunications function[3] makes early use of ECMA standards for the transport protocol and CSMA/CD local area networks in advance of ISO standards, so too the IPA data interchange function relies on ECMA in this transitional period. ECMA has produced standards for session (ECMA-75), generic presentation (ECMA-86) and data presentation (ECMA-84) protocols, and these are in advance of the equivalent ISO drafts. At the current stage the DI-service is defined compatibly with the ECMA session and generic presentation services, and an integrated DI protocol is defined which is derived from the ECMA-75 session protocol.

Another aspect of ECMA work which is currently adopted by IPA DIF is the relationship of ECMA virtual terminal protocol to the presentation and application layers of the OSI model. ISO is feeling its way towards an interpretation of the model which places the VT service in the application layer over a generalised presentation service: ECMA regards its generic and basic VT protocols (ECMA-87 and 88) as examples of specific presentation protocols derived from its generic presentation protocol (ECMA-86)[9]. Once the ISO standards for presentation and application layer services and protocols have become established, IPA support for them will be provided.

2.3 DIF Affiliate architectures

Kernel IPA DIF is primarily intended to provide its service between products of the ICL Networked Product Line which support its protocols directly. This means that products which conform to Affiliate architectures will normally be supported indirectly through boundary functions incorporating both 'high' and 'low' gateways, as described by Brenner[4]. In some specific cases, products will contain both Kernal and Affiliate protocol support, thus obviating the need for use of gateways. The support of existing ICL products, such as terminals with their FXBM protocol structure, is made more efficient by the definition of closely equivalent DI-environments on the Kernel side of 'high gateways', and fewer products need to retain direct support for the full Affiliate protocol.

However, there are some Affiliate architectures which are based on the same OSI transport service as IPA. For them, it is often more appropriate in the boundary function to pass their upper layer protocol straight through to the end system, and not perform any conversion on it. This is an example of a transport level 'low gateway', with an Affiliate-specific DI-service. This approach is thought of as building a series of 'DI-pillars' on the common transport 'pedestal'. Such pillars will either require integrated support by an appropriate application process or be mapped onto the standard IPA DI-service for access by application entities in general. A transport-level low gateway is illustrated in Fig. 3.

3 IPA interworking – a practical set of facilities

Unlike IPA data interchange function, the provision of IPA facilities for interworking owes far more to long established ways of designing operating systems and program or data support environments than to the production of international standards for OSL Work is, however, under way in ISO for OSI standards in the application layer, particularly for file and job transfer and for virtual terminal protocols, and these too will be the subject of appropriate IPA intercept strategies.

This paper describes IPA Phase 3 architecture, and at this stage of its development

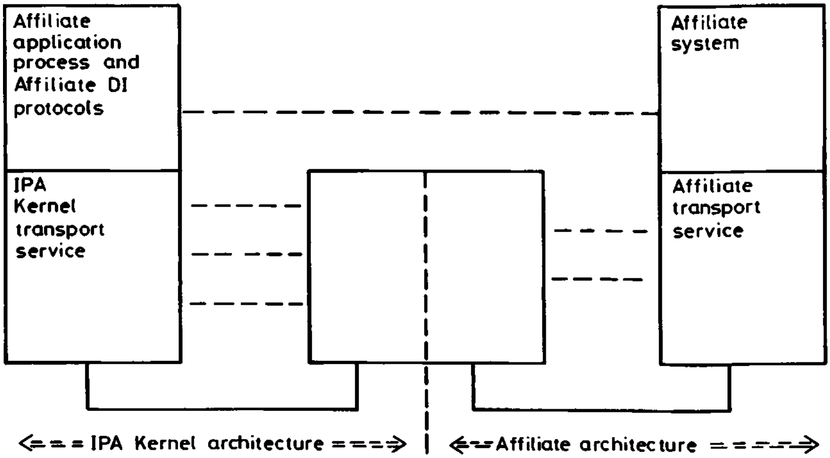


Fig. 3 Low gateway boundary function

the interworking facilities are defined to be compatible with the Phase 2 facilities, which were described in outline by Kemp and Reynolds[2]. A clear indication is, however, available of the direction in which they will evolve for ISO intercept and meet the needs of IPA networking. This section therefore describes each major facility in terms of a generalised description of its function, and the particular way in which it is provided by IPA at Phases 2 and 3.

The facilities described are not the full potential capability of the Networked Product line, but represent the current state of the art. They are:

- terminal access facility
- remote service selection facility
- distributed transaction facility
- direct data interchange facility
- distributed filestore facility
- job transfer facility
- message transfer facility.

3.1 Terminal access facility

3.1.1 Facility outline: IPA provides a direct interface to the data interchange function for the purpose of connecting applications to terminals. This same interface also provides the means for terminals to gain access to other facility handlers such as the remote service selection facility described below, and therefore represents the IPA Kernel view of terminal access protocols.

The richness of function inherent in terminals is reflected by the extreme diversity of transfer syntaxes (see 2.1 above) available at the DI-service interface, and in the DI-environment negotiation and selection functions provided in the presentation layer. The Kernel IPA terminal access facility is therefore simply the provision, at an appropriate application program interface within the supporting system environment, of data input and output, connection control and error notification facilities.

There is a class of transfer syntaxes (possibly related to the ECMA generic virtual protocol[10]) associated with the flexibility of multiple virtual screens and split screen working, which will form the Kernel of IPA and will be generally implemented on Networked Product Line systems. In addition, there will be syntaxes designed to facilitate the construction of high gateways at the IPA boundary to provide interworking with systems which support only Affiliate architecture protocols such as FXBM and 'industry standard' terminals.

3.1.2 IPA Phases 2 and 3: terminal access: In IPA Phase 2, terminal access is provided almost completely through the medium of the Full XBM Affiliate architecture, with a number of 'access levels' (transfer syntaxes) concerned with monochrome character screen displays, 'scroll mode' devices (typically terminals connected by asynchronous line facilities), and employing a number of product specific, or *de facto* industry standard, device access protocols and bulk input and output devices (card readers, printers and input/output spoolers).

These capabilities are supported by ICL terminals, and, in some cases are emulated by 'main frames' as a means of providing simple intersystem communication.

In IPA Phase 3 this method of terminal access remains as one of the application-visible transfer syntaxes; in some systems, the same protocol support is retained as a directly supported Affiliate architecture. But, more generally, high gateways are provided to these Affiliate protocols. New transfer syntaxes for the support of colour and graphics terminals will be provided.

3.2 Remote service selection facility

3.2.1 Facility outline: The first facility to be encountered by the user of IPA networking is the means by which the user of a terminal may be connected to a desired service provided 'somewhere in the network'. The function of handling the requests for selection of such a service is carried out by an IPA *agent*. The system hosting the required service is an IPA *server*. A simple diagram illustrates the main concepts (Fig. 4).

In some cases, a default mechanism in the agent ensures that a predetermined relationship is provided automatically for a given terminal (e.g. the operator of a supermarket till will always expect to be connected to the appropriate checkout control application). But, in general, the first stage of any use of a terminal is to establish a number of levels of routing and authorisation. For example, some or all of the following stages will be needed, as seen from the viewpoint of the terminal's user:

(i) Establish the working state of the terminal (switch on, load program, test etc.).
(ii) Connect terminal to IPA agent function. (This may involve setting

up a switched network call to a computer centre, or the agent function may be built in to the terminal controller itself.)

(iii) Provide agent with terminal identity and terminal user's identity, obtain authorisation to use network, etc.

(iv) Obtain any waiting messages (e.g. system broadcasts).

(v) Obtain list of available services – either a short fixed list or a dynamically selected class of service

(vi) Select required service name and request agent to initiate connection to it (may be local to the agent's system, or remote from it).

(vii) Provide service with terminal user's identity, obtain authorisation to use service, and possibly obtain authentication of the service itself.

(viii) Commence service use.

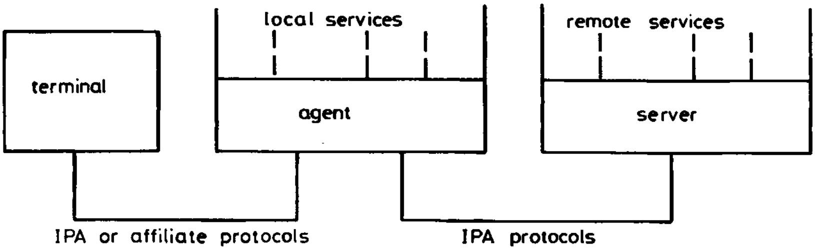


Fig. 4 Remote service selection

The IPA remote service selection facility is concerned with stages (ii) to (vi) of this process, and its objective is to present a common style of operation irrespective of the location of the functions being invoked (for example standard layout of menu display, choice of terminology). Part of this consistency is ensured by the use of IPA community management functions such as access control, accounting and directory management[5].

The facility makes use of the mechanisms of IPA data interchange as they are available and appropriate to the task it is performing. These will vary over time and according to the capability of the individual agent implementations. If a full connection service is available, then stages (ii) to (vi) can be performed by it, perhaps setting up a direct switched network connection to the desired server. If such a service is not provided, then the necessary connections to all accessible agents or services need to be predefined and invoked. If a virtual terminal service is available, then this may provide the means of maintaining access to a number of services in parallel; if not, then the facility may manage such usage itself.

Other aspects of the service, not listed above, include monitoring the connection to deal with faults and failures, to record statistics, to deal with requests for controlled disconnection and reconnection or to switch between services being accessed in parallel. Closely related to this service is the inverse function of arranging for subsequent service-initiated functions, like the output of printed results from the service, to be routed automatically to an appropriate mechanism. This usually involves a path through the same agent.

3.2.2 IPA Phases 2 and 3: RSA: In IPA Phase 2, the remote service selection facility is called *remote session access* (RSA). This is implemented as an agent function in VME, TME, System 25 and NPS, and permits the users of terminals to access services in VME, TME and DME. The role of DRS in the context of RSA is primarily that of a terminal, rather than of an agent; although the user of a DRS-connected screen may have the choice between accessing local DRS applications and remote services. This is not handled within the structure of an IPA RSA agent function.

The transfer path between the Agent and Server is either Full XBM or X.25; the device access protocol used over it is normally equivalent to that of the FXBM video terminal, but, in the case of VME systems, it is also possible to define it as the console of an RJE terminal or as a 'scroll mode' device (i.e. a terminal which operates on a line-by-line basis, rather than on a formatted rectangular screen concept).

RSA agents exercise local mechanisms for stages (ii) to (iv), provide fairly similar, but not identical, selection menus and user controls for stages (v) and (vi) and may make use of predefined connections to the requested server. They also recognise a standard control message for enabling the terminal user to revert to the service selection stage of operation (the 'return to agent' function). RSA permits the user to set up an indirect connection to a requested service through a number of agents in sequence.

In IPA Phase 3, the same RSA facility is provided, but now mapped onto the Kernel data interchange service.

3.3 Distributed transaction facility

3.3.1 Facility outline: A different requirement to that described for service selection is met by the capability offered by a transaction processing (TP) service. This typically incorporates the means of identifying the required processing function either directly from the text of a terminal-originated message or from analysis of the content of a message by a local processing function. For example: a 'message type code' could be used to indicate the difference between a stock update request and a stock issue request, and to cause the messages to be passed to their respective processing applications. Alternatively, the issue of stock items which carry a particular security classification could be identified as such by the stock issue process, and cause it to enquire separately of the authentication process before carrying out the request.

When such routing activity is contained under the control of a single TP service, the routing function is simply one of attaching messages to appropriate queues within itself. When the various processes are distributed between a number of TP services a more extensive control function is required to link them together, to carry out destination analysis and routing functions, to maintain message and conversation sequence, to ensure reliability and recoverability of the distributed system and to relate message texts to their control formats ('templates'). Many of these functions are identified as belonging to a common 'message transfer service', which is described separately below.

Control of interrelated TP services in this way is an IPA interworking facility, which requires the maintenance of data interchange connections between TP services and one or more types of application-layer protocol to provide these functions. A further aspect of the facility is the provision of a standard interface to application processes to simplify and harmonise their view of the TP service and to assist in program portability.

The general model for the distributed transaction facility is shown in Fig. 5,

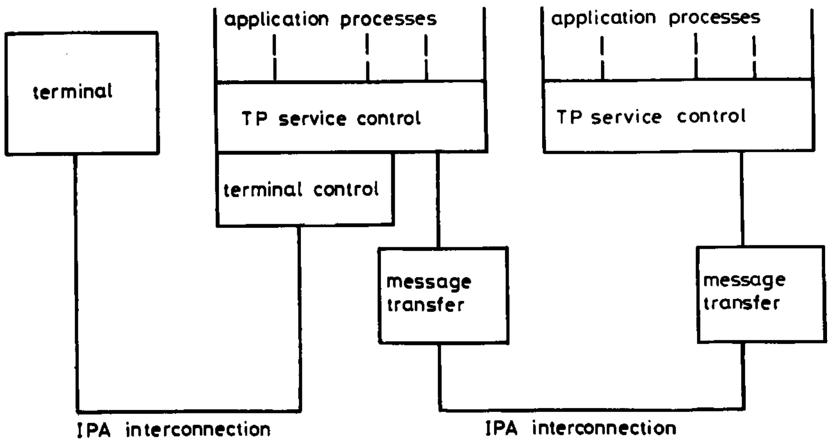


Fig. 5 Distributed transaction service

3.3.2 IPA Phases 2 and 3: DTS: In IPA Phase 2, the distributed transaction facility is called the distributed transaction processing service (DTS), which splits into two subfunctions, distributed message router (DMR) and distributed application facility (DAF). These correspond to the generic capabilities described above as follows:

DMR: routing of terminal messages through one TP service to and from an application process located at a remote TP service; message template handling.

DAF: control of transactions between application processes in separate TP services and inter-TP service system and error management functions.

DTS is supported by VME, TME, DME, System 25 and DRS products to varying degrees of functionality, according to the capability inherent in the product itself. As a result, DTS contains some modes of working which are designed to be more effective between TP services in like regimes than between different regimes.

No significant change in DTS occurs in IPA Phase 3, although, as with RSA, it is defined to use the Kernel data interchange service.

3.4 *Direct data interchange facility*

3.4.1 Facility outline: Similar to the facility for terminal access is the means of enabling a program on one system to converse in an application-dependent way with a program in another system, using an appropriate transfer syntax and data interchange service subset. This, in fact, is identical to the capability provided by IPA interconnection to all the facilities described in this paper, the only difference being the provision of a range-standard interface in high-level language terms to the application programs.

3.4.2 IPA Phases 2 and 3: ADI: In IPA Phase 2 the direct data interchange facility is called application data interchange (ADI) and includes a COBOL definition of the interface to the communications facilities provided by FXBM-DI (the device independent form of FXBM) and X.25. There is no data interchange protocol additional to these communications functions, and the application programs provide their own synchronisation and dialogue control functions within the protocol they use between themselves. ADI is provided on VME, System 25 and DRS. TME provides a similar capability through simple use of DTS.

In IPA Phase 3, the service is enhanced by being mapped onto the Kernal data interchange service, which thus provides for synchronisation and connection control functions.

3.5 *Distributed filestore facility*

3.5.1 The basis of any distributed data processing system is its data; the effects of physical distribution over several sites of the functions of data storage, access and processing are handled within this facility. IPA does not itself define the structure and method of managing a fully distributed database, but it does provide a set of facilities concerned with the movement of filed data from one system to another to aid interworking of the types described in previous sections.

The basic requirement is for a terminal user to cause a file of data to be brought from its current storage location to one where it can be accessed quickly and efficiently by the application or system function he is connected to. While facilities may be available for remote access to data on a record-by-record basis, this can be very expensive in time and communications costs if it is in the 'wrong' place. Frequently, the efficient operation of an application system will depend on the transfer of interactively collected and modified data to a central site for co-ordinated processing and the return of relevant sections of it to the local sites after processing. This facility is generally termed file transfer, and IPA standards will intercept the standards being defined by ISO or other *de facto* standards as appropriate.

As indicated above, there is also a requirement for remote access to data in a file on a per record basis where it is not economical or appropriate for security reasons to transfer the whole file. ISO is defining a protocol for file access. This will be intercepted via an appropriate choice of functions in an earlier IPA protocol.

3.5.2 IPA Phases 2 and 3: FTF: In IPA Phase 2 the file transfer facility (FTF) provides the means of moving files between systems. It uses a file transfer protocol (FTP) based on the network independent file transfer protocol (NIFTP)[17]. It is supported by VME, TME, System 25 and DRS. VME also makes it available in actual NIFTP form for use in non-IPA environments (for example over X.25 and the network independent transport service[14]). It is available over either X.25 or FXBM as its communications service.

In IPA Phase 3 the same protocol is used, but mapped onto the Kernel transport service for use within the ICL Networked Product Line.

3.6 *Job transfer facility*

3.6.1 Facility outline: Not all use of data processing systems is based on interactive terminals connected directly or indirectly to application programs. There is much use of what has traditionally been termed 'remote batch processing', whereby a unit of work is defined in terms of a suitable job control language, transferred to a system capable of carrying it out along with any input data required, and the results of such processing are returned to the originator, generally in the form of printed output. This constitutes the basic outline of a job, but one which is capable of infinite variety in terms of nature of input and output, linking of separate phases in a controlled sequence, return of partial results, enquiries relating to job progress and status etc.

IPA includes the ability to define such a method of running data processing systems and will intercept ISO standards for a job transfer and manipulation service. The essence of this is not in the definition of a cross-range job control language but in a family of protocols for packaging jobs and their input/output data and managing their transfer to appropriate sites in a network.

Work in ISO is still at a relatively early stage, but is based quite strongly on the UK job transfer and manipulation protocol[13]; early implementation of this is planned for VME systems. Both the UK and the ISO work depend on the use of an appropriate file transfer facility to move job descriptions and data around between the relevant systems.

3.6.2 IPA Phases 2 and 3: RRJE: In IPA Phase 2 the nearest equivalent to a job transfer facility is provided by the range remote job entry facility (RRJE). This is the implementation in VME, TME, System 25 and DRS of the emulation of an RJE terminal controlled by Full XBM protocols, with either direct printing of output, or (more usually) spooling for later printing. Jobs may be prepared as files of 'card images' for transfer to a server, which may be VME, TME or DME.

In IPA Phase 3 the Phase 2 support will be retained, and similar transfer syntaxes provided over the data interchange service. However, much more use will be made of file transfer facilities in support of this mode of remote job processing, paving the way for the development of Kernel job transfer facilities.

3.7 Message transfer facility

3.7.1 Facility outline: It has already been noted above that the distributed transaction facility requires the use of a reliable message transfer service, which is an enriched form of the IPA data interchange service. The chief form of such enrichment is in terms of integrity control (ensuring that data not only reaches its destination, but is securely stored there before it is released by the sender) and in the ability to suspend (voluntarily or involuntarily) and restart a unit of transfer. This has much in common with file transfer, but the unit of transfer is generally not as large or defined in so much detail as a file being retrieved from one filestore and deposited in another.

IPA therefore has a message transfer facility which also provides a base for the important application area of electronic mail. This too needs a reliable message transfer service, a high level routing function and the support of the IPA data interchange service. ECMA and ISO are both proceeding down a path which identifies the need for such a service, and IPA will similarly anticipate and intercept that path.

4 Affiliate architecture facilities

The description of the architecture of the data interchange function included references to the ways in which IPA handles interaction between systems using other protocols which are known collectively as 'Affiliates'. In the area of IPA interworking facilities it is equally important that there should be ways in which the universe of IPA facility standards should be made to interwork with affiliate facilities. An example can be found in the Teletex service; not only is it likely that its DI-service and protocols will be supported directly within the IPA Kernel, but also its application functions will be represented within the electronic mail group of facilities.

This shows that the most likely form of support of affiliate facility protocols in an IPA context is by providing it directly, rather than by conversion in a boundary function. It will still be necessary to adopt boundary gateway techniques to handle the DI and telecommunications function support, but conversion of facility protocols, with their wealth of application-oriented significance, is likely to be uneconomic, or even impracticable. For example, conversion of the IPA Phase 2 file transfer protocol into a future ISO standard protocol would not be attempted; either the Phase 2 FTP support would be retained in the host system alongside the new standard or else the file would be stored and forwarded by an 'application level' boundary function.

Examples of this general approach already exist in IPA Phase 2 facilities:

- RSA provides the means for a terminal user to converse with either IPA services or with SNA services through a common agent function. The terminal implements IBM 3270 data interchange functions as well as FXBM video functions, and these are passed through the agent onto either an IPA transport service or SNA.

– DTS provides the means of routing transactions through from an IPA TP environment into an IBM CICS environment.

– FTF on VME permits standard NIFTP protocol support as well as the IPA-oriented form of the facility. File transfer between System 25 and the IBM CICS/VS environment is provided.

– ADI is provided as a means to enable applications programs to communicate independently of any standard facility protocol; VME provides mapping of ADI not only onto the standard IPA telecommunications function but also onto the UK network independent transport service[14], thus enabling ICL and non-ICL-based applications to interwork freely in an X.25 network environment.

– a number of implementations of IBM 2780/3780 emulators exist in ICL products, permitting them to operate as RJE stations to IBM-based services.

It is likely that this general approach will be continued, but where a particular Affiliate facility has a need to be integrated closely into an IPA facility, then the provision of a 'high gateway' type of boundary function will be possible, converting this level of protocol too into an IPA standard.

5 Conclusion

This paper has concentrated on showing how the importance of the upper layers of protocol is growing in the context of IPA and its implementation on the ICL Networked Product Line. Much attention has been focused in the past on the achievements of intercommunication in the transport, network and link layers of protocol. IPA is now progressing to higher levels of standardisation by bringing order to the problems of data interchange protocols and by defining standards for practical interworking facilities. The economic advantages of a well defined, optimised Kernel set of standards are enhanced by practical use of gatewaying techniques to preserve the means of communicating between them and the many-coloured world of existing and non-IPA facility standards.

Acknowledgments

Thanks are due to the many members of ICL who have contributed to the evolving design of IPA data interchange and facility standards. Practical experience of the system implementation teams provides good evaluation criteria for the architects and standards developers. This aspect of IPA owes much to the continuing work of the UK and international standards communities.

References

1 'Data processing open systems interconnection – basic reference model', Draft international standard, DIS 7498
2 KEMP, J., and REYNOLDS, R.: The ICL information processing architecture IPA. *ICL Tech. J.*, 1980, 2, 119-131
3 TURNER, K.: 'The IPA telecommunications function', *ICL Tech. J.*, 1983, 3 (3), 265-277
4 BRENNER, J.B.: 'IPA networking architecture', *ICL Tech. J.*, 1983, 3 (3), 234-249

5 GOSS, S.: 'IPA community management', *ICL Tech. J.*, 1983, 3 (3), 278-288
6 'Session layer protocol', Standard ECMA 75
7 'Data presentation protocol', Standard ECMA 84
8 'Virtual file protocol', Standard ECMA 85
9 'Generic presentation services and protocol', Standard ECMA 86
10 'Generic virtual terminal service and protocol', Standard ECMA 87
11 'Basic class virtual terminal service and protocol', Standard ECMA 88
12 'Network independent file transfer protocol', DCPU, 1980
13 'Network independent job transfer and manipulation protocol', DCPU, 1981
14 'Network independent transport service', PSS User Forum (DCPU), 1980
15 BENNETT, C.J.: 'JNT mail protocol', University College, London, 1982
16 'Full-XBM protocol – ICLC-03' RP3105, 1982

The IPA telecommunications function

Kenneth J. Turner

ICL Network Systems Division, Kidsgrove, Staffs

Abstract

ICL's information processing architecture (IPA) is closely aligned with ISO's work on open systems interconnection (OSI). The IPA telecommunications function, which realises the lower four layers of the OSI reference model, is at an advanced stage of design and implementation. The IPA telecommunications function architecture, its services and its protocols are described. An introduction to OSI concepts and a selective glossary of IPA terminology are also provided for the non-specialist.

1 Introduction

1.1 The Present

The concept of open systems interconnection (OSI)[1,2] will be familiar to many. However, non-specialists would be excused for thinking that OSI was still very much an intellectual exercise with as yet little practical relevance to the inter-working of computer systems over communications links. But ICL has taken an early lead in the implementation of the latest OSI telecommunications standards and already has products intercommunicating using these. The exciting thing about these standards is that they are internationally approved and are truly open: other manufacturers are implementing them or will implement them, and all will be able to interconnect. The purpose of this paper is to explain the architecture and protocols which make this possible.

1.2 The Past

Data communications in the past has been plagued by two problems: an inter-mingling of functions, resulting in monolithic designs, and the use of proprietary protocols, resulting in incompatibility between different manufactuers. Over the past 30 years a number of significant steps have been taken towards resolving these problems, most recently the development of the open system interconnection basic reference model[3] by the International Organisation for Standardisation (ISO). The reference model prescribes an abstract architecture which frames the functions necessary for open interworking between computer systems. Although the reference model uses abstract concepts so as not to constrain any real implementation, it is not hard to see the real-life counterparts of these abstractions. For example, service access points could be thought of as system interfaces, and entities as sub-systems. The reference model has served as the basis for developing a modular and manufacturer-independent set of communications protocols, some of which are described later in this paper.

ICL has played a leading part in communications developments for many years. The significance of the layering principles found in OSI was recognised early in ICL and applied to the development of ICL's own protocols. In recognition of the growing importance of OSI, ICL announced its information processing architecture (IPA)[4] with a clear commitment to OSI standards. This commitment continues unabated, and ICL is active in ISO, the European Computer Manufacturers, Association (ECMA) and elsewhere to further the development and implementation of OSI. The scope of IPA is very wide and diverse. IPA seeks to control this diversity by distinguishing clearly between the Kernel, or central, part of IPA and its Affiliate parts[5]. Some examples of this are given later in this paper.

1.3 The Future

Services and protocols at almost all layers of the OSI reference model are at an advanced state of definition. This is particularly true of the lower four layers, which collectively form the IPA telecommunications function (TF). ICL has taken standards for these lower layers and turned them into a fully-fledged product architecture. The IPA telecommunications function architecture, its services and its protocols are the subject of this paper. Sections 2, 3 and 4 deal with each of these aspects in turn. The IPA approach to the interconnection of networks is discussed in section 5. A glossary of some IPA and OSI abbreviations is provided at the end.

2 Telecommunications function architecture

This section introduces some basic OSI concepts which underlie the IPA telecommunications function. Later sections expand on these, with particular emphasis on the strengths of the IPA realisation of OSI.

2.1 Layering

The OSI reference model specifies seven principal layers as illustrated in Fig. 1. The telecommunications function spans the lower four layers, namely those concerned with the movement of data from one system to another without regard to the meaning of the data. Later on in the paper the characteristics of the lower four layers are described in detail, but for the moment it is sufficient to note that their functions are:

4 – (Transport) end-to-end data transfer and cost optimisation
3 – (Network) routing of data between systems
2 – (Data Link) transfer of data across a single communications link
1 – (Physical) access to the communications medium

at the end. Companion papers in this issue describe the general strategy and future directions for ICL networking[5], IPA Networking Facilities[6] and IPA Community Management[7].

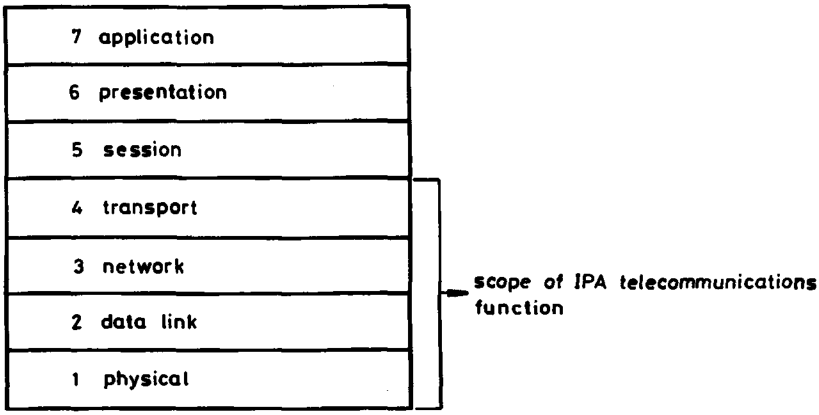


Fig. 1 OSI layering and scope of IPA telecommunications function

2.2 *Layer Structure*

Each layer offers a set of *services* to the layer above and relies on the services of the layer below. The services of a layer are obtained through exchange of *primitives* operating at a *service access point* (SAP), which is the abstract representation of an interface. Within a layer the functions are performed by *entities*, which communicate using a *protocol*. The OSI layering rules require that all communication occurs on a peer-to-peer basis within a layer. The means whereby a layer fulfils its functions is visible to the next layer up only in terms of the service primitives. In particular, the protocol used in a layer is not visible from above, which means that protocols may be changed without affecting upper layers. The architecture therefore accommodates advances in hardware or software technology with minimal impact. Fig. 2 summarises the structure of a layer and the terminology used to describe it.

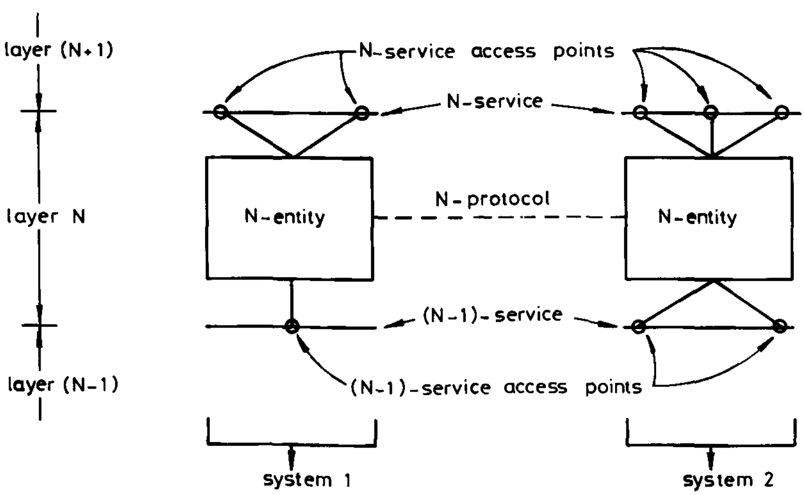


Fig. 2 Layer structure and terminology

2.3 *Sub-Layering*

The reference model recognises the concept of *sub-layering* as a technique for partitioning of a layer. As illustrated in Fig. 3 this is used in IPA to define the finer detail of the network and data link layers. The sub-layering of the network layer permits a convenient split of the telecommunications function itself into three divisions: transport, multinet and subnet. The transport function, in concert with the others, provides the required end-to-end data transfer facilities. The multinet function looks after the interconnection of networks, whereas the subnet function concerns itself with the structure of individual networks.

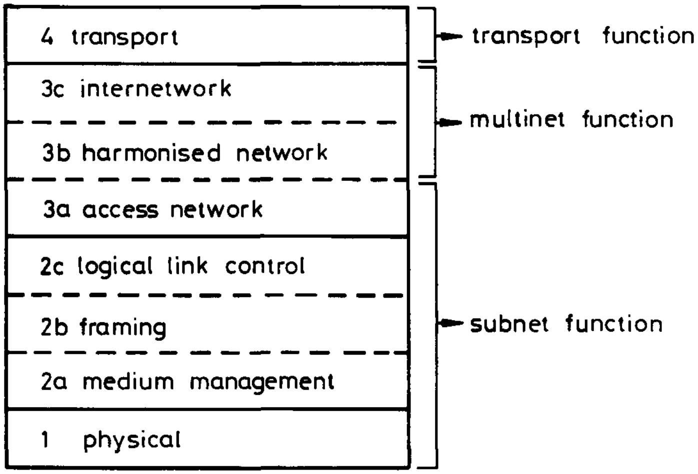


Fig. 3 **Telecommunications function sublayering**

2.4 *Layer Service Characteristics*

Layer services may be understood in terms of their *class* and *quality*. Two distinct classes of services may be identified: *connectionless* and *connectionlike*. A connectionless service offers a facility for the transmission of possibly unrelated messages, without a guarantee of delivery or correct sequencing. A connectionlike service offers a facility for the transmission of a series of messages, with a guarantee of delivery and correct sequencing. The names for the two classes of service stem from the fact that reliable delivery necessitates maintenance of state information from one message of a series to the next, thus requiring a logical or physical connection. Between the extremes of connectionless and connectionlike there exists a spectrum of intermediate possibilities.

The *quality of service* offered by a layer is specified in terms of parameters meaningful to the service user. Typical parameters are throughput, error rate and transit delay. The layer attempts to meet the quality of service requested by the service user, but may have to offer a poorer quality than that which was asked for. If an end-to-end connection is required, the remote service user is also involved in the decision as to what the quality of service should be. In the IPA telecommunications function, quality of service parameters are used to make decisions about multiplexing, choice of route, class of service or protocol needed etc.

3 Telecommunications function services

This section discusses the characteristics of the services offered by the lower four layers. Fig. 4 summarises the layer services for reference when reading the following text. The stability and modularity of the IPA telecommunications function stems from the careful design of these service boundaries.

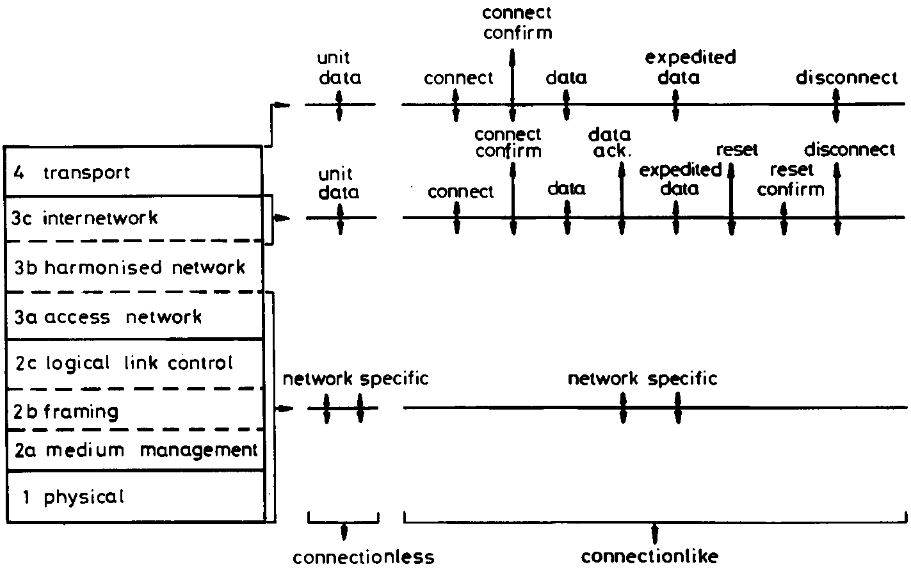


Fig. 4 Telecommunications function services

3.1 Transport Service

The reference model uses the term *end-system* to refer to the true ends of a communications path, as opposed to a *relay* which serves only to forward data and is not itself a source or sink of data. The transport layer is responsible for effecting data transfer between end-systems at minimum cost for the quality of service requested. This leads the transport layer to optimise the use of the network service and to bring the quality of service actually provided by the network layer up to the level required by the transport service user. The transport layer therefore contains its own multiplexing and enhancement functions. The characteristics of the transport service are that it is end-to-end, two-way simultaneous and transparent. IPA provides both connectionless and connectionlike transport services. The connectionless service does not guarantee delivery and is not flow-controlled end-to-end; it is intended for specialised management purposes. The connectionlike service guarantees delivery and is flow-controlled; it is intended for data traffic. The connectionlike service has connection establishment and termination facilities, and has normal and expedited data streams. Expedited data bypass normal flow-control procedures and are used for priority messages. The IPA transport service is the same as the current definition of the ISO transport service[8,9].

3.2 *Network Service*

The network layer is responsible for the transfer of data between end-systems, and so concerns itself with the interconnection of networks and routing between them. In IPA the network layer shares the responsibility for cost optimisation ot data traffic with the transport layer. Although both the transport service and network service operate between end-systems, it is the need for quality enhancement and cost minimisation that justifies the existence of a separate transport layer. The characteristics of the network service are that it is end-to-end, two-way simultaneous and transparent. Like the transport service, the network service offers connectionless and connectionlike facilities, but the connectionlike network service also has delivery confirmation and reset features. The connectionless service is the preferred IPA choice for data traffic. This brings benefits of simplicity and resilience. In the appropriate circumstances, for example if the underlying networks are high quality, the connectionlike service is used. The IPA network service is the same as the current definition of the ISO network service[10,11].

Within the network layer there are three sub-layers: inter-network (Internet, sub-layer *3c*), harmonised network (Harnet, sub-layer *3b*) and access network (Acnet, sub-layer *3a*). The Internet sub-layer concerns itself solely with the interpretation of global network addresses, with routing and with quality of service processing. The Internet sub-layer is presented with a uniform level of service by the Harnet sub-layer, irrespective of what the underlying networks are. It follows that the Internet sub-layer performs no service enhancement or de-enhancement of its own. The service supported by the Harnet sub-layer is therefore identical to the global Network service. The Harnet sub-layer, however, must operate over a variety of networks and must enhance or de-enhance these to bring them to a common level. The service supported by the Acnet sub-layer depends on the particular type of network.

3.3 *Data Link Service*

The data link layer is responsible for the transfer of data over a single communications link. The data link layer service is generally hidden inside the definition of a particular network, and is not usually explicity identified. Connectionless and connectionlike examples are both found.

Within the data link layer there are three sub-layers: logical link control (sub-layer *2c*), framing (sub-layer *2b*) and medium management (sub-layer *2a*). The logical link control sub-layer concerns itself with the transfer of messages over the communications link and with link addressing. The framing sub-layer looks after the assembly and disassembly of messages and performs transmission integrity checks. The medium management sub-layer is responsible for the control of physical medium signals. The data link sub-layers are too network-specific for any general statement about their service primitives to be meaningful.

3.4 *Physical Service*

The physical layer interfaces the physical medium and handles functions such as timing, modulation and signal levels. The physical layer services are even more network-specific than the data link services.

4 Telecommunications function protocols

This section discusses the characteristics of the protocols supported by the lower four layers. Fig. 5 summarises the layer protocols for reference when reading the following text. The Kernel IPA aspect of the telecommunications function is considerably simplified by a judicious selection of protocols.

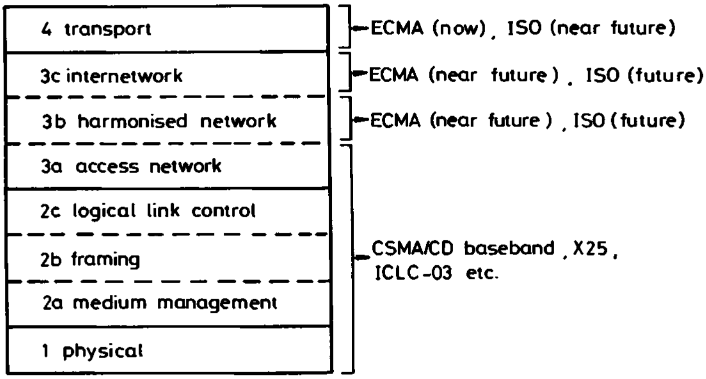


Fig. 5 Telecommunications function protocols

4.1 Transport Protocol

The transport protocol currently used in IPA is the one developed by ECMA[12]. ISO have evolved their own very similar transport protocol[13] from the ECMA one, and the two are expected to converge. Once the ISO definition has reached stability, and this is expected very soon, then ICL will formally adopt it.

The transport protocol consists of five protocol classes which offer differing degrees of enhancement. For a specified quality of service the transport layer can select the appropriate protocol class and options, dependent on the nature of the underlying network service. The five classes are:

0 – simple terminal class
1 – basic error recovery class
2 – multiplexing class
3 – error recovery class
4 – error detection and recovery class

Each class contains the functions of lower-numbered classes, except that class 1 contains error recovery functions which are present in class 3 but not in class 2. The functions of the various classes are briefly summarised below:

0 – transport connection establishment

– transfer of data
– segmentation of data (i.e. splitting up long messages)

1 – all class 0 functions
– transfer of data during the connection and disconnection phases
– transfer of expedited data
– sequence-checking
– re-transmission of data after failure

2 – all class 1 functions except data re-transmission
– flow control
– multiplexing

3 – all class 1 and 2 functions

4 – all class 3 functions
– error detection independent of the network service
– re-transmission on time-out
– re-sequencing (i.e. re-ordering messages received out of sequence)
– handling of corrupted or duplicated messages
– use of multiple network paths.

The ECMA and ISO transport protocols support a connectionlike service. Work in ISO is proceeding on a connectionless transport protocol, but is currently at an early stage of development. As an interim measure a connectionless extension to the transport protocol has therefore been defined for IPA purposes. The preferred protocol class for Kernel IPA is class 4, because this can be operated over practically any type of network service and hence over practically any type of network. Additional benefits of choosing class 4 are that the ultimate responsibility for reliability is placed in the end-systems, where it matters, and that concomitant simplifications in network interconnection are possible. Other protocol classes have specialised applications, for example class 0 for Teletex (the CCITT-developed replacement for Telex) and class 2 for communication over a high-quality network service.

4.2 Internet Protocol

Work on an Internet protocol to interconnect networks is proceeding apace in ECMA[14], whereas ISO are at an early stage of studying this issue. The current IPA target is therefore the ECMA protocol, but with the expectation of a move to the ISO Internet protocol once this is better defined:

ECMA have so far addressed only the provision of a connectionless network service. Future work on connectionlike operation is anticipated. The ECMA protocol carries Internet *datagrams*, the communications equivalent of telegrams. Three types of datagrams are defined:

•	Data (fixed format)	– header length – maximum permitted lifetime – protocol identifier and version number

– datagram type
– destination and source addresses
– user data

- Data (variable format)
  – all fixed format fields
  – segmentation control (i.e. splitting up long messages)
  – options concerned with routing, security, quality of service etc.

- Error
  – as for data but with the datagram type field changed to type error.

The two data formats are essentially the same, the fixed format being optimised for the local environment. The error datagram is used for reporting protocol failures. In the particular case of a single network, or more exactly a single network addressing domain, all header fields may be omitted leaving only a length indicator of zero.

4.3 Harnet Protocol

A Harnet protocol is in general required to harmonise the services available from differing networks. There are many possible such protocols because networks vary widely. As a simplification, ECMA are considering developing the transport protocol as the basis of a harmonisation protocol; only minor changes are needed to achieve this. Such a re-use of an existing protocol would not, of course, constitute another transport protocol because it would be fulfilling a different purpose.

4.4 Subnet Protocol

The Subnet protocol varies enormously from one type of network to another. A few examples of networks embraced by IPA are described in this section. There are many others of interest, for example:

- high bandwidth digital links
- ISDN (integrated services digital network, intended for digital data and voice)
- ring-type local area networks
- satellite links
- SNA (systems network architecture, from IBM)
- X.21 (digital switched networks).

4.4.1 CSMA/CD Baseband: ECMA has developed local area network (LAN) standards[15] based on the well-known Ethernet* technique. This style of network uses carrier-sense multiple-access (CSMA) techniques, with collision-detection (CD) mechanisms. Baseband transmission is used, i.e. the signal is not modulated on a carrier. An Ethernet-like network is built out of segments of coaxial cable

*Ethernet is a trademark of the Xerox Corporation connected to form an unrooted tree. The network operates at 10 Mbit/s on a free-for-all basis. Normally only one device at a time is transmitting on the network, but in the event of a simultaneous transmission attempt by two or more devices the collision is resolved by each station re-trying after a random delay. Another interesting property of this kind of network is that uses a broadcast medium, so that more than one device can receive the same message. A CSMA/CD network supports transparent data transfer and does not guarantee delivery, although it is generally reliable. It is inherently connectionless.

4.4.2 X.25: X.25[16,17] was developed by the International Telegraph and Telephone Consultative Committee (CCITT) as the means of interfacing to public packet-switched data networks.

X.25 can be viewed as a digital replacement for analogue telephone network capabilities. For example, it offers call set-up and clearing, call re-direction, permanent connections, reverse charging, accounting etc. X.25 supports reliable flow-controlled, two-way simultaneous, transparent data transfer. For X.25 just the access link to the network is required and calls through the network are statistically multiplexed over this. The X.25 equivalent of a telephone network connection is called a *virtual circuit* and shares many of the same properties. However, X.25 offers a richer variety of facilities, including normal and expedited data streams. An X.25 network is inherently connectionlike.

4.4.3 Full XBM: ICL's full extended basic mode (FXBM, also known as ICLC-03)[18] is supported by all current range ICL communications equipment. ICLC-03 is the embodiment of a layered architecture and can easily be separated into network-oriented and device-oriented functions[4]. It is therefore possible to think in terms of an ICLC-03 network which has the same status as any other network. Equally, it is possible to run the upper layers of ICLC-03 over any network, although in practice this will be done on top of the IPA data interchange function. This approach is vital in achieving a smooth transition from ICLC-03 to OSI. An ICLC-03 network is typically a multi-dropped communications link, i.e. a primary station and several secondary stations. The primary polls for input data and is responsible for scheduling output. In its network role, ICLC-03 supports reliable, flow-controlled, two-way alternate data transfer, and can optionally support transparency. ICLC-03 is inherently connectionlike.

5 Telecommunications function gateways

The IPA telecommunications function contains the most important watershed in the architecture: this is where the wide variety of network types is brought together to support a rich diversity of data processing facilities. The systems which interconnect networks are called *gateways*. This section discusses some broad categories of networks and shows the different ways in which IPA can bring them together.

5.1 Network Types

Networks may be broadly categorised as local area networks (LANs) or wide area networks (WANs). LANs are typified by their restricted geographic extent (a few kilometres), high bandwidth, short transit delay and private administration. WANs are typified by their considerable geographic extent (country-wide), low to medium bandwidth, lengthy transit delay and public administration. A CSMA/CD baseband network is a good example of a LAN, and an X.25 network is a good example of a WAN. Studies have recently begun of an intermediate type of network; the metropolitan area network (MAN), roughly city-sized.

5.2 Role of Local Area Networks

LANs may serve in three capacities[19]: as the means of distributing the components of a system, as networks in their own right, and as extensions of WANs. The three applications are not exclusive and the LAN may fulfil all of them at once. The IPA use of LANs and WANs is shown in Fig. 6: LANs serve to interconnect systems in the local environment, typically an office building or company site, and communicate across WANs. ECMA has recently given considerable impetus to LAN standardization through the publication of a LAN architecture[20].

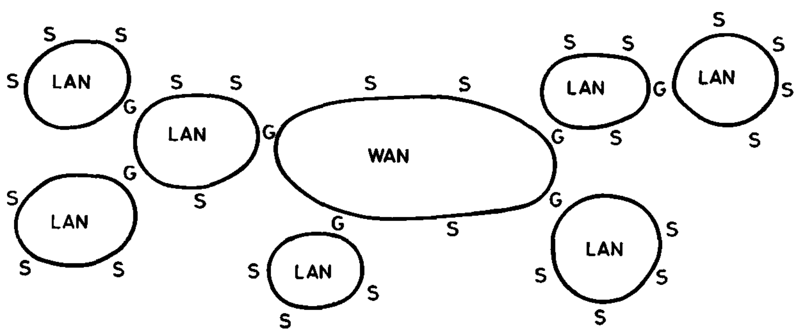


Fig. 6 IPA use of LANs and Wans (S=system, G=gateway)

5.3 Network Interconnection

There are three practical ways in which networks may be interconnected: at the transport layer, at the Internet sub-layer and directly at the Acnet sub-layer. The following sections discuss the three kinds of gateway implied by these network interconnection styles. The three gateway types support the three roles for a LAN mentioned above.

5.3.1 Distributed System Gateway: A distributed system gateway joins networks, or more precisely collections of networks, at the transport level. The name of this kind of gateway stems from the fact that it hides the distribution of a system across a network from the outside world. It has the characteristic of completely decoupling the networking domains thus connected, and is ideal in certain cases of LAN to WAN interconnection.

5.3.2 Internetwork Gateway: An Internetwork gateway joins networks at the Internet level. It is therefore the agent which binds networks to form the appearance of a single uniform network. If the network service is connectionless this binding is not tightly constrained, with the result that gateway design and control can be more relaxed. In the case of interconnecting a connectionless LAN with a connectionlike WAN, the gateway must enhance the LAN to have connectionlike features or must de-enhance the WAN to have connectionless features. An Internet gateway is therefore most appropriate for the connection of similar networks, for example LAN to LAN or WAN to WAN.

5.3.3 Network Extension Gateway: A network extension gateway joins networks on an intimate basis. One network then effectively becomes part of the other. This would be an attractive option if it were necessary, say, to extend X.25 into the private domain.

6 Conclusions

The IPA telecommunications function is at an advanced stage of definition and implementation. It is soundly based on OSI and draws on ICL's considerable communications experience from the past. There is no shortage of further challenging work, however. The incorporation of new digital services and networks, the amalgamation of digital voice, data and image traffic, the completion of addressing standards, and the development of a comprehensive management system are all important future goals.

Glossary of Abbreviations

The following abbreviations include those used in this paper and some used in work on OSI in general. Combinations of these are common, for example TSAP = T+SAP = transport service access point.

Acnet	access network (sub-layer *3a*)
C	connection
CCITT	International Telegraph and Telephone Consultative Committee
CSMA/CD	carrier-sense multiple-access with collision-detection
DCE	data circuit-terminating equipment
DTE	data terminal equipment
DL	data link (layer 2)
E	entity
ECMA	European Computer Manufacturers Association
Harnet	harmonised network (sub-layer *3b*)
Internet	inter-network (sub-layer *3c*)
ISO	International Organisation for Standardisation
L	link (layer 2)
LAN	local area network
MAN	metropolitan area network
N	network (layer 3)
OSI	open systems interconnection
Ph	physical (layer 1)
S	service
SAP	service access point
Subnet	sub-network
T	transport (layer 4)

TF	telecommunications function (layers 4-1)
WAN	wide area network

Acknowledgments

The architecture presented in this paper is the result of collaborative work in many standards bodies. Many people in ICL have helped to realise OSI standards in the shape of IPA and have influenced both for the better.

References

1 BRENNER, J.B.: 'Using open system interconnection standards', *ICL Tech. J.*, 1980 2(1), 106-116
2 HOULDSWORTH, J.: 'Standards for open network architecture', *ICL Tech.J.*, 1978, 1(1), 50-65
3 ISO: Draft International Standard 7498 (Information Processing Systems – Open Systems Interconnection – Basic Reference Model). International Organisation for Standardisation, Geneva, 1982
4 KEMP, J., and REYNOLDS, R.: 'The ICL information processing architecture (IPA), *ICL Tech.J.*, 1980, 2(2), 119-131
5 BRENNER, J.B.: 'Networking Architecture', *ICL Tech. J.*, 1983, 3 (3), 234-249
6 LLOYD, R.V.S.L.: 'IPA networking facilities', *ICL Tech. J.*, 1983, 3 (3), 250-264
7 GOSS, S.T.F.: 'IPA community management – concepts and principles', *ICL Tech.J.*, 1983, 3 (3), 278-288
8 ISO: Draft Proposal 8072 (Information Processing Systems – Open Systems Interconnection – Transport Service Specification). International Organisation for Standardisation, Geneva, 1982
9 ISO: TC97/SC16/N1163 (Working Draft for an Addendum to the Transport Service Definition covering Connectionless Data Transmission). International Organisation for Standardisation, Geneva, 1982
10 ISO: TC97/SC6/N2610 (Information Processing System – Open Systems Interconnection – Network Service Specification). International Organisation for Standardisation, Geneva, 1982
11 ISO: TC97/SC6/N2611 (Working Draft for an Addendum to the Network Service Definition covering Connectionless Data Transmission). International Organisation for Standardisation, Geneva, 1982
12 ECMA: Standard 72 (Transport Protocol). European Computer Manufacturers Association, Geneva, 1982
13 ISO: Draft Proposal 8073 (Information Processing Systems – Open Systems Interconnection – Transport Protocol Specification). International Organisation for Standardisation, Geneva, 1982
14 ECMA: TC24/82/198 (Working Paper on Connectionless Internet Protocol). European Computer Manufacturers Association, Geneva, 1982
15 ECMA: Standards 80, 81, 82 (Local Area Networks (CSMA/CD Baseband) – Coaxial Cable System, Physical Layer, Link Layer). European Computer Manufacturers Association, Geneva, 1982
16 CCITT: Recommendation X.25 (Interface between DTE and DCE for Terminals operating in the Packet Mode on Public Data Networks), Yellow Book Vol. VIII 2. International Telecommunications Union, Geneva, 1981
17 TURNER K.J.: 'Designing for the X.25 telecommunications standard', *ICL Tech. J.*, 1981 2(4), 340-364
18 ICL: RP3105 (Full XBM Protocol – ICLC-03). ICL, Reading, 1982
19 ECMA: Technical Report 13 (Network Layer Principles). European Computer Manufacturers Association, Geneva, 1982
20 ECMA: Technical Report 14 (Local Area Networks (Layers 1-4) – Architecture and Protocols). European Computer Manufacturers Association, Geneva, 1982

IPA community management

S.T.F. Goss

ICL Network Systems Division, Kidsgrove, Staffordshire

Abstract

IPA community management is a set of tools, facilities and methodologies which meets the needs of management functions in IPA networks. This paper is a general overview of the concepts and principles which motivate the design of community management. It also includes a brief survey of progress to date on international standards for OSI management.

1 Introduction

1.1 The need for community management

'Top down' descriptions of networking architectures such as IPA generally begin with a consideration of the requirements of the distributed applications which the network architecture is designed to support, and the end users who are served by these applications. From these requirements will be derived a statement of the service to be provided by the network architecture. Then the mechanisms which support this service will be specified. The result of this descriptive process is the definition of a networking environment in which end users, distributed applications and supporting products co-operate to satisfy requirements for distributed information processing.

For such a networking environment to function as required, many activities of preparation, installation and maintenance are necessary. A design for the network will need to be specified and checked to ensure that it has the necessary capacity and reliability. A plan for installing the equipment and software will need to be defined and carried out. These functions will be repeated from time to time as the information processing requirements of the network owner evolve. Also, during the normal running of the network, it will be necessary to monitor it to check that it continues to meet its objectives of throughput, availability, security and reliability, and to take corrective action in case of degradation or failure.

The above are examples of management functions. They are secondary requirements, in that the end users and distributed applications would be quite happy without them so long as the network environment continued to meet their needs for distributed information processing. When management functions are a practical necessity, end users and application designers tend to regard them as an overhead. If this overhead threatens to become too high it will be perceived as disruptive, and in such cases the overhead should be removed from the end users and applications to designated managers and management tools.

Community management is the part of the ICL information processing architecture (IPA) which provides management functions in IPA networks. It is a set of tools, facilities and methodologies for the execution of management functions, aiming for high productivity of managers and very low management overheads for end users and applications. Its design is part of the IPA architecture[1] and it is implemented in the products of the ICL Networked Product Line.

1.2 Managers and their roles

Having defined community management as tools, facilities and methodologies to enhance manager productivity, the logical next stop is to identify who are these managers and what are their goals and responsibilities. In practice this direct approach is thwarted by the wide variety of management styles associated with different types and sizes of network. In very large networks, shared by several independent enterprises covering a wide geographical area, we often find a team of staff dedicated to the management functions of system planning, budgeting and accounting, liaisons with equipment suppliers and PTTs, and so on. At the other extreme, where a few workstations are connected on one site by a simple and reliable local area network (LAN), many of these functions are not needed at all and the rest can be performed by end users with no perceived disruption. There will often be no computer expert in the house.

More helpful than identifying types of manager is to identify management roles. We find that all the wide variety of management jobs that exist can be described as combinations of these roles, of which there are less than a dozen. The roles, some of which are described in the next section, strongly suggest the sorts of aid which help to perform them.

1.3 Some management roles

The following paragraphs describe some of the most common and important management roles. The intention is to illustrate the way in which the roles are defined, and the list given here is not comprehensive.

1.3.1 Network design: In the simplest and most common cases, the design of a network is a straightforward task of selecting the most suitable standard configuration (such as LAN with office workstations), and specifying how many stations are required and where they are to be situated. Larger networks, involving critical elements such as internetwork gateways, public communications facilities and mainframes supporting multiple users, require considerably more care in their design. In the most complex cases it will be necessary to propose a trial configuration, check its throughput, reliability and security by means of analytical models and possibly simulation, and then improve the proposed configuration, iterating until an acceptable design is achieved.

1.3.2 Installation and modification: Once an acceptable network design has been achieved it is necessary to make a plan for installing it, and to execute this plan. For example, this involves liaison with suppliers of equipment and services, and arranging for contractors to visit the site and install cables and equipment.

The functions of network design and installation are not once-for-all tasks. As an enterprise evolves, so will its information processing needs, and this will lead to a need for occasional modifications to the network design and installation.

1.3.3 Resource control and accounting: Many types of network are designed to support the shared use of limited resources by several groups of users. The management of such networks may wish to control access to these resources, to set budgets for the amount of resource consumption and to account for actual consumption. This is not a universal requirement, however, and indeed these functions will often be completely absent in some common types of network, for example one-site office systems. IPA community management, therefore, provides support for this role as an option.

1.3.4 Network monitoring: The care that is taken to achieve a desired level of capacity, reliability and security in some network designs is wasted unless there is a way to be sure that these levels are achieved in operation. A common means of achieving this is a regular, say daily or weekly, analysis of statistics and event reports relating to the running of the network.

In addition to providing reassurance, such analysis can suggest ways to improve an underachieving network, or cut the cost of an overachieving one. So this function may generate input to reiterations of the network design process.

1.3.5 Fault-related roles: Faults develop in many types of network component and require rapid detection and evasive action, followed by complete repair or replacement of the faulty component. These requirements lead to the identification of a number of management roles: quick evasive action, expert diagnosis, maintenance, repair and replacement.

As a part of its Customer Services Strategy, ICL has developed a very comprehensive analysis of these roles and the technical approaches to them. The emphasis is on simplifying the tasks and avoiding unproductive travelling time by scarce experts This increases productivity by avoiding wastage of human resources, and by enabling much faster repair in most cases.

2 Fields of Management

A key aspect of IPA as a whole is the recognition that communications networking is only a means to an end; the goal is distributed applications. This goal also implies a further requirement: interworking of the systems which host the distributed applications. A separation of the three requirements, applications distribution, systems interworking and communications networking, leads to simplified methods of achieving each of the three.

Many currently available network management systems model a 'network' as a set of 'nodes' linked by communications facilities, and provide a set of facilities to control the nodes and links all together. There has been little distinction made between, say, running an echo test to locate a cable fault and loading application software into a 'node'. The facilities provided by such systems are limited by the inevitable growth in complexity as new functions and concepts are added.

IPA community management naturally reflects the modularity of IPA as a whole, and thus differs from the current norm in network management. In doing so, it gains the same benefits of separate and simplified requirements, leading to greater development potential and ultimately to the goal of increased user productivity.

The separation of telecommunications management, systems management and applications management forms a second dimension from that of the division of roles in Section 1.3 above. For example, the functions of telecommunications planning, system planning and applications planning, telecommunications installation, systems installation and applications installation all exist. In fact, all the management roles are applicable to each of the three levels of management separately.

2.1 Telecommunications management

In IPA telecommunications management there is a further major separation between subnetwork management, which manages each communications facility (subnetwork) individually, and interconnection management, which manages the layers that group the subnetworks into a complete Internet (layer *3c*) and use the Internet to provide higher-level interconnection (layer 4). This corresponds to the equivalent division within the IPA interconnection architecture itself[2].

2.1.1 Subnetwork management: This involves functions that manage the subnetwork mechanisms themselves and other functions which supervise the use of them by attached systems. The former are within the scope of community management only when the subnetwork mechanisms are defined by IPA, which is a minority of cases. The latter are always within the scope of community management.

For example, consider the UK packet-switching network, PSS. Management of the mechanisms of PSS, such as the internal links and switches and the interfacing equipment, is the responsibility of British Telecom as the proprietor for this network, and so this area of management is not covered in IPA. However, when ICL Networked Product Line equipment is attached to PSS, this equipment will manage its use of the network to achieve the best effect. For example, it will take care to exploit the capacity of already paid-for permanent virtual circuits before making extravagant use of switched virtual circuits.

The need for subnetwork management functions has to be considered specifically for each type of subnetwork. If the example chosen for the previous paragraph had been a LAN, there would have been no mention of switches and no hint of algorithms to optimise connection usage. There would instead have been reference to the requirements of installing and commissioning the cable and the need to avoid significant scheduling overheads in order to exploit the high data capacity of LANs.

2.1.2 Interconnection management: This is concerned with the more global aspects of communications. One of its major aspects is the management of routes: deciding on the criteria for routing between subnetworks (the planning function), establishing and maintaining the appropriate routing tables (installation) and the monitoring of these routes for reliability and use to capacity. Interconnection management in general is independent of the details of specific subnetwork types, although it refers to some attributes of subnetworks such as capacity, circuit costs and security, in order to make best use of them.

2.2 System management

A principle of IPA community management is that each interconnected system has both the authority and the responsibility to manage its own resources and internal working. Therefore much of the management of systems interworking in IPA is defined in terms of co-operation between autonomous systems which provide services to each other but do not ultimately control each other.

A basic system management requirement is to establish in each co-operating system any knowledge it needs of the other systems with which it will interact. This is especially true of systems which are closely linked by a LAN, where the high data capacity gives an opportunity for the more powerful systems on the LAN to provide management services to the simpler 'servers'. For example, the larger systems can take responsibility for holding the network-wide address directories, for loading and dumping the control programs of the smaller systems and for collecting error logs and statistics which would be beyond the capacity of small systems.

2.3 Applications management

The effective working of distributed applications requires care in the placement of the component programs, and especially of the data involved. The use of multiple copies of data to allow quick access has to be balanced with the need to ensure integrity of the whole. Access permissions have to be defined and formalised. These requirements imply facilities to help plan and establish data placement, to control access according to privilege and monitor it for effeciency, and to effect relocation when needed.

3 Design and product concepts

3.1 Community self-management

Section 1.3 intentionally omitted operations from the list of management roles. It is certainly true that some functions have to be performed by people in attendance, especially physical actions such as loading new media on printers and magnetic devices. However, it is ICL strategy in general to simplify such functions to the extent that they can be done by end users without extensive training and without annoyance, in the same way that users of word processors load their own floppy discs and printer stationery.

Community management follows the principle that the scheduling criteria and similar operational criteria for a network should be decided at the planning stage. The criteria should be formally expressed to the systems involved at the time of installation/modification, and the systems should then be responsible for obeying these criteria. The involvement of skilled operators in making what should be automatic decisions is a wasted expense and a restriction on the geographical location of the networked systems. Community self-management thus contributes to raising productivity, and by giving faster and more automatic response to failure also increases network resilience.

3.2 Product variety, standards and transparency

There is considerable variety in the products of the ICL Networked Product Line, reflecting the variety of market needs which these products are designed to meet. To make this diverse product set into a Networked Product Line, therefore, clearly implies a need for standards. At the same time, these standards must give room for each product to achieve excellence in its own individual sphere.

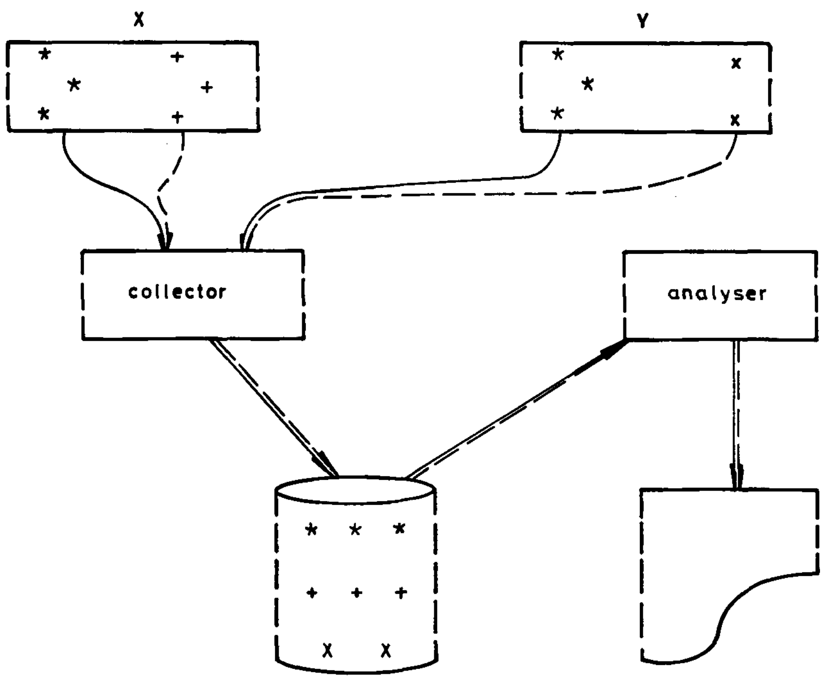


Fig. 1 Design for a statistics analysis facility

The principle adopted is that the standards should provide the transparency needed to allow specialised solutions to specialised problems. This is not the same as defining many options, leading to an '*N* squared problem' of option combinations. The standard and transparent fields of standard messages are clearly distinguished. The standard parts allow unlike products to combine effectively and simply. The transparency allows special added value when like combines with like.

3.3 System monitoring facility

As an example of the application of transparency in standards, this Section describes a statistics collection and analysis facility which applies the principle stated in Section 3.2. Let us imagine a network in which a variety of applications and systems are geographically dispersed. The monitoring role of management is performed at one of the sites, at which statistics for the network are collected and analysed once a week, say.

Fig. 1 illustrates how this works. Two of the networked systems, X and Y, are shown. Each of these collects IPA standard statistics (shown with an asterisk), and product specific statistics (+ and x). These are transmitted from time to time to a collector program by means of a protocol that represents the standard statistics in a standard fashion (unbroken line) and specialised statistics transparently (broken line). All the statistics are stored in a file which is analysed weekly by the analyser program that produces the final printed output.

3.4 Online monitoring and control

Another networking construct arises when a person at a terminal requires to interact with the networked systems in real time. For reasons stated in Section 3.1, community management seeks to avoid requiring users to do this, but it is occasionally necessary for advanced fault diagnosis and users may desire to do it for other reasons of their own. Fig. 2 shows the general case, in which a terminal connected to one system (A) may be used to interact with others (B and C).

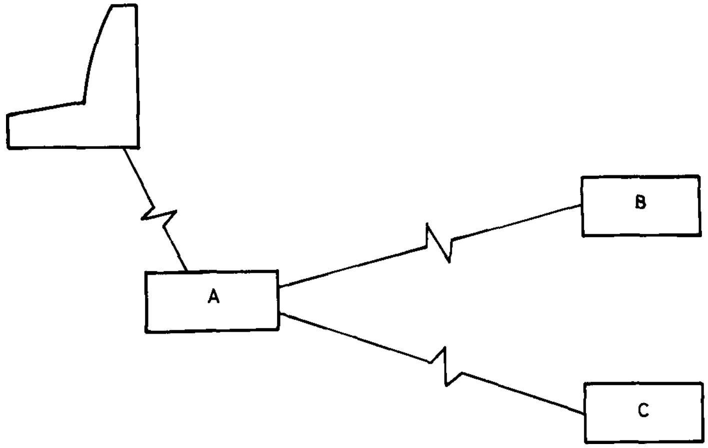


Fig. 2 Remote operating general configuration

The concept of Fig. 2 can be applied in two distinct ways. First, the user may wish to use a service provided by one or other of the remote systems B and C. The role of system A is to provide a 'service selection service' and then to relay the dialogue between user and selected service transparently. This is, in fact, identical to the RSA facility of IPA Phase 2[3]. Community management makes direct use of RSA.

The second application applies to users wishing to have a wider view of, and interaction with, several systems in the network simultaneously. RSA is inappropriate for this, because it would be necessary to cycle rapidly, selecting services at each system with which the user is concerned. In this case, system A is required to multiplex dialogues with the other systems on to the one terminal, in a more dynamic manner.

3.5 Topography map

Any application or system involved in a network needs to have available to it a representation of the network environment in which it must function. Managers, too, need such a representation, from the network design stage onwards. We refer to such a representation of a network as a topography map. It contains information about the software, hardware and communications subnetworks involved in the network, and the linkages between them.

Such a requirement exists in a more limited sense even in stand-alone system environments, in that any system must have a representation of the 'object space' on which it operates. Therefore all ICL operating system and application system products already have such a representation in the form of a catalogue, dictionary, configuration table etc. These are building blocks for the topography map.

The product-specific building blocks are exploited in a way already hinted at in Section 2.2. Each system in a network is deemed to hold the master copy of all information about the immediate environment which it controls. This information is made available, by means of a directory management protocol, to other systems, applications and users who need to know it. In this way, a complete picture can be accumulated at one place if needed, and secondary copies of the information can be made and updated to allow the most efficient access.

3.6 Aids to the network design and planning

The design of a network and the planning of its installation and subsequent modifications can, in the more complex cases, involve considerable intellectual difficulties. The following is a brief list of the types of aid that will be provided to help overcome such difficulties:

- *Analytical models,* designed to predict the throughput and reliability, for example, of trial configurations for complex networks. Such models can be purely documentary in very simple cases but are often appropriate for automation as computer programs.
- *Simulation tools*, to assist with the special cases not covered by the more

general analytical models.

– *Ticketing applications*: these assume that action plans, for example for network installation, can be represented as sequences of steps, and each step represented by a 'ticket' which describes a required action. This method of interfacing between planning and execution is implemented as a computer application and this approach gives scope to evolve tools to automate some of the execution and monitoring of plans.

4 Progress towards OSI management standards

4.1 Summary of history and current activities

Early in the process of defining the requirements for OSI standards it was recognised that standards would be needed for management protocols in environments that use an OSI. Therefore ISO/TC97/SC16, the international body responsible for defining OSI standards, formed a working group to work on OSI Management. This group, SC16/WG4, first met in April 1980.

The early work of SC16/WG4 concentrated on identifying and prioritising the requirements for OSI management functions, and defining an OSI management framework[4]. The framework is described in more detail in the next section.

More recently, SC16/WG4 has defined a programme of work[5] directed at the definition of standards in selected subject areas. The chosen areas are:

– accounting, error reporting and capability management[6]
– commitment, concurrency and recovery[7]
– control of groups of application processes[8].

The references, which are current as this paper is written in January 1983, give information on what is meant by these terms, and the approach being taken.

The major contributions to the work of SC16/WG4 to date have been the national standards bodies, especially of the UK, Japan, the USA and Germany. A recent newcomer to this activity is the European Computer Manufacturers Association (ECMA) which formed a task group on OSI management at the November 1982 meeting of TC23, the ECMA committee on OSI. The OSI management task group of TC23 has made initial studies in the fields of accounting, directory management and application layer access control.

Study of management has also begun recently in the Institute of Electrical & Electronic Engineers (IEEE) under Project 802, which is concerned with LAN standards. The concepts developed by this group may be of particular significance, as they complement the model of protocol inter-relationships in Reference 6 with a model of the configuration of the OSI environment in which these protocols may be used. An outline is given in Section 4.3 below.

4.2 *The OSI management model*

Reference 4 is the ISO working paper which defines the OSI management model. It consolidates the early work done to identify the requirements for OSI management and the management services to be provided to meet these requirements. It also outlines the fundamental technical concepts to be used in relating OSI management standards to one another.

A basic technical concept is the distinction[4] between application management, systems management and layer management. Application management contains those functions which control the working of a single distributed application. Systems management controls the sharing of resources among several distributed applications. Layer management exists for each of the seven layers of OSI, partly within the layer itself to control that layer and partly within the scope of systems management to control the wider effects of operations in the layer.

Finally, Reference 4 contains a detailed model of all types of discourse that may be invoked in the course of OSI management. For reasons of completeness, it identifies all types of discourse and not only those which will be subject to OSI management standards. Those which are subject to such standards are the following:

- system management protocols
- application management protocols
- layer management protocols

4.3 *The IEEE 'domain' concept*

The IEEE study referred to in Section 4.1 took the view that OSI environments would be organised into hierarchic domains for management purposes. Each domain would contain one primary, or master, and one or more secondaries that are, at least by some extent, controlled by the primary. The model is hierarchic because the primary in one domain can also be a secondary in a larger domain.

Fig. 3 illustrates such a hierarchy, *P* denoting a primary, *S* a secondary and *S/P* an element that is primary in one domain and secondary in a higher domain. This example has three levels: domain A at the top level, B at the second level and C, D and E at the third.

A configuration like that in Fig. 3 exists at one point in time, and the possibility of reconfiguration is recognised: end systems may exchange roles, and they may be moved from one domain to another.

4.4 *Intercept of OSI management standards*

IPA community management is planned to conform to international standards for OSI management as these become available. Because they are not yet available, community management is following the 'intercept' approach already established for IPA as a whole[1].

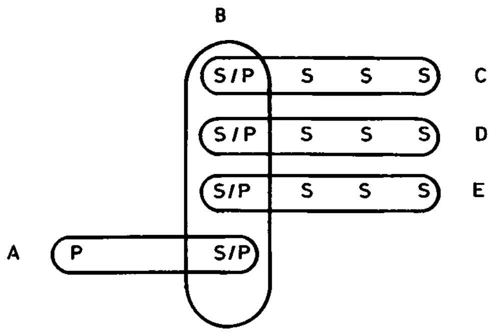


Fig. 3 Hierarchic management domains

The intercept approach involves contributing to rapid progress towards international standards by making substantial technical contributions to the standardisation bodies. ICL staff are contributing actively to the work in ECMA, the British Standards Institution, the IEEE and, through these, to ISO itself.

The intercept approach also involves technical measures which prepare the way for the adoption of future standards. The architectural work done in ISO gives some good indications of which management functions fall within the scope of which protocol standards, and how each protocol will relate to other OSI protocols. The earliest community management standards, although unavoidably proprietary to ICL in their detailed design, already conform to the indicated architectures of OSI management standards.

5 Conclusion

This paper has given a brief overview of the concepts and principles which motivate the design of IPA community management. It aims to give a lead-in to a more detailed description of specific community management topics and products which will be given in future papers in this journal.

Acknowledgments

Many people from several divisions in ICL, have contributed to the concepts described in this paper, and are continuing to contribute to their realisation. Without this widespread collaboration, this paper would not have been possible.

References

1 BRENNER J.B.: 'IPA networking architecture', *ICL Tech. J.*, 1983, 3 (3), 234-249
2 TURNER, K.: 'The IPA telecommunications function, *ICL Tech. J.*, 1983,3 (3), 265-277
3 KEMP, J., and REYNOLDS, R.: 'The ICL information processing architecture IPA', *ICL Tech. J.*, 1980, 2, 119-131.
4 ISO/TC97/SC16 N1217 'OSI management framework (third working draft)',
5 ISO/TC97/SC16 N1227 'Programme of work for Working Group 4',
6 ISO/TC97/SC16 N1214 'Accounting, error reporting and capability management',
7 ISO/TC97/SC16 N1215 'Commitment, concurrency and recovery',
8 ISO/TC97/SC16 'Control of groups of application processes',

MACROLAN: A high-performance network

R.W. Stevens
ICL Mainframe Systems Development Division, Manchester

Abstract

The principles and applications of local area networks (LANs) are now well publicised. They offer substantial advantages over conventional point-to-point links in a computer environment, since they impose little constraint on system configuration. Peripherals and processing nodes can be distributed on the network in any manner physically convenient to the user. The availability of complete cross-communication between stations permits distributed processing and shared access to storage and input/output media. In certain areas, however, conventional LANs have insufficient performance to cope with the traffic rate. Coupling a processor to main store via a LAN would be an absurdity; this clearly requires the use of a direct point-to-point interface. In some areas LAN flexibility is required, but at a performance level more typical of a dedicated link. This paper describes the implementation of a network which fulfils this requirement: MACROLAN. The transmission medium adopted is optical fibre and is thus a new technology serving a new application. The physical aspects of this network are therefore emphasised in this paper.

1 Introduction

Magnetic disc storage devices are now becoming available with performances in the range 3-5 Mbytes/s. Such devices clearly require interconnections of compatible bandwidths if available data rates are to be maintained. Distributed mainframe processors require tight coupling to each other but still need the flexibility of siting and reconfiguration offered by a local area network (LAN). General purpose LANs used for the interconnection of very large numbers of microprocessor-based terminals, and which must serve devices as diverse as giant mainframe OCPs and line printers, are less appropriate to the needs of a high-performance distributed system. MACROLAN has been designed to fulfil the performance objectives while retaining the LAN philosophy.

2 Choice of network type

There have been many network types proposed in the last few years, each having features appropriate to certain applications. These types may be simply categorised by their method of resolution of contention (two or more stations attempting to transmit simultaneously) and by their physical configuration. The latter is basically a choice between rings and buses. Methods of contention resolution include:

- token passing
- empty-slot technique
- carrier-sensed multiple access (CSMA)
- polling
- time multiplexing.

A comparison of the relative merits of these networks is beyond the scope of this paper; however, the following briefly outlines the main choices.

The scheme most widely adopted is the CSMA bus, typified by Ethernet[1]. The arrangement is highly resilient to station malfunction, and the physical access to the LAN is excellent. The best known ring configuration is an empty-slot type developed by Cambridge University and known widely as the Cambridge Ring[2]. This has less resilience than the CSMA bus but the protocol does allow guaranteed access and is arguably more suitable for voice and facsimile in addition to DP traffic. Excellent though these networks are for most applications, they do not offer sufficient performance in terms of delay throughput for the application required. A comparison of the performances of various LAN types has been made by Werner Bux[3].

Neither the empty-slot technique, with its limitation on packet sizes, nor CSMA, which introduces delays in order to resolve contention, compares with other options if performance is used as the sole criterion. Token passing offers the simplest scheme if a ring topology is acceptable (Fig. 1).

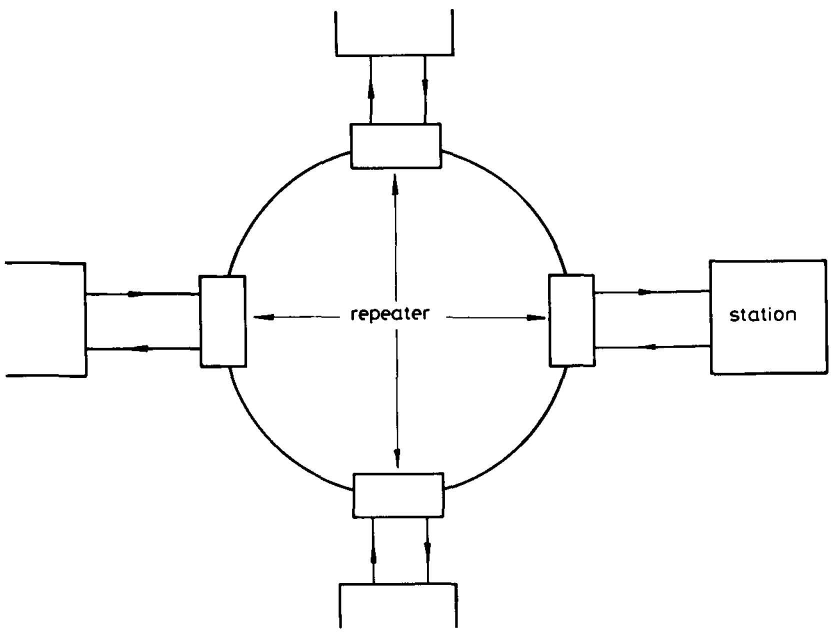


Fig. 1 Token passing ring

In such a network a number of communicating units are arranged in a ring around which a single token circulates. At any time when no message is being sent the ring will contain in its overall time delay a series of bits corresponding to the token and a few odd bits which were at the end of the last message sent. No unit is permitted to transmit unless it is in possession of the token. A transmitting unit will remove the token from the ring and insert a message, appending a new token upon completion. In this manner all units have the same chance to transmit. When there is a high demand for opportunity to transmit then units will, in effect, form an orderly queue, each awaiting its turn. Since, in general, messages are long compared with the ring length the efficiency of the ring is very high.

The main objection to this type of ring is its lack of resilience. Attached to each station must be a repeater in series with the transmission medium. The repeaters must all be powered whether the attached stations are operable or not; failure of any repeater leads to total network collapse. This drawback can be reduced considerably by centralising the repeater logic; this does not incur any significant hardware overhead[4]. By making the central (port switch) unit 'poll' the end units, rather than a single token being passed from station to station, the ring concept is lost (Figs. 2 and 3). This latter method has been adopted for MACROLAN.

By cascading port switching units in the manner shown in Fig. 4 the network may be extended indefinitely and may be considered to be a bus to which further 'taps' may be added.

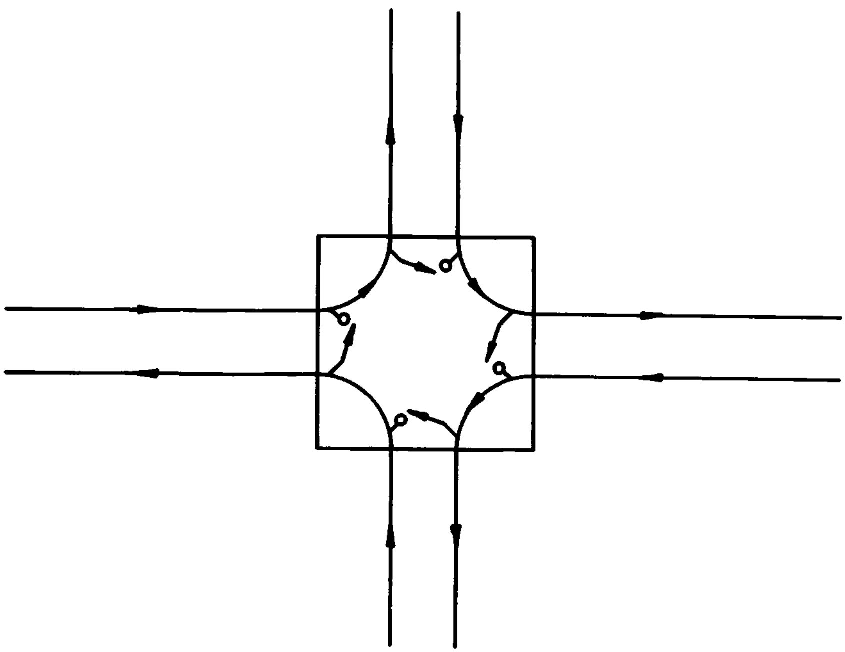


Fig. 2 Port switch: ring

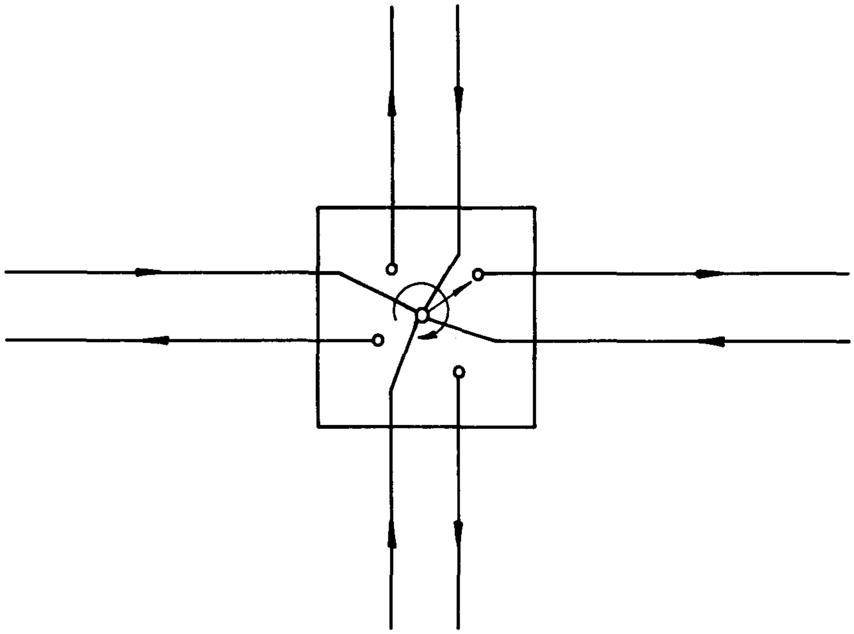


Fig. 3 Port switch: polling

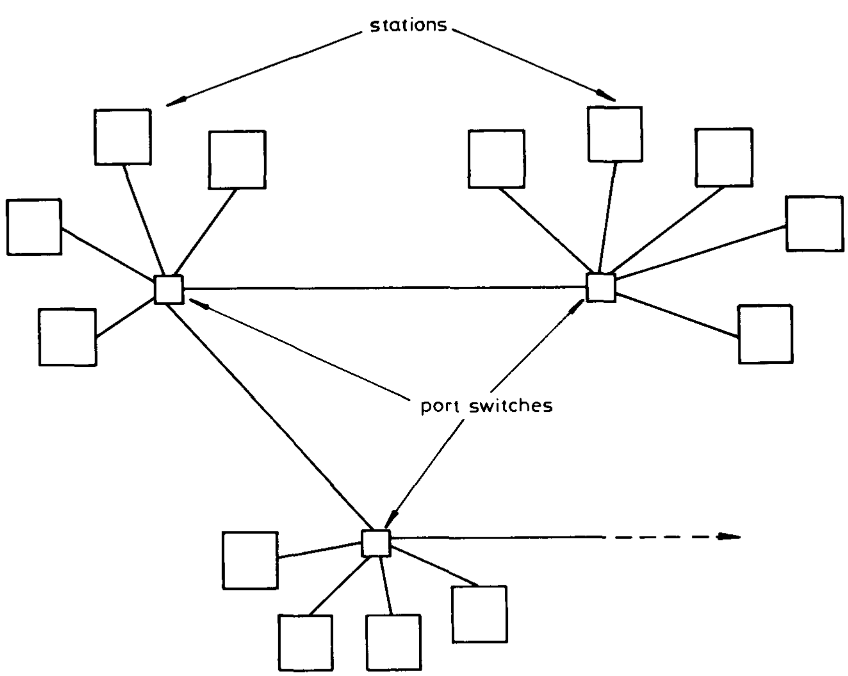


Fig. 4 Cascaded port switches

3 Choice of transmission medium

Copper cables still remain the most cost effective technology with which to implement low-to-medium performance interconnections. Lack of cheap optical components has allowed copper to maintain an otherwise unjustified dominance in short-haul data transmission. The advantages of optical fibres are very significant: freedom from electrical interference, no generated radio frequency interference, low bulk and weight, eminent safety and excellent data security are a few of the features which have invoked their use to resolve specific problems. The cost equation is, however, changing rapidly. Fibre costs are in many cases now competitive with copper; transducer vendors have gone through evolutionary stages from simple diode devices, to which complex circuitry must be added in order to produce a transmission link, to sophisticated transmitter and receiver devices incorporating onchip encoding and decoding. Such devices are exemplified by the Plessey HRCL series[5]. By using such devices at 50 Mbits/s it is possible to gain all the advantages of optical fibres with no significant extra cost compared with copper. At this data rate the bandwidth requirements of a purely electrical interface demand very high quality cables and components.

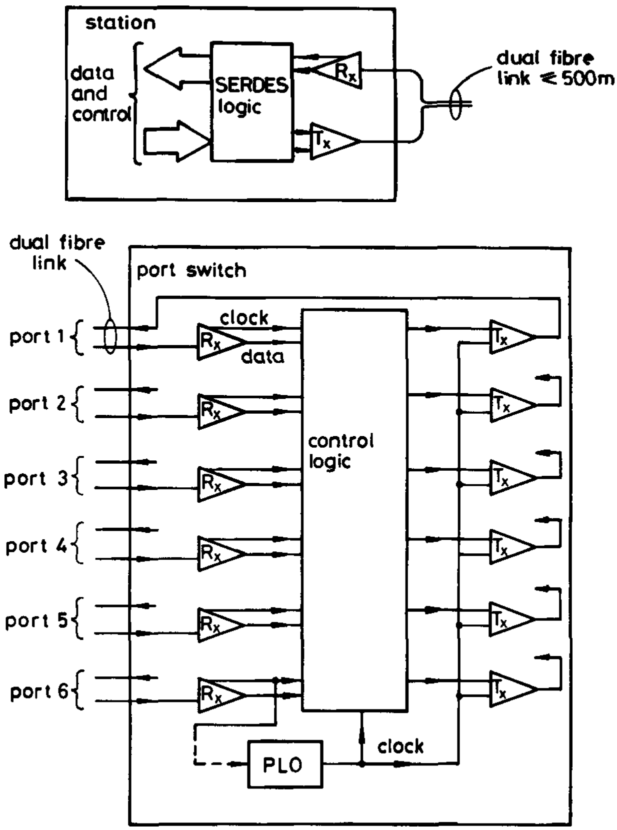


Fig. 5 Simplified schematic
SERDES: serialisation and deserialisation
PLO : phase-locked oscillator

R_x receiver
T_x transmitter

50 Mbits/s may seem pedestrian in comparison with the performance of some telecommunications links. It does, however, effectively mark the threshold beyond which high-cost laser technology is required. Furthermore, as the data rate increases beyond this point, it becomes difficult to perform some protocol functions at the bit serial level. Although it is possible to perform such functions at a higher level, an overall loss in performance results due to increased station delay.

4 MACROLAN: Operation

The following is a brief description of the network operation (Fig. 5).

The overall network comprises optical fibre transmission links which connect communicating units (stations) together via one or more distribution units (port switches). This provides a network having the topology of a star or interconnected stars, each port switch serving a local star (cluster of stations).

Each station is connected to its associated port switch via two optical fibres. One fibre is used for transmission and one for reception. The optical transmitter and receiver within each station interface with local serialisation and deserialisation logic (SERDES). The port switch(es) continually monitor the activity of attached stations, switching out those stations which are inoperable (i.e. those providing invalid optical output).

Activity is initiated by a station sending a go ahead (GA) character to its attached port switch. Any station may do this; there is no master unit. The port switch responds by sending a similar character to the next operable link; should this be attached to a further port switch then this behaves in a similar manner. On receiving a GA a station that wishes to transmit changes it to a start of frame (SOF) character and then transmits its message. A station not wishing to transmit simply returns any GA it receives to the port switch.

On detection of SOF the port switch changes to a different mode. Instead of circulating the GA as before, it broadcasts the SOF and the associated message simultaneously from each port. On completion of the message the port switch takes no further action but simply waits for a new GA to be generated. This will normally be performed by the transmitting station, and the polling operation then continues as before.

Certain message types require an acknowledgment, and this requirement can be costly in elapsed time and therefore in performance. A major feature of MACROLAN is that a facility exists to allow an acknowledgment to be performed at a low level. This is an important advantage in a distributed processing environment. Any station may qualify a message as requiring an acknowledgment; a special field outside the message frame is allocated to this purpose. Stations receiving such a message successfully pass an acknowledgment (AK) character back to their attached port switch. The port switch, having received AKs from each station or port switch, then sends a further AK character to the transmitter. No action above link level is required to complete this operation. Should no AK be received by the transmitter then control may be passed to a higher protocol level and error management routines may need to be invoked.

5 Physical implementation

To construct a complex, high-performance network such as MACROLAN with the additional objectives of low cost and high reliability much dependence is placed on modern LSI technology. Indeed, without such technology it is doubtful that such a network would be practicable.

Plessey HRCL series data links have been adopted. These operate from a single 5 V supply and are packaged in familiar integrated circuit style packages with the addition of an optical fibre pigtail. These devices have integral Manchester biphase encode/decode functions providing an electrical interface comprising separate clock and data I/O. The transmitter consists of a high-radiance etched-well light-emitting diode driven by an amplifier having a range of selectable drive currents. The receiver has a silicon PIN photodiode detector providing the input to an amplifier and decoder fabricated on a single chip. Optical fibre for the link is a step index type having a nominal internal diameter of 140 μm. Such fibre is adequate for the application and permits the use of low-cost, and more importantly, field-terminatable connectors. Graded-index fibre, in general, being of smaller diameter, requires more sophisticated termination techniques. Such fibre is more suitable for longer haul transmissions than the application generally demands: the fibre chosen is adequate for links up to 500 m in length.

The port switch unit contains six transmitter/receiver pairs and may hence serve a cluster of six stations or further port switches. Port-switching logic has approximately 1200 gates which are provided by three identical ECL ULAs. This technology is chosen since all logical operations of the port switch take place at 50 Mbits/s: a deserialiser function introduced at this point would introduce intolerable network delay times. The ECL ULAs chosen can cope comfortably at this rate. The unit is housed in a skirting or wall mounting enclosure approximately 15 x 9 x 3·7 in; it has its own ac mains supply and is thus fully independent of any attached stations. The port switch also provides the network clock and a clock synchronisation function. The latter is performed by a phase-locked crystal oscillator; this effectively prevents accumulation of clock jitter when many port switches are cascaded.

At each station, in addition to the optical devices, a 2000-gate ULA is available to perform SERDES functions. This device carries out most of the link level functions required by the network; in particular it is used to serialise and deserialise data from the eight bit I/O interface to its host. The device contains the necessary logic to perform bit stuffing and stripping, framing, frame integrity and error detection, acknowledgment detection and generation. Error detection is provided by a 16-bit cyclic redundancy check; the integrity of the fibre optical devices is already high because they are free from errors due to environmental noise. SERDES is TTL compatible and is therefore optimised for TTL and CMOS station logic. To minimise the power supply requirement within the host all MACROLAN components operate from a single 5 V rail. A special IC is provided to couple SERDES to the optical devices, which are ECL compatible. At this level the 50 Mbit/s data stream is split into two 25 Mbits/s channels. This is necessary to obviate the problem of skew between the clock and data signals which occupy separate TTL lines at this point.

All the station components associated with MACROLAN up to and including the medium access level can be easily accommodated on a single PCB of approximately 7 x 4 in.

6 Conclusion

By using LSI and state-of-the-art optical devices an interconnect which provides a performance comparable with existing dedicated parallel interfaces but offering the advantages of networking can be produced. This is achieved with a longer haul capability and with electromagnetic compatibility characteristics far superior to those obtained with copper cables.

This facility provides the ability to couple processors sufficiently tightly to produce an efficient distributed system. Furthermore, it offers a solution to the problems of cable bulk and distribution associated with the interconnect of high-speed peripherals, particularly in some installations where large filestore requirements exist.

Optical fibre technology will continue to improve and become more cost effective; LANs of all performance levels will gradually migrate to fibre. Even higher performance levels will become feasible as the problems of cost and complexity are resolved.

References

1 'The Ethernet, a local area network, data link layer and physical layer specifications', Digital, Intel & Xerox, 1980
2 HOPPER, A.: 'A local area computer communication network' Ph.D thesis, Computer Laboratory, University of Cambridge, April 1978
3 BUX, W.: 'Local area subnetworks: a performance comparison', IBM Zurich Research Laboratory, Switzerland
4 SALTZER, P.: 'A star shaped ring network with high integrity', (Proceedings of LACN Symposium, May 1979, Massechusetts Institute of Technology.
5 SUMERLING, G.W. and MORGAN, P.J.: 'A low cost transparent fibre optic link for inter rack communication', Plessey Research (Caswell)

Specification in CSP language of the ECMA-72 Class 4 transport protocol

A.S. Mapstone
Consultancy & Training Services Division, ICL Reading, Berkshire

Abstract

The paper gives a formal specification of the data-transfer phase of the ECMA-72 Class 4 transport protocol, written in the CSP language devised by Prof. C.A.R. Hoare of Oxford University. It also discusses the merits of using this kind of formal specification technique to define large-scale protocol standards.

1 Introduction

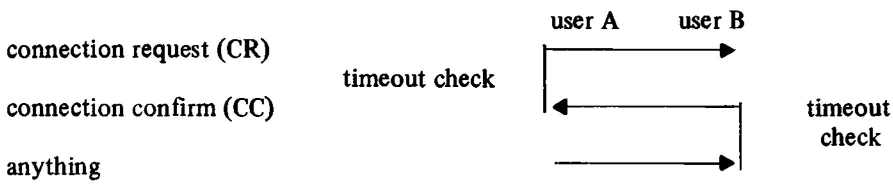
A transport protocol in a communications system is the procedural means whereby data are transmitted end-to-end between systems in a network-independent way. More specifically, it is a process that provides a transparent, reliable, sequenced, flow-controlled communication service between two users. The ECMA-72 transport protocol[1], which in addition provides full duplex service, meaning that transmission can proceed independently in the two directions, is one of a set of international standards which is being actively promoted by several computer manufacturers. Class 4 is the version that is capable of operating over any network, for example over X-25, Ethernet or a satellite network; it is best over datagram networks which may corrupt, lose or mis-sequence message packets.

The application of formal specification languages to this protocol is of interest to both the academic and the industrial communities. The existing English language specification provides a source of realistic problems on which to evaluate formal languages; the work reported here is the formal specification of a part of this protocol in the CSP language. As the specification of the whole protocol would be a large task, the paper covers only the data transfer phase, allowing credits greater than 1, allowing the receiver to renege on credit values and incorporating an inactivity timer – these terms are explained later in the paper. This phase is constructed out of two 'half transfers'. No expedited data service is included and the connection phase is reduced to an 'initialise system' operation. A network which loses messages occasionally is assumed. All options in the protocol apart from credit size and the number of transport connections are ignored or are assumed to have fixed values.

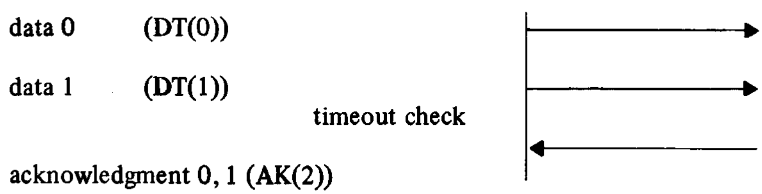
The specification produced here differs in a few points of detail from what is currently stated by ECMA-72. These are indicated at the appropriate places in the text by references to notes in the Appendix.

2 ECMA-72 Class 4 transport protocol

Two transport users exchange reference numbers during a connection phase initiated by one of them; these numbers then distinguish this connection from all others made by either of the users. Each user has one globally unique transport service access point identifier (TSAP-ID), or address:



The subsequent data transfer phase can be considered as two symmetrical and independent transfers, one in each direction; the explanation here is in terms of one of these half transfers:



(NB. sequence numbers start from 0)

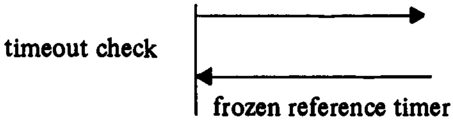
Each data transfer message (DT) is sequentially numbered to enable the receiver to recover the proper message order, and each message's associated acknowledgment (AK) may be timed out, causing retransmission of the message. The number of messages, if any, that may be outstanding and unacknowledged at one time is called the *credit*; the initial value of this is agreed at connection time and the current value indicated in each acknowledgment, allowing the receiver to control the data flow. The acknowledgment also gives the sequence number of the next message which should be received by the receiver.

The protocol has a 'last' bit (EOT) which allows the data in limited size packets to be concatenated, so that the transfer appears to the user to be of unlimited size. The maximum size of a data packet is agreed at connection time. The sequence numbers used for the messages must not 'wrap round' in a period less than the maximum lifetime of any packet in the network, which is specified as a network service constant.

The normal data transfer can be blocked for long periods if the credit is set to zero, so an expedited transfer mechanism is provided as a limited bypass. This works in the same way as the normal transfer except that 'ED' and 'EA' are used instead of 'DT' and 'AK', with a separate sequence number, the credit is fixed at 1, there is only limited space for data and there is no 'last' bit EOT. This means that there is no explicit flow control by means of credit and there can never be more than one expedited message outstanding in either direction at any one time. The normal data transfer (DT) is suspended while an expedited transfer is in use.

A disconnection may be initiated by either user. It will always occur if either end goes quiet for too long; to prevent this, if no user data has been transferred for some time the ends exchange acknowledgments which are copies of the previous ones to keep the connection 'warm'. This is called the *inactivity timer*:

disconnection request	(DR)
disconnection confirm	(DC)



The reference numbers used during the connection cannot be reused immediately. There is a 'frozen reference' interval which is longer than the maximum lifetime, including any retries, of any message packet in the network. Any messages arriving during this period are discarded.

3 Communicating sequential processes: the CSP language

The protocol is concerned with the communication between processes along channels. A *named channel* is defined as a channel of communication connecting two and only two processes, and a process is regarded as a potential component of a network of processes connected by named channels. The notation used in this paper to specify the behaviour of the processes is a subset of that used in the formal mathematical model for communicating sequential processes defined by Hoare and his co-workers.[2-4] Not all of the symbols and operators defined there are used here because not all are needed; those which are used are listed and explained below.

The basic concepts of CSP are the symbol, the alphabet, the trace and the process.

A *symbol* may be intuitively understood as denoting a class of event in which a process may participate; there are these definitions:

$a!x$ means 'output the value of x on the named channel a'

Thus if there are channels conventionally labelled 'right' and 'left', then 'right! x' denotes the output of x on the 'right' channel.

$a?x$ means 'input the value of x from the named channel a'

As a channel is assumed to have only two ends, $a!x$ is equivalent to outputting x to the process 'owning' the other end of the channel a.

The *alphabet* of a process is the set of all symbols denoting the events in which the process can participate; thus for the 'timer' process described in Section 4.2 the alphabet is

{set?time, reset!timeout}

A *trace* is a finite sequence of symbols recording the actual or possible behaviour of a process from its beginning up to some given point in time. In the written form the sequence is enclosed in angled brackets, so that, for example, a possible trace for the initial behaviour of the timer is

< set?time, set?time, reset!timeout >

The empty sequence $< >$ is the trace of the behaviour before it has started.

A process P can now be defined as the set of all traces of its possible behaviour; it follows that for any process P:

(i) $< >$ is in P; i.e. the set P is non-empty
(ii) if s, t are traces and if st, the concatenation of s with t, is in P then so is s by itself

If u is a symbol and P a process, $(u \rightarrow P)$ denotes a process that first 'does' u and then behaves like P; formally, using standard set-theory notation

$$(u \rightarrow P) = \{< >\} \cup \{< u > s \mid s \text{ is in } P\}$$

where u is the sequence consisting solely of the symbol u. By convention, the arrow associates to the right, so

$$u \rightarrow v \rightarrow P \;=\; u \rightarrow (v \rightarrow P)$$

If P, Q are two processes, $(P \square Q)$ is defined as a process that behaves either like P or like Q, the choice being determined by the environment in which it is placed. Formally, $(P \square Q) = P \cup Q$, and there is the obvious extension of the concept to several process, $(P \square Q \square R \ldots)$. In the protocol specification being considered here the choice is always govered by the comparison of two or more integer values or by the choice between inputs or outputs on one or more channels.

By convention, $\square$ is the most loosely binding of all operators, so

$$(u \rightarrow P \square v \rightarrow Q) = (u \rightarrow P) \square (v \rightarrow Q)$$

The alphabet of a process P is denoted by $\bar{P}$; usually we shall assume that the alphabet of a process is given by the set of all symbols occurring in the set of its traces, and therefore

$$\overline{(u \rightarrow P)} = u \cup \bar{P}, \qquad \overline{(P \square Q)} = \bar{P} \cup \bar{Q}$$

In the process $(P \parallel Q)$ resulting from the operation of P and Q in parallel each process ignores those events of the other that do not require its participation. If the two alphabets are the same then $(P \parallel Q)$ is just the intersection of the sets of traces of P and Q; if they are disjoint it is the set of all interleavings of a trace from P with a trace from Q.

The process $P[a/b]$ is defined as one which behaves exactly like P except that whenever P uses the name b, $P[a/b]$ uses the name a; here a, b can denote anything – channels, parameters, symbols etc. Thus if we want two processes P, Q with named channels a, b, respectively, to communicate with one another we create a common channel c and define P in parallel with Q as

$$(P[c/a] \parallel Q[c/b])$$

Recursive definitions are used to specify the behaviour of long-lasting or infinite processes. These recursions are to be understood in the same sense as the recursive equations of a context-free grammar such as is used, for example, in the definition of some programming languages. Thus a process that just copies on to channel b what it inputs on channel a would be written

$$\text{copy} = (a?x \rightarrow b!x \rightarrow \text{copy})$$

If $P(n)$ is defined as a process that behaves like P using the current value of n then

$$(u \rightarrow P(n+1))$$

will repeat u continuously with the value of n incremented after each repeat.

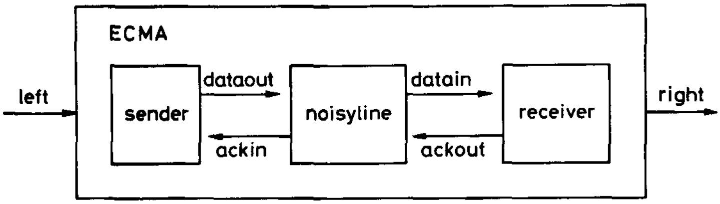
4 Specification of the protocol

4.1 The model of the protocol

In this specification the protocol has been modelled as three processes: a sender, a receiver and a noisy line, the last representing the action of the transmission medium which separates the first two. Conventionally, the sender copies messages from its left to the DATA channel on its right and expects acknowledgments of these on its ACK channel. The receiver copies messages from the DATA channel on its left to its own channel on its right and acknowledges them on its ACK channel. The noisy line copies messages (again conventionally) from left to right on the DATA channel and from right to left on the ACK channel. This network of processes is shown in Fig. 1.

4.2 The sender

As the protocol allows the sender to have a credit value greater than 1, more than one message can be active in the system at any one time and consequently there is not a strict sequence of data message followed by acknowledgment. Both sender and receiver must therefore be prepared to send or receive at all times. For the sender this means that it must be prepared to accept further messages on its 'left' channel while it is waiting for an acknowledgment or a timeout. For the receiver,



ECMA = (sender [dataout/data, ackin/ack]
||noisyline [dataout/datain, datain/dataout, ackout/ackin, ackin/ackout]
|| receiver [datain/data, ackout/ack]

Fig. 1 The ECMA protocol

which no longer has to acknowledge each message, it must have some criteria for acknowledging a sequence of messages while at the same time being prepared to receive more.

The commonest problem with communications networks is that messages are lost and do not reach the receiver. If the latter receives no message it will not send any acknowledgment, so the sender must be made to retry its message if it receives no acknowledgment (within some stated interval). This needs the idea of a 'timer' process which is started when a message is output and which informs the sender if a fixed period of time has elapsed since the last message was sent. If the sender has received no acknowledgment during this period it assumes that the message has been lost and re-sends it. If however the sender does receive an acknowledgment of its last message it does not want to be informed that the time period has elapsed, but instead wishes to restart the timer from zero when it sends its next message. So the timer must be able to restart at any point in its process, even when it is prepared to output a 'timer expired', or 'timeout', message. The timer is assumed to have two channels, one (SET) to input 'set timer' messages which contain the value of the time period to elapse before a timeout is issued, the other (RESET) to output the 'timeout' message.

The provision of a formally correct specification of a timer process of this kind, which also behaves in a manner usable in practice, is extremely difficult to achieve. To overcome this problem two specifications of the timer are provided both shown in Fig. 2. The first is formally correct as far as its behaviour appears to the outside world, in that is inputs 'set' and outputs 'reset' and never outputs more 'resets' than it inputs 'sets'. It is not, however, usable in practice because it either times out instantly or does so after a fixed period which cannot be varied. Also it cannot be reset until after the fixed period has expired, a feature which in practice would slow the message rate.

In a practical protocol it must be possible to vary the time before 'reset' in proportion to the 'time' associated with the 'set' and to reset at any time. For this, the specification is revised to include a countdown from the 'time' value which has been input; this gives the informal version shown in Fig. 2.

It will be noted that the 'time' that elapses before a timeout is not related to real time but to the value of the time set. In reality a delay of a fixed time period has to be introduced after each decrementing of the time value. Because of this inability to synchronize with real time this version of the timer specification cannot be accepted as formally correct; but if it is intended to attempt a future implementation of the protocol a practical timer is necessary and some loss of formality must be accepted. Either version can be used for this specification of the protocol because they have identical external characteristics.

Theoretical (formal) version:

```
timer  =  (set?time → timer1)
timer1 =  (reset!timeout → timer □ set?time → timer1)
```

Practical (informal) version:

```
timer  =  (set?time → timer1)
timer1 =  (set?time → timer1
           □ time = 0 → (reset!timeout → timer □ set?time → timer1)
           □ time ≠ 0 → time:= time-1 → timer1)
```

Fig. 2 The timer

An alternative approach would be to create a new time process each time it was necessary to set the timer, which would simply timeout when it had expired and would never need to be reset because copies which were superseded could be merely discarded. However, such an approach would not be usable in practice unless there were a means for dynamically creating and discarding processes.

A further problem for the sender is that when it is necessary to resend messages there may be more than one to be resent. Therefore a retry queue which is only one message deep is insufficient, and the queue must be as deep as the maximum credit value. It is important to avoid the need for separate processes for the initial sending of messages and the resending, when necessary, of those same messages; consider therefore the idea of queuing messages received from the left channel and transmitting messages from this queue whether they are re-transmissions or not: this has the extra benefit of providing a certain amount of buffering between the left channel and the data channel.

The size of queue which is theoretically most convenient is an unbounded queue. The sequence number of any message is then defined by its position in the queue taken modulo $(N+1)$ where N is the maximum sequence number, and does not need to be stored. This means that the queue can contain an unlimited number of entries and also that it can contain more than one inner queue with pointers to show the heads and tails of the queues. Pointers are needed to indicate four things:

- the last message received from the 'left' channel (i)
- the last message sent on the 'data' channel (n)
- the last message acknowledged by the receiver (a)
- the last message whose maximum lifetime has expired (l)

In practice it is easier to have each of these pointers pointing to the first queue item available after the last message mentioned.

There are five events which the sender must be prepared to process at any time:

(i) when a message is input on the left channel it is stored in slot i of the queue and i is incremented
(ii) when any message has been input but not yet sent the message in slot n of the queue should be sent and n incremented. This must be constrained by the fact that the number of messages sent but not acknowledged must not exceed the current credit value c
(iii) when an acknowledgment is received the pointer a is updated to the sequence number given in that acknowledgment (which may be the same as that of the previous acknowledgment) and the new credit value c is stored
(iv) when timeout occurs the pointer to the last message sent (n) is reset to the sequence number of the last acknowledgment (a); this will cause the unacknowledged messages to be resent (See Appendix, Note 1)
(v) when the maximum lifetime of a message in the queue expires l is incremented; this event is not catered for in the definition at present.

Thus the definition of a process which controls the sending of messages, receiving of acknowledgments, queueing and unqueueing of messages and setting and checking of timeouts is as follows:

$$\text{sendmsg} = (S(c, 0, 0, 0))$$

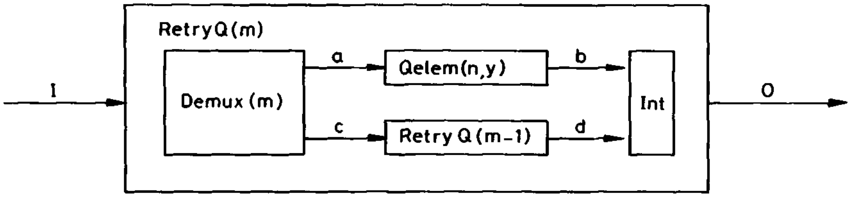
$$\begin{aligned} S(c,a,n,i) &= \text{left}?x \rightarrow Qi!x \rightarrow S(c,a,n,i+1) \\ &\quad \square\ (n \neq i) \& (n-a) < c \rightarrow Qn?x \rightarrow \text{data}!(x,n) \rightarrow \text{set}!\text{time} \rightarrow S(c,a,n+1,i) \\ &\quad \square\ \text{ack}?(x,y) \rightarrow S(x,y,n,i) \\ &\quad \square\ \text{reset}?\text{timeout} \rightarrow S(c,a,a,i)) \end{aligned}$$

(see Appendix, Note 2)

This assumes that the retry queue is treated as an unbounded set of variables, each with its own channel Qx, and that c is the most recent credit value received. For the purpose of the specification a bounded retry queue is more convenient; 'sendmsg' is therefore redefined later in the paper to have this characteristic.

If the credit value received on an acknowledgment actually reneges on a previously received value – that is, it is lower – this does not affect the action of the sender. If the new credit value has not yet been exceeded it is simply used instead of the old value in checking whether or not more messages can be sent. If the new value has been exceeded the sender will resend the excess messages after the next timeout, assuming that the new credit allows them to be sent. The sender will not attempt to adjust to the lower credit value prior to receiving the next timeout (see Appendix, Note 3).

In practice the retry queue is defined somewhat differently, because one cannot define an unbounded queue or a variable number of channels to the queue. It is necessary to define a finite queue, denoting the length by m, and the most convenient length, if practically possible, is the maximum sequence number. So the queue actually consists of a number of elements, each of which will either input and store a new value from its channel I or will output its current value on its channel O, depending on whether it receives the signal Q, for 'queue', or U, for 'unqueue'. The element must first input its own index number in the queue, which in this case is the sequence number of the queued message. Thus element 'n' is defined as 'Qelem(n,y)' in Fig. 3. Note that the parameter n of 'Qelem' is used only to distinguish the various instantiations of 'Qelem' and is not used directly to indicate the queue element number.



RetryQ(m) = (Demux(m) [a/On, c/O] for $m \geq 1$
|| Qelem(m, x) [a/I, b/O]
|| Interleave [b/a, d/b, O/c] || RetryQ($m-1$) [$c/1$, d/O])

RetryQ(O) = (Qelem(o, x))

Demux(n) = ($I?x \rightarrow (x = n) \rightarrow On!x \rightarrow I?e \rightarrow On!e \rightarrow$
$(e = U) \rightarrow$ Demux(n)
□ $(e = Q) \rightarrow I?x \rightarrow On!x \rightarrow$ Demux(n)

□ $(x \neq n) \rightarrow O!x \rightarrow I?e \rightarrow O!e \rightarrow$
$(e = U) \rightarrow$ Demux (n)
□ $(e = Q) \rightarrow I?x \rightarrow O!x \rightarrow$ Demux(n))

Qelem(n, y)=$I?x \rightarrow (x=U) \rightarrow O!y \rightarrow$Qelem($n$, y)

□ $(x=Q) \rightarrow I?x \rightarrow$ Qelem (n, x))

Fig. 3 The queue

Notes: channel I: inputs elements number and 'U' for 'unqueue' *or* inputs elements number, 'Q' for 'Queue' and element value
channel On: outputs what was input on I if the element to be dealt with is the same one as handled by that instantiation of 'Demux (m)'
channel O: outputs what was input on I to the next retry queue process if the element to be dealt with is not the same as handled by that instantiation of 'Demux(m)'

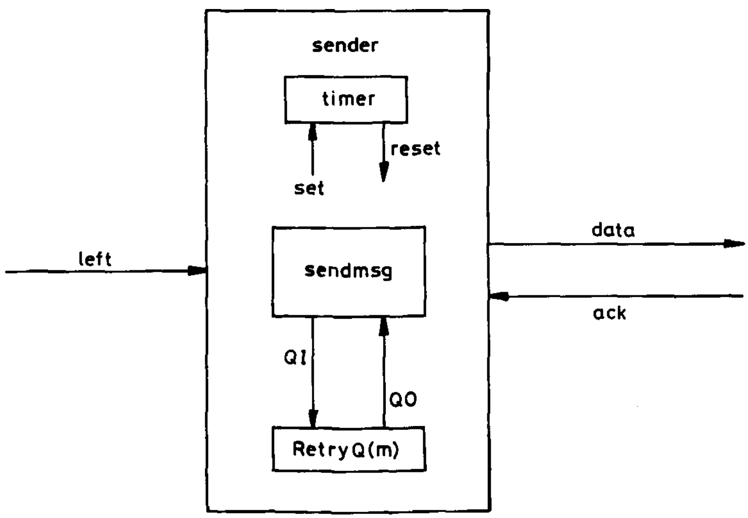
'Interleave' is a process which takes inputs from either of two channels a, b and puts them out on a third c:

Interleave = ($a?x \rightarrow c!x \rightarrow$ Interleave
□ $b?x \rightarrow c!x \rightarrow$ Interleave)

It is also necessary to define just one input channel QI and one output channel QO. When we want to store an item in the queue we output to it the index number of the item, a 'Q' to indicate that we are queueing and the value of the item. When we want to read an item from the queue we output to it the index number of the item and a '*U*' to indicate unqueueing, and then input the value of the item from the queue.

Each element in the queue is preceded by a demultiplexer which decides if the current item is for its own element or for a later one. It does this by comparing the first input on channel *I* against its own index number; if they are equal the current set of inputs is passed to its own element on channel On, otherwise they are passed to the rest of the queue on channel *O*. In Fig. 3 the demultiplexer for channel *n* is identified as Demux(n). The outputs from the queue elements are interleaved to give the final output from the queue; in Fig. 3 a retry queue with *m* elements is identified as RetryQ(*m*).

The number of messages input to this finite queue but not expired must not exceed the maximum sequence number, for otherwise the sequence numbers would 'wrap round' and messages input from the left channel could corrupt unexpired messages still in the queue. The 'sendmsg' process must therefore be additionally constrained by this condition. As the value of the maximum lifetime of a message in the system is supplied by the lower levels of the protocol, and as these are not included in present model, this constraint is, not included in this version of the specification. The potential danger caused by the lack of his guard can be lessened in practice by




```
sender   =  (sendmsg || timer || RetryQ(m) [QI/I, QO/O])

sendmsg =  (S(c,0,0,0))
S(c,a,n,i)= (left?x → QI!(i,'Q',x) → S(c,a,n,i+1)
            □ (n ≠ i) & (n−a) < c → QI!(n,'U') → QO?x → data!(x,n) →
                    set!time → S(c,a,n+1 i)
            □ ack?(x,y) → S(x,y,n,i)
            □ reset?timeout → S(c,a,a,i))
```

Fig. 4 The sender

using a maximum sequence number large enough to prevent the processing of that number of messages in less than the given time (see Appendix, Note 4).

The informal practical version of the sender, consisting of the message sending process in parallel with a timer and a message retry queue, is shown in Fig. 4.

4.3 The receiver

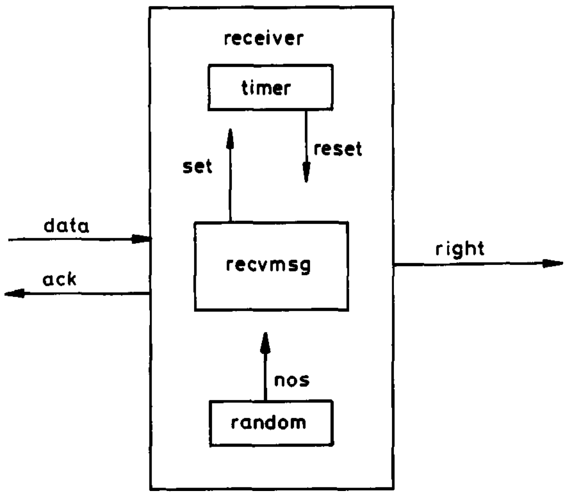
As stated previously the receiver must now, like the sender, be prepared to accept more messages at any time. Also, as it does not necessarily have to acknowledge every message received, it must have criteria for deciding when to send an acknowledgment and what credit to give the sender. In the real protocol the value of the credit is controlled by a kind of 'back-pressure' from the higher levels: in effect, the receiver has to buffer the messages it receives and the credit is an indication of the amount of buffer space available. As these higher levels do not exist in the present model and as the 'right' channel is unbuffered and does not create 'back-pressure', an artificial means is needed for generating credit values and acknowledgment criteria. The following simple method is used: messages are acknowledged whenever they are received and the credit value is reduced by 1 at each acknowledgment; when the credit reaches zero a new value is obtained from a random process and is sent with the next acknowledgment. This has the virtue of behaving correctly from the point of view of the sender. The random process, shown in Fig. 5, is one which outputs random positive integers on demand.

To control this processing, sequence number pointers are needed to indicate three things:

- last message received on the 'data' channel (n)
- last message output on the 'right' channel (o)
- last message acknowledged on the 'ack' channel (a)

As with the sender, it is actually easier to have each of these pointers pointing to the first available sequence number after the last message mentioned. Again in a theoretical model these numbers can be unbounded, whilst in practice they are modulo (maximum sequence number + 1). A note must be kept also of the current value of the credit outstanding (c) so that this can be included in all acknowledgments.

Normally the receiver has no need to repeat acknowledgments if no data are subsequently received, because it is the responsibility of the sender to retry unacknowledged messages. However, when the receiver sends an acknowledgment which effectively opens the credit window after this has been reduced to zero it must set a time-out so that it can repeat this acknowledgment if no data are subsequently received. This is because the sender will at this point have zero credit value and so will make no attempt to restart the link if the acknowledgment re-opening the credit window is lost. The receiver must therefore contain a timer similar to that used by the sender, so that it can notify the sender that a certain



receiver = (recvmsg || timer || random)

recvmsg = $(R(c, 0, 0, 0))$

$R(c, a, n, o)$ = (data?(x,y) → $(y=n)$ → right!x → set!time → $R(c-1, a, n+1, o+1)$
□ $(y \neq n)$ → $R(c, a, n, o)$
□ $(n \neq a)$ → ack!(c,n) → $R(c, n, n, o)$
□ $(c = 0)$ → nos?x → ack!(x,a) → set!time → $R(x, a, n, o)$
□ reset?timeout → ack!(c,n) → set!time → $R(c,a,n,o)$)

random = (RO)

RO = (nos!x → RO) where $0 \leqslant x \leqslant (\text{maxseq} + 1)$

Fig. 5 The receiver

time has elapsed since the last acknowledgment was sent; the same definition of 'timer' is used, but the sender's process and the receiver's process are logically distinct. When the timeout expires the receiver simply repeats its last acknowledgment and restarts the timer. There is no need to set the timer after sending other acknowledgments.

The timer can also be used as an 'inactivity timer' to keep the link 'warm' when no data are being transferred. The timer is set after each message is received and the last acknowledgment is repeated if no subsequent message is received before the timer expires. The timer value set for the inactivity timer should be larger than for the acknowledgment and should only repeat frequently enough to prevent disconnection of the link (see Appendix, Note 5).

Thus there are four events which the receiver must be prepared to process at any time'

(i) when a message is received on the data channel its sequence number is checked against the next expected sequence number (n); if it is correct, the message is output on the 'right' channel, the inactivity timer is reset, the outstanding credit value (c) is decremented and the sequence numbers of the next message expected (n) and the next to be output (o) are incremented. If the sequence number received is incorrect the message is ignored. The receiver does not check whether the credit value it last sent has been exceeded as this is the responsibility of the sender (see Appendix, Note 6).

(ii) when a message or messages have been received but not acknowledged, an acknowledgment is sent on the 'ack' line containing the reduced credit value (c) and the sequence number of the next message expected (n). The sequence number of the next message to be acknowledged (a) is updated to the next message expected.

(iii) when a credit value falls to zero a new value is input from the random process and an acknowledgment is sent, containing this value and the sequence number (n) of the next message expected. The timer is reset as the credit is being raised from zero and the new value replaces the previous one (c) in future processing. This action is artificial and is included for the reasons given above.

(iv) when the timeout expires the last acknowledgment is repeated and the timer is set again (see Appendix, Note 7).

4.4 The noisy line

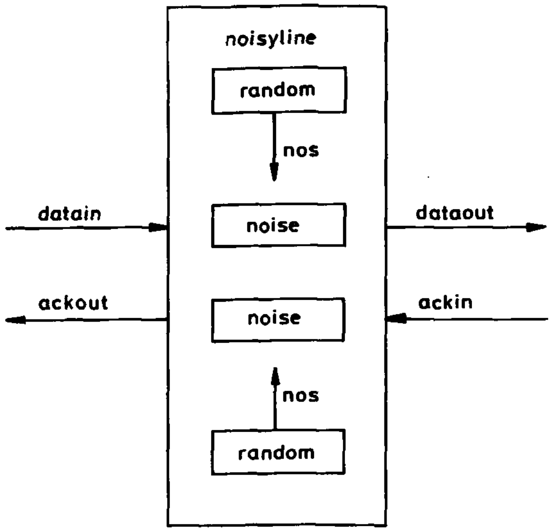
As stated in the Introduction, for a practical protocol we need to take into account the unreliability of the transmission medium over which the messages are sent. This can be modelled as a process which communicates with the sending process (on its left) and with the receiving process (on its right). As an example of an unreliable medium we consider a noisy line which loses messages and acknowledgments randomly; in the model the choice of which to lose is decided by the output from a process which generates random integers. The generating process could be designed to simulate the behaviour of any particular network, but for this specification it is sufficient to use a random process having the same definition as that used to provide credit values to the receiver.

Two noisy-line routines of this type are combined into a single process which passes messages in either direction on the 'data' or 'ack' lines, and loses the nth message in each direction. This is shown in Fig. 6, the random process being the same as that of Fig. 5.

5 Conclusions

The specification derived here of a part of the ECMA-72 Class 4 transport protocol suggest that the CSP language does provide a clear and precise method of describing the action of the protocol; and there is no evidence in the work so far completed to suggest that this specification could not be extended to encompass the whole of the protocol. The full specification so produced would be shorter, more exact and less ambiguous than the existing English language specification.

The provability of the specification so produced is less clear. The production of



noisyline = (noise[datain/left, dataout/right] || random
|| noise [ackin/left, ackout/right] || random)

noise = (nos?n → noise(n))

noise(n) = (left?(x,y) → (n=0) → nos?n → noise(n)

□ ($n \neq 0$) → right!(x,y) → noise ($n-1$))

Fig. 6 The noisy line

a full formal proof would be a difficult and time-consuming task and might only be possible if some processes, like the timer, were specified in an unrealistic manner. Also the proof method given by Chen and Hoare[3], which would form the basis of any proof, has an admitted deficiency in that it cannot prove or even express the absence of deadlock. In a later paper[4] Hoare states that the absence of a deadlock can be proved by showing that a protocol conforms to the definition of 'BUFF', i.e. that it is a buffer; but it is not clear whether or not it is always possible to prove that a protocol conforms to this definition. It would be difficult for instance to show the difference in the ECMA protocol between a deadlock and a valid disconnection caused by excessive noise on the line. An alternative method for partial validation would be by simulating the action of the complete protocol, using an abstract machine system.

During the project various discussions were held both within the Programming Research Group at Oxford and with ICL staff on the specifications produced. It was noticeable that the use of the CSP specification as a basis for discussion made it possible to focus much more rapidly on essential points, and to identify areas of disagreement more readily, than would have been possible using an English language specification.

One question raised by this project is the problem of defining a standard for a protocol. A standard is not a description of the action of the protocol but rather a set of limitations on its behaviour; several different behaviours within these limitations may all constitute valid implementations of the standard. What is defined in the specification given in this paper is not *the* ECMA-72 Class 4 transport protocol standard but a protocol which conforms to the standard. For example, at one point in the project the specification of the receiver was valid but was not what one might have expected. An alternative but equally valid version was later substituted in the interests of being more 'natural'. Specifications of the standard at a level of abstraction where implementation issues were largely absent would be a larger task and may not even be possible by this method. It might be possible to specify a process which contained all possible behaviours within the standard, and this could then be said to be a definition of the standard. Alternatively, formal specification techniques could be used as a way of making standards more restrictive by specifying permitted behaviours more exactly.

Acknowledgments

The work covered in this paper was undertaken as part of the M.Sc course in Computation at Oxford University, and applies techniques learned and knowledge gained during that course. I am grateful to all the staff of the Oxford University Programming Research Group for their teaching and guidance during the year, especially to Dr. Peter Henderson who supervised my work on the project, and to Prof. Tony Hoare whose lectures and papers on Communicating Sequential Processes provided a basis for the work. I am grateful also to Ken Turner, Andrew Brightwell and Graham Pratten of ICL Kidsgrove, who were instrumental in organising the project and who made many interesting and relevant comments on the specifications during various discussions. Particular thanks are due to Ken Turner for turning a student dissertation into a paper suitable for publication.

References

1 European Computer Manufacturers Association 'Standard ECMA-72 Transport Protocol (Second Edition)', ECMA, September 1982

2 HOARE, C.A.R.: 'A model for communicating sequential processes'. Programming Research Group (Oxford) Technical Monograph PRG-22, June 1981

3 ZHOU CHAO CHEN, and HOARE, C.A.R.:
(a) 'Partial correctness of communicating sequential processes, and
(b) 'Partial correctness of communication protocols'.
Technical Monograph PRG-20, May 1981

4 HOARE, C.A.R.: 'A calculus of total correctness for communicating processes'. Technical Monograph PRG-23. April 1981

Appendix

Notes on points of difference from ECMA-72.

1 (p.304) From the published specification, ECMA-72 appears to require that only the first unacknowledged message is retransmitted in the event of a timeout.

The use of digital techniques for voice has been possible since the invention of pulse code modulation (PCM) in 1936, but it is only in recent years that digital microelectronics techniques can be justified on economic grounds for the whole of the switched voice service. Once digital voice operation is introduced into the network it paves the way for a range of nonvoice services which can be carried on the same channels as voice.

2 Current developments

2.1 Integrated digital network

The reason for the increasing popularity of digital voice techniques lies in the use of low-cost time-division multiplexing techniques that can be used once the signals are in digital form. Furthermore, time-division switching of the multiplexed channels greatly reduces the costs of switching at intermediate nodes in the network. Because of the close combination of transmission and switching the concept has become known as the integrated digital network or IDN.

In the network shown in Fig. 1 the cost of analogue-to-digital (A/D) conversion has to be offset against the savings possible in the multiplexed transmission and intermediate switching. Until recently, an overall advantage could only be achieved by locating the A/D convertors after concentration of the subscriber traffic. The concentrator was based on conventional analogue switching techniques such as reed relays or crossbar switches. Advances in large-scale integration now permit the A/D conversion to be introduced at the subscriber line input to the exchange.

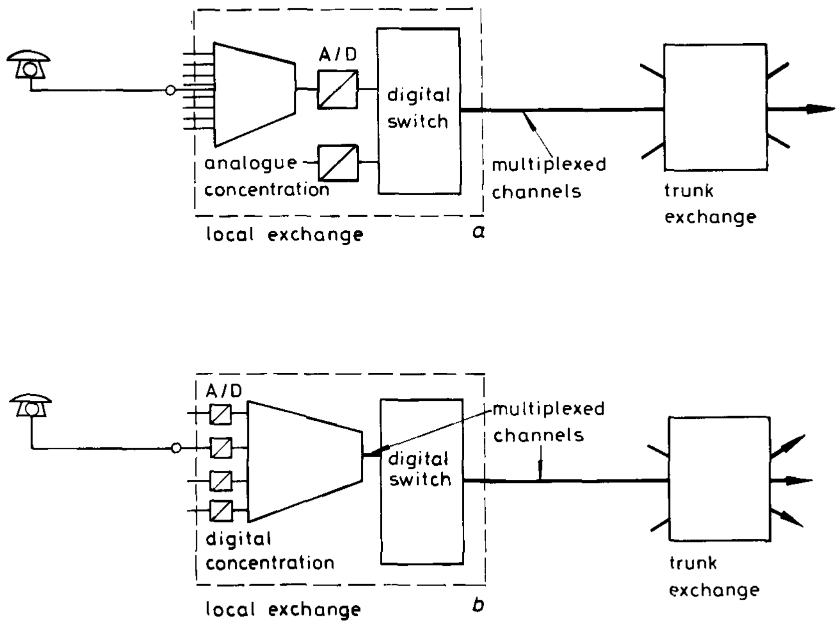


Fig. 1 Stages of development of the integrated digital network (*a*) A/D conversion after concentration (*b*) A/D conversion before concentration

This is the basis of the digital System X network that is rapidly being established in Britain[1].

2.2 *Integrated services digital network*

Once the concept of a network that employs digital operation right through from the local exchange has been established, it is an obvious further step to extend the 64 kbit/s digital channels out to the subscriber. For voice, this involves essentially moving the A/D convertor out to the subscriber and, apart from a possible improvement in transmission, gives no advantage. However, the digital channel can now be used for a wide variety of nonvoice services at speeds up to 64 kbit/s. The system is upgraded to an integrated services digital network or ISDN[2], as shown in Fig. 2. This represents a considerable improvement over the Datel service, which cannot be extended above 9·6 kbit/s for many connections through the (analogue) switched network.

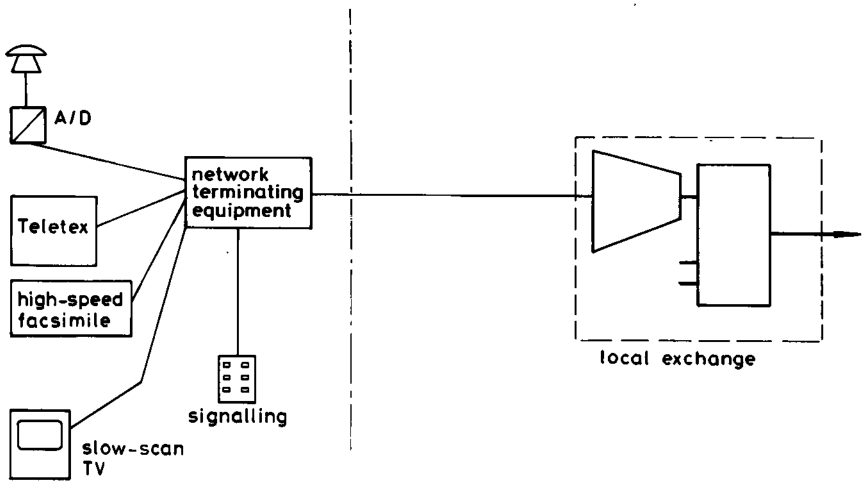


Fig. 2 Integrated services digital network (ISDN)

The extension of 64 kbit/s channels to the subscriber does not present any major technical difficulties. The easiest approach is to use two pairs of wires, one for each direction of transmission, but this is expensive in many cases since it may involve additional cabling. Techniques for bothway transmission over existing cables for distances up to about 5 km have been demonstrated successfully. One approach involves the transmission of 'bursts' of bits at speeds in excess of 64 kbit/s alternately in each direction. This is known as time-division duplex or 'burst mode'. A more advanced system uses an adaptive echo-cancelling technique to give improved balance over the normal bridge duplex.

In the system to be introduced in Britain in 1984, 80 kbit/s will be available in each direction over a single pair of wires to give:

– a 64 kbit/s 'B' channel for either voice or data

– an 8 kbit/s 'B1' data channel that can be used simultaneously with the 64 kbit/s channel and to a different destination. (In the System X network it will be carried on a 64 kbit/s channel.)
– an 8 kbit/s bothway signalling 'D' channel, which can be used without affecting the other traffic.

Connections for nonvoice services will be made to a standard X21 or X21 *bis* interface[3].

Current discussions indicate that the internationally recommended standard for connection to the ISDN may differ somewhat from that described above, but the basic principles of access to 64 kbit/s IDN channels and adequate signalling capacity will still apply.

2.3 *Packet switching*

Whereas the ISDN is essentially a digital voice network that has been extended to carry nonvoice services, the packet switched network[4] is an attempt to take into account the special needs of some data services. Although many claims have been made for a packet switched network the prime justification for its introduction is economic.

For some forms of data transaction a relatively quick response is required but the information transmission is sporadic, extending over possibly several hours. It would not be feasible to set up a new call over the existing telephone network each time a transaction takes place. Equally, it would be very expensive to hold the connection over the total period, with the circuit idle for most of the time.

A packet switching system introduces the concept of a 'virtual call' which is logged at the network nodes over an extended period but in which transmission capacity is engaged only when there are packets to be sent. The tariff, apart from the rental of lines and line terminal equipment, has then two elements:

– a time charge which is much less than the time charge for the corresponding telephone network
– a charge based on the number of packets sent.

For many data users the pattern of activity is such that the packet switched service is less costly than Datel over the telephony network even though the packet switched network is relatively expensive, since it has had to be established as a separate network.

2.4 *Network interconnection*

Although the packet switched network is normally treated as completely separate from ISDN (albeit with some sharing of transmission plant), the services offered by the two networks are not clearly distinguishable as are voice and telex. However, it is by no means a simple matter to provide for full interconnection between the PSS and ISDN networks. Apart from the differences in protocol, differences in data rates and numbering plans make it doubtful if the provision of full integration is worth while.

A more realistic approach is to keep the two networks essentially separate but to permit transparent access to the PSS network from ISDN subscribers. This is an extension of the present access from the telephony network via a packet assembler/disassembler but provides for full X25 operation as shown in Fig. 3. The local ISDN switch is only used in establishing level 1 of X25 and then acts as a transparent connection for levels 2 and 3. Outgoing calls can be made without difficulty but an incoming call can only be accepted if a level 1 connection has already been established.

More complex forms of interconnection have been proposed, but this raises the general question of the number of networks it is desirable to establish to carry a range of telecommunication services.

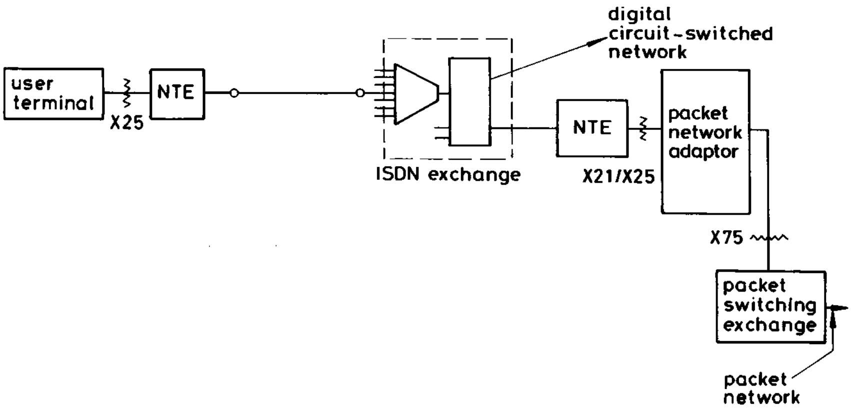


Fig. 3 Access to the packet network from ISDN. NTE = network terminating equipment

3 Multiservice networks

A possible approach to the provision of a range of voice and nonvoice telecommunication services would be to establish a new network for each new service or group of services. However, a telecommunications network can only realise its full potential if it is available over a wide geographical area and it would be extremely expensive to establish a widespread network for each telecommunications service. The commercial risk would also be considerable since the takeup of the new service might not be as rapid as expected. Such consideration would inhibit the provision of new services.

On the other hand, a unified network capable of carrying a wide range of services would have a great advantage in the establishment of new services since no great commercial risk would be involved. It would be necessary only to provide the terminal equipment to establish a new telecommunications service on the unified network.

A further advantage of a multiservice network is that it simplifies user operation by providing a single access and a unified numbering plan. This is of particular importance where networks are interconnected. The requirement to access a terminal on one network from a different network, with its own numbering plan, greatly increases the complexity of the operation and thereby the possibliity of user error.

4 Implications for network design

4.1 Characteristics of telecommunication services

If we accept for the moment that a multiservice network is a desirable goal, then there appears the considerable problem of designing the network to be able to absorb a wide range of services, even those not yet defined. A range of potential services is shown in Table 1. These can be considered in four main categories, although combinations of services may be required in specific applications.

Table 1 Customer telecommunications services

Data and Text services	Voice services	Still picture services	Real time picture services
telegraph	telephone	facsimile	slow-scan TV
telex	voicegram	– line drawing	high definition TV
teletex	voicedata	– halftone	picturephone
telecommand	radiophone	– colour	confravision
intercomputer data	hi-fi (music)	picture viewdata	colour TV
electronic fund transfer	stereophonic	home newspaper	stereo TV
home newspaper	conference	still hologram	moving hologram
radio paging	etc.	etc.	etc.
alarms			
viewdata			
etc.			

4.1.1 ***Data and text services:*** For man/machine communication based on alphanumeric text, bit rates of 256 kbit/s are adequate even for rapid browsing (reading a few words on each page). Machine/machine communication is a relatively new field and, although bit rates currently in use are low, higher bit rates may well be required in the future.

4.1.2 Voice services: The current use of 64 kbit/s for digital voice, although widespread, may now be regarded as an historical accident. 'Commercial' quality speech can be transmitted at bit rates as low as 24 kbit/s[5]. Lower bit rates are possible if some reduction in naturalness can be accepted. On the other hand, 64 kbit/s is hardly adequate for high-quality stereophonic transmission.

4.1.3 ***Still picture services:*** The requirements for this type of service are very flexible since the full information is normally available before the start of transmission. However, pictures and diagrams are so often associated with text that it is often necessary to consider the two basic services together.

4.1.4 Real-time picture services: Real-time pictures require a wide range of bit rates, as shown in Fig. 4. The need for greater quality and realism leads to higher bit rates, but this is counteracted to an unknown extent by advances in redundancy techniques.

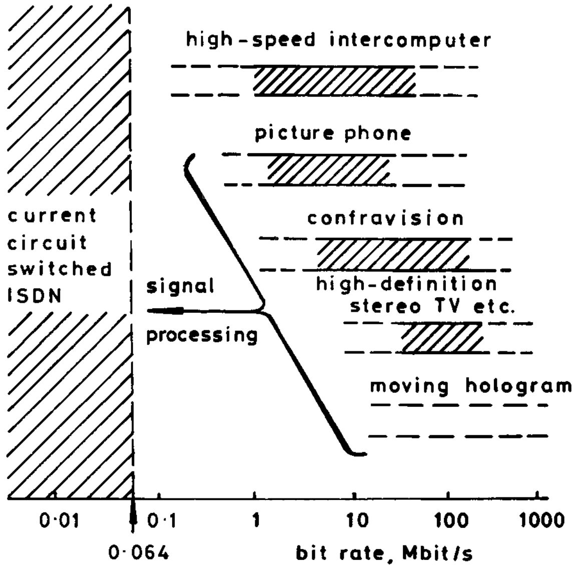


Fig. 4 Wideband services

At this stage it is worth while to revert to fundamentals and consider those characteristics of the customer services that need to be mapped onto the network, which is then considered as a bit-transport system. It is assumed that only digital operation is used and that all signals are converted to digital form before they are offered to the network.

4.2 *Call characteristics*

A call may be regarded as a condition in the network during which information input at one terminal will be output to one or more other specified terminals of the network. It is a concept which dates from the very first switched networks and has now, in effect, been incorporated in the 'Session' level of OSI[6].

The call connection characteristics that need to be defined include:

- call duration characteristics (holding times)
- number and configuration of terminals involved. A simple call involves only two terminals but more complex services may require 1:*n* (selective broadcast) or *n*:*n*-1 (conference) connections.

It is also necessary to indicate the limiting call impairments that can be tolerated, including for example:

- proportion of lost calls ('grade of service')
- delays in setting up calls.

4.3 Communication characteristics

The communication characteristics of the information transactions after the call has been established for each telecommunication service include:

- bit rate profiles (i.e. the variations of bit rate during the call)
- symmetry of information flows
- storage that may be required if a receiving terminal cannot immediately accept the information being sent by the transmitting terminal.

The limiting communication impairments that can be tolerated by the service include:

- transmission delays
- lack of transparency
- loss of information
- errors.

4.4 Signalling

In addition to the basic communication across the network, it is necessary for control signals to be exchanged between each terminal and the switched network. Such signals need to be sent and received at least during the set up and clear down phases, but it is preferable to permit signalling in both directions at any time during the call and preferably without interrupting the basic communication.

It may also be necessary to permit control signals to be exchanged between terminals. This can be achieved either by providing transparent communication between the terminals or by sending the control signals into the network and repeating them to the other terminal. This latter approach has the advantage that the network control is made aware of end-to-end signals that are relevant to the operation of the network. An example of this would be a requirement to switch the terminal equipment (e.g. text to facsimile) during the call.

5 Implications for network design

Even if the telecommunication service characteristics could be defined in the above terms for all known services, the design of a cost-effective multiservice network would not be easy. However, because of the considerable inertia in the geographical coverage, it is desirable that the network should, in its basic framework, be configured so as to meet the needs of the future, as well as existing, services.

This may appear to present an almost insuperable problem but there are signs now that the achievement of such a forward-looking network could become a possibility. First, the basic technologies, particularly microelectronics, have reached a stage of maturity that enables fast-acting logic and switching operations to be carried out at an acceptable cost. Secondly, minor impairments in a service can often be compensated by the design of terminal equipment provided the major network facilities are provided economically.

However, there are some key requirements for telecommunication services that must be met by the network.

5.1 Bit rate

In an multiservice network the bit rate available for a communication channel

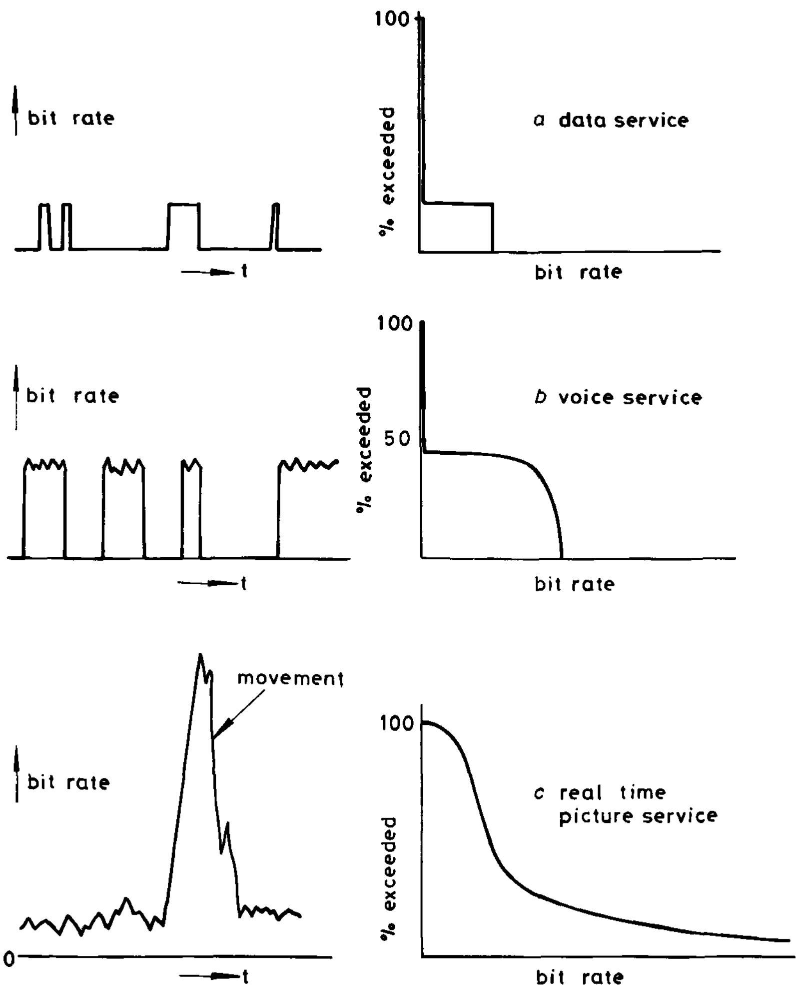


Fig. 5 Bit rate profiles showing typical variations in bit rate with time and the fraction of time for which a given bit rate is exceeded

is of prime importance. Fig. 5 shows typical instantaneous bit rate profiles for some of the common telecommunication services, but variations over a wide range may be expected, particularly for new, and as yet undefined, services.

A multiservice network must be capable of transporting bits at the highest rate required by a user terminal both now and in the future.

5.2 *Delay*

Propagation delays are inevitable in any telecommunications system but additional delay may be introduced at a switching stage. For example, if synchronous multiplexed channels are to be switched in a matrix it is possible that a particular time slot is already occupied at the outlet by signals from another source. A time switch is then used to delay the signals in that time slot to a vacant slot, either later in the same frame or to the next frame. The maximum delay at each switching is thus equal to one frame minus one time slot.

5.3 *Error rates*

Errors must be expected in any practical system. In some cases these can be tolerated but for some services it is essential to correct the errors before the message is delivered to the receiving terminal. Forward error correction is possible but very wasteful in transmission capacity. Luckily, services that require negligible error rates can normally tolerate additional delays, so that it is possible to store the block in which errors have been detected to permit a repetition to be obtained.

6 Design of a multiservice network

The design of any switched network falls naturally into two parts. The first part is concerned with local feeder distribution and the concentration of traffic so that signals from several sources can be multiplexed. The second is the switching and transmission involved in passing the traffic through the network and delivering it to the required destination node.

6.1 *Local feeder distribution*

If a 'star' form of local feeder distribution is adopted a 'pair' of conductors or optical fibres is provided between each user terminal and the concentrator node as shown in Fig. 6*a*. The distances involved are usually quite short (less than about 5 km) so that transmission capacity can be provided on a liberal basis.

There is an economic advantage if only one 'pair' need be provided to each user premises. If the existing telephone distribution network is to be exploited, only one pair of copper wires is available to many user premises. It has been shown possible to transmit at speeds up to 80 kbit/s (possibly up to 144 kbit/s) in both directions over most of the pairs in a cable.

If wideband feeder distribution is required, coaxial cable or optical fibre must be used. The cable should be intrinsically capable of carrying the maximum bandwidth

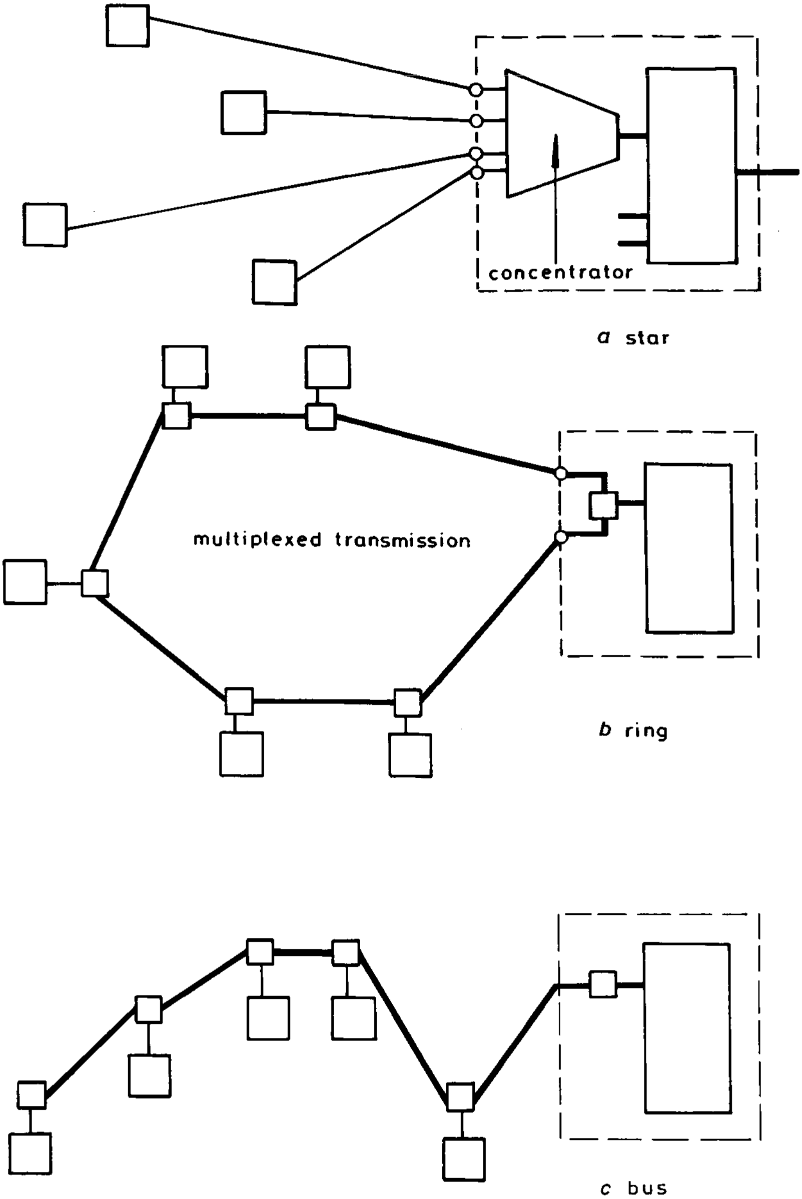


Fig. 6 Local distribution services

that might be required in the future since it is hardly feasible to dig up the city streets every time an increase in bit rate is required. With optical-fibre technology the provision of ample bandwidth over 5 km or so does not impose an appreciable cost penalty on the system.

The main problem in providing a wideband distribution system lies not so much in the technology but rather in the start-up economics. Unless a large number of subscribers are connected to the network within the first few years, it becomes very difficult to justify the provision of wideband cables over a wide geographical area.

One proposal is to use entertainment TV as the basis for the installation of such a distribution, although some of the schemes currently under consideration do not readily admit two-way point-to-point communication. An alternative, which is particularly advantageous in town business centres, is to use microwave radio links in the initial stages of development. Since only short distances are involved it is possible to use one of the absorption bands (e.g. 60 GHz) to facilitate frequency reuse in nearby towns. As the number of users increases, wideband cables can be laid and the microwave links recovered for use elsewhere.

In some local distribution systems a ring or bus structure may be used as shown in Figs. 6*b* and *c*. The concentration function is inherent in the distribution system but the basic principles still hold. Such systems have been designed mainly for operation within a building or at least within a restricted site, but there may be advantages in using them over a somewhat greater area.

Overall, the design of the local distribution to meet the needs of future services is less of a problem than the design of the main trunk network. There appears to be only relatively small economic penalties in designing the cables and concentrator switches to provide ample bit rates, and it is possible to provide an open-ended signalling system that will meet the needs of the future.

6.2 *Main trunk network*

After traffic concentration, some form of multiplexing is essential for the transmission links. At the switching nodes it is possible to switch the signals in multiplexed form or, as has been the case until recently, to separate the channels before switching and recombine them afterwards.

6.2.1 Multiplexed transmission: There are three options for multiplexing, and these are illustrated in Fig. 7.

The synchronous fixed slot allocation (Fig. 7*a*) is the technique used for the current ISDN system. It is characterised by a fixed duration frame divided into a fixed number of time slots. A call is allocated a slot during setup and that slot is normally used throughout the duration of the call. The main disadvantage of such a system is that it offers only a fixed bit rate. It is possible to subdivide the slots but, in a switched network, such an arrangement leads to considerable complication unless all the subchannels are connected to the same premises. For services requiring a higher bit rate more than one slot may be allocated at the beginning of the call but sufficient slots must be provided to carry the maximum bit rate required.

Packet operation involves the transmission of groups of bits or packets usually of varying lengths, with each packet being preceded by an address indicating the destination (Fig. 7*b*). Packets may be sent as frequently as required, up to the

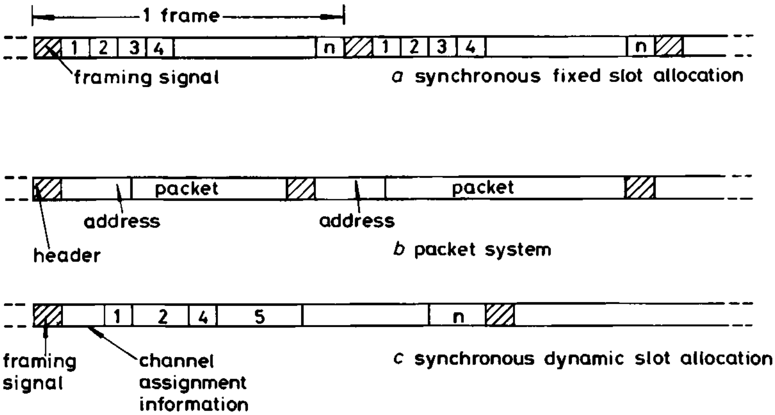


Fig. 7 Multiplexing systems

limit of the system, but may be subjected to variable and relatively long delays.

It is also possible to use a dynamic slot allocation technique as a basis of multiplexing (Fig. 7*c*). The structure of a fixed duration frame divided into a predetermined number of slots is retained as for the fixed slot allocation system, but there is no longer a one-to-one correspondence between slots and channels. During the frame interval each communication channel is allocated as many slots as required, the number being determined by the instantaneous bit rate of the source. Additional signalling information must be sent with each frame to indicate the channel/time slot relationships. If the aggregate instantaneous bit rate of all the channels is such that more slots are required than are contained in the frame a condition known as 'freeze out' occurs;

The fixed slot allocation approach leads to a *fixed bit rate* system or, if multiple slots are allocated at the beginning of a call, it may be regarded as a *preset bit rate* system. Both packet multiplexing and dynamic slot allocation give rise to *variable bit rate (VBR)* systems.

6.2.2 Multiplex switching: Once the traffic is concentrated and in time-division multiplexed form it is advantageous for economic reasons to adopt a multiplexed approach to switching. Fig. 8 shows the forms of typical switching nodes corresponding to the multiplexing methods described above.

In the fixed slot allocation the information from a particular incoming route or an adjacent concentrator switch is first 'time switched' by writing into a store the contents of each channel time slot in sequence and reading out from each store location when there is a free path through the matrix switch to the second time switch associated with the outgoing route. The time switch is controlled by a simple control store driven in synchronism with the multiplexing frame. The control store is written with the address of the information store location at call setup, and retains that information for the duration of the call. A similar process takes place in the second switch.

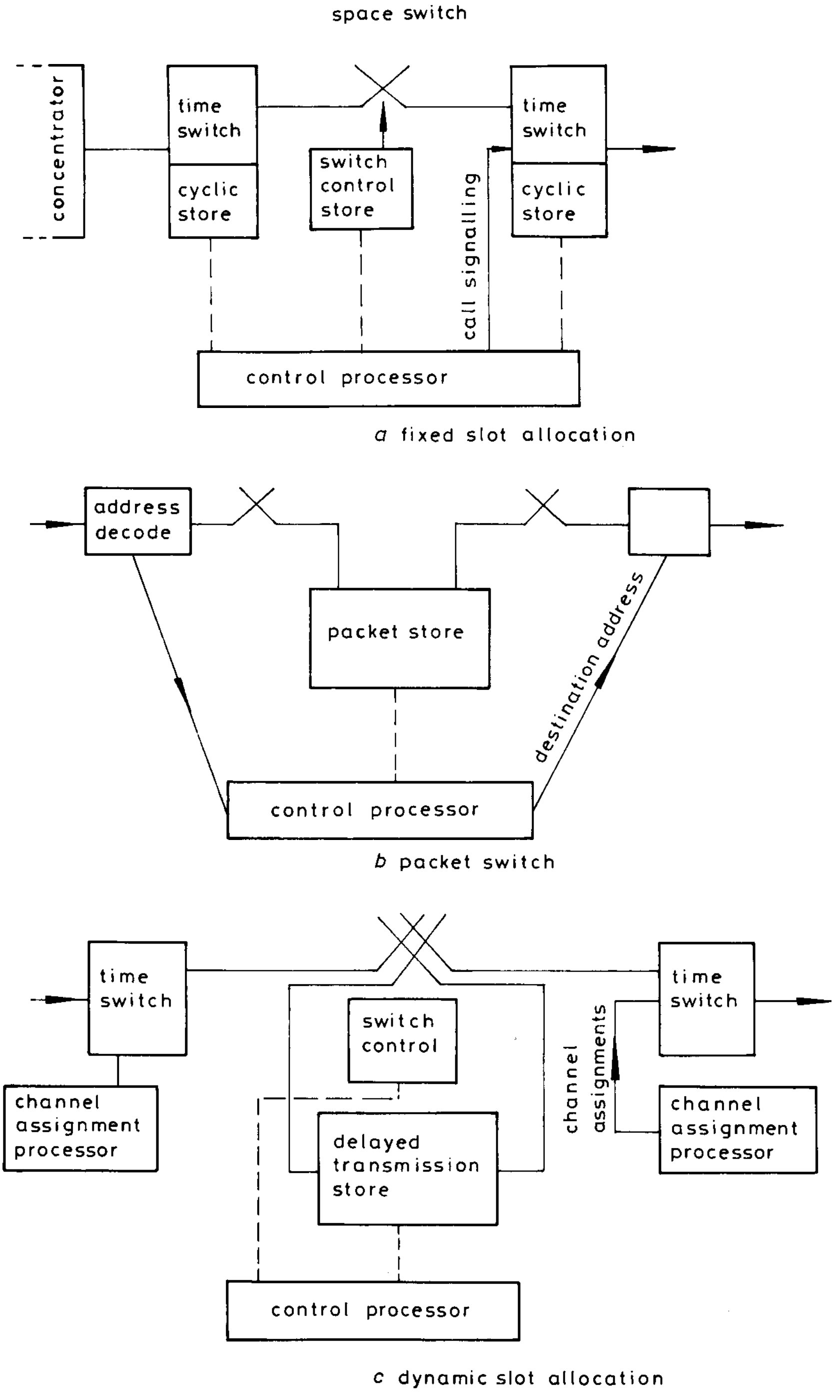


Fig. 8 Techniques for switching multiplexed signals

If multiple slots are allocated at the beginning of the call to give increased bit rate they may arrive at the distant end in a different sequence to that in which they were transmitted owing to different delays in the time switches. The slots then have to be put in the correct sequence before delivery to the terminal.

Since information can be stored in the time switch for a maximum of one frame's duration, the maximum delay through the switch is one frame less one time slot. This is the arrangement shown in Fig. 8*a*, where delay through the switching node is fixed at call setup and need not exceed two frames.

In the packet switched system a complete incoming packet is stored and the destination address processed. When the transmission link on the required outgoing route is about to become free, a search is made for the first packet in the store requiring that route and this packet is then transmitted. The maximum delay suffered by a packet is not normally fixed, although some form of flow control is usually imposed to ensure it does not become excessive.

The dynamic slot allocation has some properties in common with a fixed slot allocation system, but the simple cyclic store used to control the time switch must be replaced by a channel assignment processor. This allocates the time slots in each frame and adds the channel assignment signalling information. As with the fixed slot allocation arrangement, the delay at each time switch is effectively restricted to one frame.

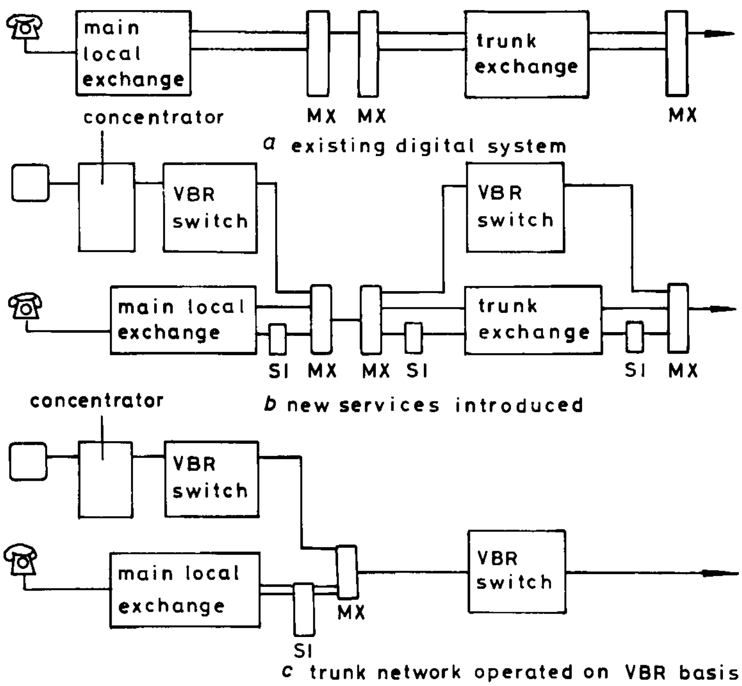


Fig. 9 Evolution from circuit-switched to VBR network
MX = higher-order multiplex terminal
SI = speech interpolation equipment

It should be noted that, by the addition of a delayed transmission store, a synchronous dynamic slot allocation switch can be enhanced to provide an effective packet switching capability. If the instantaneous bit rate required for all the channels exceeds the frame capacity on a particular outgoing multiplexed route it is possible, as an alternative to losing the information by freeze out, to direct the information from some of the channels to the delayed transmission store for sending in a later frame. This is necessary for those telecommunication services in which loss of data cannot be tolerated although some delay is acceptable. The delayed transmission store may also be used for the retransmission of information in which errors have been detected. Again this is achieved at the expense of delay.

In the design of a multiservice variable bit rate switch it is preferable to regard the dynamic slot allocation approach as fundamental and to provide additional facilities for those services that require integrity and accuracy to be traded against delay. This is preferable to starting from a packet switching approach since, once delay is introduced, it is impossible to reverse it even if the service can tolerate a small proportion of errors or loss of information.

7 Network evolution

If it is assumed that the ultimate goal is a multiservice network based on synchronous variable bit rate operation, there are considerable difficulties in achieving that end by evolution from the existing networks. Even if it can be asssumed that:

- the telex network will be absorbed into more advanced text communication services
- the analogue part of the voice network will be replaced by digital equipment
- the packet switched network will be absorbed into the VBR network

it will still be necessary to achieve an orderly evolution from the IDN/ISDN network based on 64 kbit/s channels.

Fig. 9 shows the main stages of a possible evolution plan. In the first stage the the 64 kbit/s channel are retained in the switching nodes but more effective use is made of the transmission capacity by signal processing and time assignment for the transmission links only.

As wideband and other new services are introduced, separate switch units are used to provide the VBR switching capability. Common signalling and common higher-order multiplexes are used for both the circuit switched and VBR parts of the network.

Finally, as the intermediate trunk switches are replaced, VBR operation may be extended to the whole of the main network. Existing circuit switches are retained mainly to switch locally and to concentrate voice traffic on to the main network.

8 Conclusions

Progress in digital technology has provided a sound basis for converting the existing analogue telecommunications network to digital operation. This has been justified on economic grounds for the voice service alone but it has incidentally presented an opportunity for carrying a range of nonvoice services, based on 64 kbit/s channels, on the same network. These ISDN services will come into operation in Britain starting in early 1984.

However, it has become evident that the current embodiment of ISDN is not the last word. Although a unified multiservice network is desirable, further advances are necessary to carry some of the services, particularly those involving wideband (high bit rate) transmission or sporadic interchange of information.

An examination of the fundamental requirements of telecommunications services has shown that the requirements would best be met by a variable bit rate network that would permit the bit rate to be allocated to each channel 'on demand'. This would remove many of the restrictions of the circuit-switched ISDN.

The design of a VBR network should not present any outstanding difficulties. However, it is not just a question of designing the signal processors and switching nodes. The evolution from a circuit switched ISDN network will take considerable planning and foresight.

The rate of progress towards a VBR network will depend on a variety of technical, economic and political factors. It is difficult to forecast progress in view of uncertainties of, for example, market demand for wideband services and the political framework for the establishment of new networks. But at least we think we know the general direction in which it is desirable to progress the evolution of the telecommunications network.

References

1 HARRIS, L.R.:'Introduction to System X', International Switching Symposium, Paris 1979, ISS 79, 201

2 HARDY, J.H.M. and HOPPITT, C.E.: 'Access to the British Telecom ISDN', 6th International Conference on Computer Communication, London, 1982, ICCC 82, 43

3 'CCITT yellow book', Vol. VIII. 2 (1980)

4 RATZ, H.C. and FIELD, J.A.: 'Economic comparison of data communication services', *Computer Networks*, 1980, 4, 143.

5 HOLMES, J.N.: 'A survey of methods for digitally encoding speech signals', IERE Conference on Digital Processing of Signals in Communications, Loughborough, 1981, 291

6 'Open systems interconnection – basic reference model', ISO/TC97/SC16

7 LUDWIG, L.F.: 'Adaptive links – a methodology for dynamic bandwidth allocation', 6th International Conference on Computer Communication, London, 1982, ICCC 82, 718

8 HUGHES, C.J. and ATKINS, J.W. 'Virtual circuit switching for multiservice operation', International Switching Symposium, Paris, 1979, ISS 79, 344

DAP in action

J. Howlett

ICL Group Communication Division, Putney, London

D. Parkinson and J. Sylwestrowicz

DAP Support Unit, Queen Mary College, University of London

Abstract

The paper gives brief summaries of the accounts of 12 different applications of the ICL Distributed Array Processor (DAP) which were described at a symposium held at Queen Mary College on 26 May 1982. These demonstrate the ease with which the machine can be used to attack a wide variety of types of problem and high level of computing power which the novel architecture provides.

1 The DAP Support Unit - DAPSU

The ICL Distributed Array Processor (DAP) has been described in several published papers, for example those by Scarrott and Gostick, respectively, in previous issues of this Journal, which themselves give further references. The essential feature of its architecture, a large number of effectively independent arithmetical/logical units called processing elements (PEs) operating simultaneously under the control of a single master control unit, is so different from that of the classical serial, or von Neumann, machine that its potentialities are not easily appreciated without actual experience. To help build up this experience, and at the same time to make a machine with this novel and powerful architecture widely available to research workers, the DAP Support Unit (DAPSU) was set up at Queen Mary College in the University of London in 1979, financed jointly by ICL, the Computer Board, the Science & Engineering Research Council (SERC) and the Social Sciences Research Council (SSRC). The equipment is the 'standard' DAP with 4096 PEs in a 64 x 64 array, each with a store of 4096 bits, attached to the College's ICL 2980 mainframe computer which forms the 'host' machine. The DAP store will be increased to 16 K (16384) bits per processor in May 1983, giving a total of 8 Mbytes.

A research worker in any of the British universities or polytechnics, or in any of many other research establishments, working in any field of study, can apply for time on the machine and use of the Unit's services; and the application is judged on its merits as a piece of research. There are now about 200 registered users and the machine time is fully booked.

References

1 SCARROTT, G.G.: 'Wind of change', *ICL Tech. J.*, 1979, 1, 35-59
2 GOSTICK, R.W.: 'Software and algorithms for the Distributed Array Processor', *ICL Tech. J.*, 1980, 1, 116-129

2 The DAP in Action symposium

A one-day symposium with this title was held at Queen Mary College on 26 May 1982, made possible by financial support from ICL and the Social Sciences Research Council. Twelve short papers were given, each describing work either in progress or completed and each dealing with a different application of the DAP. The aim was twofold: to give the presenters an opportunity to describe their work to an audience which was well informed on general computational matters but not necessarily so on DAP; and to show the wide range of problem types to which the DAP is actually being applied. The audience was of about 130 people from universities, other higher educational establishments, research centres etc.

The meeting was opened by Dr. J.D. Sylwestrowicz with a short overview of the DAP and DAP FORTRAN, the language in which the great majority of DAP applications are written. The twelve papers followed, in this order:

1 'Monte Carlo methods', J.D. Sylwestrowicz
2 'Explicit finite difference calculations', P. Kirby
3 'From plastic crystals to quantum chromodynamics', G.S. Pawley
4 'Non linear econometric modelling', Y.Y. Chong
5 'Parallel processing in numerical optimisation', L.C.W. Dixon
6 'Networks on DAP', S. McQueen
7 'Variable precision arithmetic on the DAP', S.M. Holmes
8 'Manipulation of map data on the DAP', T.A. Adams
9 'Compiling in parallel', S.R. House
10 'Image processing on the DAP', N.R. Arnot
11 'Some experience on running sea models on the DAP', L.C. Ovadia and D. Owen
12 'Parsing English text on the DAP', D.E. Oldfield

3 Summaries of the papers

The following brief summaries are intended mainly to bring out the variety of the applications described. There is a set of rather fuller summaries in a report of the meeting which is available from the DAPSU, and from which we have quoted in some places. Where possible we give references to papers by the various speakers, where more details can be found. We have been able to consult some of the summaries given here; we hope that we have not misrepresented any of those whom we have not been able to consult – we apologise in advance for any failings here and undertake to publish corrections.

3.1 Monte Carlo methods
J.D. Sylwestrowicz, DAPSU

The Monte Carlo method is a way of attacking complicated computational problems by processes which depend on probabilities and which are essentially dice-throwing techniques. The basic tool in such a method is a generator of random numbers. A large amount of research has been done on random number generation on serial machines, but little or nothing can be found in the literature on how to do this on a parallel machine. At a simple level the problem for DAP is to create 4096 independent random number generators such that the numbers which appear in a single processor are random and simultaneously there is no correlation across the elements. This has been solved rather elegantly by K. Smith of the DAPSU, and implemented by him on DAP. Briefly, the idea is as follows.

A very common way of generating random numbers is to use a multiplicative generator $r_i = a\, r_{i-1} \pmod{m}$

which with suitable values for a, m and r_0 gives a sequence r_i of pseudo random numbers.

If $R_0 = [r_0, r_1, \ldots r_{N-1}]$

it is easily shown the vector recurrence relation

$$R_i = a^N R_{i-1} \pmod{m}$$

produces the same set of numbers as the original generator, but produces N values simultaneously. Thus if the 4096 processing elements of DAP are filled with the successive values given by the first series, it is possible to progress in sets of 4096 at a time by means of this relation with $N = 4096$. All that is needed is a single computation of a^N, and then a parallel multiplication (mod m) produces each set in turn. This gives a parallel generator with the same properties as the generators used in serial environments.

A generator based on this technique is provided in the DAP subroutine library and is probaly the most widely used subroutine on the DAP. Sylwestrowicz has used it in experiments on the Monte Carlo evaluation of single and multiple integrals, in some cases getting speeds of over 100 times that of the ICL 2980.

References

SYLWESTROWICZ, J.D.: '*Parallel processing in statistics*', Proceedings of COMPSTAT 82, CUASSINUS et al. (Eds.), Physica-Verlag, Wien, 1982

SYLWESTROWICZ, J.D.: 'Application of the ICL Distributed Array Processor in econometric computations', *ICL Tech. J.*, 1981, 3, 280-286

3.2 Explicit finite-difference calculations

P. Kirby, UK Atomic Energy Authority, Culham Laboratory, Oxfordshire

The Culham Laboratory is concerned with basic research in plasma physics, underlying the work on the production of energy by controlled nuclear fusion. The large scale experimental facility JET – Joint European Torus – is being built there as a co-operative project by the European countries.

Dr. Kirby gave a brief sketch of methods for the numerical solution of partial differential equations, all of which depended upon replacing the differential system by a discrete system based on a mesh, the values of the function being computed at the mesh nodes. This maps readily on to the DAP array and regions which need to be distinguished, for example, regions of different materials, and therefore of different physical properties, can be defined very easily by the use of logical masks. Various techniques are available for dealing with meshes larger than the 64 x 64 array. Dr. Kirby mentioned a two-dimensional system which he had solved on a 256 x 256 mesh and a three-dimensional system on a 16 x 16 x 16 mesh.

The most interesting of Kirby's applications was the solution of a plasma equilibrium problem for a toroidal configuration, which required the solution of a non-linear partial differential equation of elliptic type in two dimensions. A fixed mesh would not have been suitable for reasons of accuracy; instead a mesh which was related to the contour surfaces of the magnetic field was used. The solution thus had to be computed on a mesh which was neither rectangular nor regular: also, since an iterative process of solution was used, the mesh had to be computed afresh at each iteration. Kirby showed that, contrary to what might have been expected, the entire process was not at all difficult to program for the DAP, and proved very effective. He had done a calculation using parameters relevant to JET and had made a comparative run on the ICL 2976 mainframe machine at Culham and found speed increases of approximately 30 over the 2976. The other problems referred to above also gave speed increases of about 30 over the ICL 2976.

Dr. Kirby gave several pieces of general advice on how to tackle problems of this type on the DAP. One in particular – 'first organise the data, then do the arithmetic' – probably applies equally to the use of serial and vector machines.

References

KIRBY, P.: 'A moving-mesh plasma equilibrium problem on the ICL Distributed Array Processor', *ICL Tech. J.*, 1981, 2, 403-424

WEBB, S.J.: 'Solution of elliptic partial differential equations on the ICL Distributed Array Processor', *ICL Tech. J.*, 1980, 2, 175-190

3.3 From plastic crystals to quantum chromodynamics
G.S. Pawley, University of Edinburgh

The physical problems being studied by Dr. Pawley and his colleagues at Edinburgh are all concerned with the behaviour of systems which can be represented on lattices, such as a system of particles interacting under mutual forces, or simply a system for which the field equations are known. He described their use of the DAP in the application of two particular techniques for attacking such problems, Molecular Dynamics and Monte Carlo.

In Molecular Dynamics calculations a model of the system is set up and is allowed to develop as a function of time as the particles interact under the appropriate force law. Pawley and his colleagues have studied in detail the transition from plastic to crystalline state for sulphur hexafluoride (SF_6). With DAP they were able to work with samples of 4096 molecules, which allowed physically realistic models to be set up and followed. A significantly smaller sample size, for example 500 molecules, would have been inadequate. They were able to follow the formation of crystals as the initially plastic form was cooled from 80 K to 25 K.

They found that they could use the DAP with maximum efficiency in this work, that is, that they could keep all 4096 PEs working all the time. In studies of the liquid/plastic phase transition, however, they could not achieve such a high efficiency, essentially because of the greater mobility of the molecules. The use of logical masks was a powerful and effective tool in keeping track of the occupancy of the cells into which the physical space was divided. Even with the reduction in efficiency they reckoned that DAP was an order of magnitude more cost-effective than, say, CRAY-1 for this work.

Monte Carlo methods are being used at Edinburgh for studying other phase transitions. They are using the three-dimensional Ising model for this work and also for studies of ferromagnetics and of glasses.

Dr. Pawley said that their most important application of Monte Carlo was in quantum chromodynamics, for calculations on nuclear stuctures using the quark model. This involved the processing of matrices – the SU(3) matrices of nuclear theory – at each node of a four-dimensional grid of size 8 x 8 x 8 x 8. These are very large scale calculations indeed, of fundamental importance in nuclear structure theory. Pawley estimates that about 2000 h of DAP time are needed to get reliable results, for example, estimates of meson and baryon masses. He was able to announce that the Science & Engineering Research Council (SERC) had agreed to help finance the purchase of a DAP for their use.

[Note: the machine was installed in May 1982 and is hosted by the dual 2972 system at the Edinburgh Regional Computer Centre.]

References

BOWLER, K.C., PAWLEY, G.S., PENDLETON, B.J. and WALLACE, D.J.: *Phys Rev. Lett. A.*, 1981, 104B, 481-484

PAWLEY, G.S. and THOMAS, G.W.: *Phys. Rev. Lett.*, 1982, 48, 410 and *New Scientist*, 1 April 1982, 20

3.4 Nonlinear econometric modelling
Y.Y. Chong, DAPSU

Dr. Chong characterised research in econometrics as being 'multivariate and intensive in the use of matrix operations'. More specifically:

1 Econometric nonlinear estimation usually involves maximising a nonlinear scalar function of matrices, with much use of inversion and of complicated manipulation of matrices of large order.
2 Monte Carlo simulation is used to study the properties of econometric estimators

and to model the behaviour of econometric systems in which there are a large number of participants.

3 Investigation of finite sample distributions of estimation procedures creates major demands for computing power.

She described the work she had done at the London School of Economics on estimation of large scale econometric models. This was based on the maximisation of the likelihood function in circumstances in which there were a large number, of the order of thousands, of observations. The process involved heavy matrix computations and also the implementation of analytic differentiation operations. One of the advantages of using the DAP was that this made it possible to perform calculations with respect to many observations in parallel.

Dr. Chong reported the results of comparative calculations, using a parallel process on DAP and a serial process on ICL 2980 and CDC 7600. For large systems (about 4000 observations) DAP was about 150 times faster than the 2980 and about 30 times faster than the 7600; but for small systems (less than 128 observations) the serial version on the 7600 was fastest. This is a not uncommon experience, that the advantage of DAP over serial machines is greatest for the largest scale computations.

References

CHONG, Y.Y.: 'Comparative methods of computing maximum likelihood estimates for non-linear econometric systems', Ph.D. thesis, London University, 1982.
(also SYLWESTROWICZ, op.cit.)

3.5 Parallel processing in numerical optimisation
L.C.W. Dixon, Hatfield Polytechnic

Problems which require one to find the maximum or minimum values of nonlinear functions of possibly very many variables are encountered in many fields, for example in theoretical physics, in engineering design and in statistical estimation. There is a considerable literature on the methods which have been developed to attack this problem. At Hatfield Polytechnic there is a Numerical Optimisation Centre where comparative studies of methods have been going on for several years, and which are continuing.

Dr. Dixon presented results of studies of the performance of algorithms for which parallel versions have been developed at Hatfield and run on DAP, comparing the timings for these with those for serial versions run on the Polytechnic's DEC 1091 machine. The parallel codes were:

1 Newton-Raphson iteration. The process is one of moving from point to point in the multidimensional space of the variables of the function, choosing the direction and the step length so as to reduce the value of the function (in the case of a minimisation) at each step. This requires the calculation of the gradient vector and the Hessian matrix (the matrix of second derivatives) at each step.

2 An algorithm due to W.L. Price called Controlled Random Search (CRS). As the name implies, this involves calculating the values of the function at a large number of randomly chosen points in the space of the variables, using these values to restrict the region in which the maxima or minima are likely to be found, and repeating the process.

Published standard serial routines were chosen for comparison: the Newton-Raphson method from the NAG library, the variable metric method from the NOC Optima library (routine OPVM) and the conjugate gradient method from the Harwell library (routine VA08A). The comparisons were made with functions which have become standard tests for optimisation methods, including a number of polynomials in 64 variables.

The results, as might be expected, were very varied and the relative performances depended greatly on both the method and the test function. For the Newton-Raphson method the parallel version on DAP always outperformed the serial version and the variable metric method on the DEC 1091; the speed advantage ranged from a marginal increase to about 50:1, depending on the form of the function, the choice of starting vector and whether the gradient vector and Hessian matrix were calculated from analytic expressions or as numerical approximations. The conjugate gradient method was faster in cases where the function was especially symmetric, in other cases the parallel method was up to about 15 times as fast. Different functions were used for the comparisons for the Price algorithm, most of which had more than one global minimum; the speed advantage of DAP here ranged from 3.2:1 to 68:1, the highest ratio being observed in the case of a function having 18 global minima.

Dr. Dixon's comment was that 'though these results are very encouraging, they do emphasise the difficulties to be anticipated when comparing solutions on different architectures with different algorithms.'

References

DIXON, L.C.W. and PATEL, K.D.: 'The place of parallel computation in numerical optimisation IV: parallel algorithms for non-linear optimisation', TR 125, The Numerical Optimisation Centre, The Hatfield Polytechnic, Feb. 1982

DUCKSBURY, P.G.: 'The implementation of a parallel version of Price's (CRS) algorithm on an ICL DAP, TR 127, The Numerical Optimisation Centre, The Hatfield Polytechnic, May 1982

3.6 Networks on DAP
S. McQueen, DAPSU

Network problems appear in many different forms: they can arise from actual physical networks such as gas or power distributions or from more conceptual networks derived from formulating a problem in graph-theoretic terms. Dr. McQueen is particularly interested in the study of flow through physical networks such as the gas pipes under the streets of London.

An obvious property of a distribution network is that it does not form a regular mesh-connected system. It is therefore necessary to find a way of mapping the irregular connections on to the regular mesh of the DAP processors. DAP provides a much richer choice of methods of representation of data structures than does a serial machine; a description of the topology of a network can be stored either as pairs of integers, with each pair representing one link, or as a binary matrix, the adjacency matrix, in which each row (of 0s and 1s) gives the links from one junction to all others. It appears that the adjacency matrix is the better method for a medium sized matrix of up to, say, 250 junctions; it is less suitable for large networks because of the amount of storage it requires.

A point emphasised by McQueen was that the new operations provided by DAP, including its ability to handle all the elements of a matrix simultaneously and to operate as a content-addressable machine, mean that an algorithm which is optimum for a serial machine is not necessarily so for DAP. In this context he had found that the Dijkstra algorithm for finding the shortest path through a network was less efficient on DAP than that of Bellman and Ford – the reverse of what is found on serial machines. He commented also that a parallel machine such as DAP offers a much wider choice of methods for moving data items around, and that this is often an important consideration in assessing the cost of a computation.

This work is essentially in the field of sparse matrix manipulation, and therefore results and experience gained are of potentially very wide application.

3.6 Variable precision arithmetic on DAP
S.M. Holmes, DAPSU

The DAP being made up of 1-bit processors, all arithmetic is done by software; there is therefore no 'natural' length, in bits, for a number, and while many of the standard routines work with 32-bit numbers a user can elect to work with any length which suits the problem. Dr. Holmes's paper was concerned with numbers of very high precision (many thousands of bits) and in particular with applications in number theory, specifically to the finding of large primes.

Numbers of up to 1000 bits (about 300 decimal digits) can be handled fairly straightforwardly, as three such numbers can be held in the standard 4096-bit store of each PE and still leave room for code. Significantly larger numbers have to have their binary digits spread over several store planes: thus 32 planes could be used to hold a single number of 131072 bits.

When working with numbers of very great length it becomes necessary to look critically at the details of the basic arithmetical processes, in particular at multiplication because the time for this becomes unacceptably long for such numbers. The method used on DAP by Holmes and his colleagues involves:

1 splitting the numbers into groups of bits, so that a number can be regarded as a vector
2 regarding multiplication of a pair of numbers as a cyclic convolution of the two vectors

3 using results from transform theory, in particular the Fermat number transform, to improve the efficiency of the convolution process.

Theoretical interest in large primes centres mainly on the Mersenne primes, that is, primes of the form $M_p = 2^p - 1$ where p is prime. At any time the largest known prime has traditionally been a Mersenne prime because of the existence of an efficient test for primalty, the Lucas-Lehmer test. At the time of Dr. Holmes's presentation of the largest value of p for which M_p was known to be prime was 44497, the complete set of values for p being

2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521, 607, 1279, 2203, 2281, 3217, 4253, 4423, 9689, 9941, 11213, 19937, 21701, 23209, 44497

The DAP program was used to check these, using the Lucas-Lehmer test, and to extend the search. Holmes was able to say that no value of p up to 62975 gave a Mersenne prime. Since then the value p = 86423 has been found by Slowinski at Cray Research, Chippawa Falls, Wisconsin, USA. This has been checked by the DAP program, which is continuing the search up to 128511.

References

POMERANCE, C.: 'The search for prime numbers', *Scientific American*, December 1982
SLOWINSKI, D.: (private communication)

3.8 Manipulation of map data on the DAP

T.A. Adams, Birkbeck College, University of London

The size of the problem faced by the Ordnance Survey, the body responsible for the surveying and mapping of Great Britain, is shown by the fact that it has to produce and keep up-to-date about 220,000 basic scale maps, those being on the largest scale on which the whole country is mapped.

For the past ten years the OS has been working on the transfer of the map archive from printed sheets to computer-accessible form, so as to gain the obvious advantages which this form will bring. The process from the start has involved manual operation of digitising equipment, which although effective has proved too slow to be acceptable as a long-term solution. More recently methods involving raster scanners have been developed; in these an optical reading head scans the sheet in a series of lines and converts the printed information into a binary coded form. The output from such a scan is a matrix of small picture elements ('pixels') with binary information associated with each. This raster image for a single sheet can be a matrix of 4096 x 4096 pixels and can be produced in about 30 min; this compares with the possible 40 h needed for manual digitisation.

The need is first to store these images and then to be able to manipulate them in many different ways: for example, to update the information as new observations are made on the ground – new houses, new roads, new reservoirs; or to extract specified topological features such as roads or waterways. Dr. Adams and his colleagues are studying the application of DAP to this work. An important step has been the design of a hierarchical structure for the data, which makes it possible to control the resolution of each pixel and hence the size of the image held in the DAP store. This involves the concept of Level in the structure. The lowest level is the basic 4096 x 4096 set, called Level 12 because $2^{12} = 4096$; this could be stored in 4096 DAP planes. A single 64 x 64 plane is Level ($2^6 = 64$) and a single pixel at this level can point to a complete plane at Level 12. The scheme makes it possibe to refer quickly to information at different degrees of resolution and also to compress the data. The latter is possible because if any pixel at Level 6 is unset, indicating that the corresponding Level 12 plane holds no information, that plane can be omitted.

The processing of the information is clearly mainly a matter of logical, masking and shifting operations, all of which are very fast and efficient on DAP. The work is still in progress, but the experience so far is encouraging.

References

ADAMS, T.A.: 'Raster data structures and topographical data', Ph.D. thesis, University of Durham, 1982

ADAMS, T.A. and SMITH, K.A.: 'The manipulation of raster-based topographical data on a parallel processor', Proceeding of IEEE Conference on Pattern Recognition and Image Processing. Las Vegas, June 1982

3.9 Compiling in parallel

S. House, University of Kent

The process of compiling a program written in a high-level language involves the scanning of the source code character-by-character and the recognition of the meaning of each character and its relation to others. In his use of the DAP for this process House regards the machine as a vector machine with 4096 PEs arranged in a linear sequence and each connected to its neighbours on the left and the right. He inputs the source text so that a single character is associated with each PE; therefore a 'window' of 4096 characters can be processed in parallel.

Recent advances in compiler design have advocated multiple passes. This simplifies the design of the compiler because each pass can deal with a single task or at worst only a very few tasks. The possibility of parallel scanning means that many passes can be allowed and the amount to be done at each pass correspondingly reduced. The powerful associative properties of DAP can be exploited in, for example, finding all the occurrences of a given character, or type of character, in one operation. Object code can be generated very efficiently; for example, that for multiplication can be written simultaneously to every occurrence of the 'multiply' operator. Also considerable and effective use can be made of logical masks, for example to flag errors or to broadcast values to selected PEs.

As a first experiment, House has used DAP to compile a 'mini-language' for a hypothetical stack machine. His conclusion is that parallel compilation is an efficient process which is not difficult to understand, and that it gives insight into the expression and formulation of parallel algorithms and imposes a valuable discipline on the structure of these.

He made the comment that 'the main limitation to the SIMD architecture must always be the shortage of processors'. He added that there was no reason to suppose that the problems to which this gave rise would not be solved, just as corresponding problems for von Neumann machines had been solved. The culmination would be 'a high level programming language for the SIMD model of computing which is independent of the number and nature of PEs and the interconnections between them'.

Reference

HOUSE, S.R.: 'Compiling in parallel', In *'CONPAR 81'*, HANDLER, W., Springer, 1981, 298-313

3.10 Image processing on the DAP

N.R. Arnot, Queen Elizabeth College, University of London

The following is the text of a note supplied by the speaker.

An image is a large two-dimensional matrix of numbers representing measurements of some physical quantity (for example light intensity) sampled on a regular grid in some co-ordinate system. Image processing (IP) involves taking one or more images as input and deriving from these a more directly interpretable image. For example, in computerised tomography one reconstructs a picture of a 'slice' through an object (such as a hospital patient!) from projections obtained by passing a beam of X-rays through the object at a series of angles. The reconstruction shows details too fine or faint to be visible on a conventional X-ray micrograph, and the invention of such 'body-scanner' machines caused a minor revolution in clinical diagnosis and treatment.

Even when the required processing is relatively simple, IP can present a severe computational load simply because of the volume of data – images of several thousand points square are not uncommon. Fortunately, analysis on the basic IP operations reveals that the vast majority can be broken into parallel processing of suitably sized subregions of the input, and therefore these are ideally suited to the DAP. The need to partition the input and the consequential edge effects can add to program complexity (unnecessarily – a compiler could accomplish this for one), but does not contribute significantly to runtime overheads. Geometric transformations (rotation, enlargement etc.) are unfortunately not so easily parallelised, but some parallel algorithms do exist for special cases which prove to be those most frequently encountered in practice. Fast transforms, such as the fast Fourier transform (FFT) have a highly parallel structure, and subregion operations, normally involving inefficient conditional tests in loops, are efficiently handled in terms of logical masks, both for geometrically and data-derived regions.

We have found that even where we have to use floating-point data the DAP can be extremely fast; a 64-point square FFT takes 20 ms compared with 65 ms for a hand-coded assembler routine on a CDC7600. Where one can work throughout on the short-length integers of which most input images are composed the DAP can be several times faster still, and for purely binary images the DAP can outperform any other general-purpose computer. This arises because the DAP performs arithmetic in a bit-serial manner under software control.

There are, however, two drawbacks to the use of the DAP. First, both input and output are usually images, and often (although by no means always) one finds that the (real) processing time falls well below the (real) I/O time – that is to say, the DAP converts a CP-limited process into an I/O-bound one. Secondly, much image processing work, and especially research work, is essentially interactive in nature. Most IP systems consist of a powerful minicomputer connected to one or more very fast image display terminals, and the researcher will decide what operation is required on the basis of a display of a previous intermediate result. This is clearly impracticable with the current DAP, which is configured as a batch-only resource of a large remote mainframe. Nevertheless the DAP architecture has immense potential and the current system brings within reach techniques which are too CP intensive to use elsewhere, as well as being useful for non interactive 'production' work.

3.11 Some experiences of running sea models on the DAP
D.C. Ovadia and A. Owen, Natural Environment Research Council

A note by the speakers gives the background to this work as follows:

'The Institute of Oceanographic Sciences, a component body of the Natural Environment Research Council, have a long-term research interest in the prediction of sea surface elevations using numerical hydrodynamical modelling. Sea models solve the equations expressing the conservation of mass (of water) and momentum numerically in one, two or three dimensions subject to specified boundary conditions. Closed boundaries represent a coastline and along such a boundary the normal component of velocity is specified as zero. At open boundaries the sea surface elevation and/or the normal component of current are specified, usually in a form derived from the major tidal constituents. Throughout the model the stress at the sea surface is specified, this being zero in a purely tidal model, or some value derived from a meteorological input in a surge model'.

The partial differential equations of mass and momentum conservation are solved numerically by finite-difference methods on a regular grid. Ovadia and Owen reported on a series of studies which they had made to explore the suitability of DAP for this work, in which models were progressively refined to include more physical processes and more geometric complications, such as islands and shallow-water regions which could dry out at low tide. They gave timings for comparative runs on DAP and CRAY-1. These showed that the relative performance on this series of problems ranged from about 5.1 in favour of CRAY for the simplest and most straightforward situations to about 2:1 in favour of DAP in the most complicated. This is consistent with experience in entirely different fields, that DAP shows up to greatest advantage over serial machines in the largest scale and most complicated problems.

Ovadia and Owens's main conclusions, in their own words, are as follows: 'DAP-FORTRAN is very easy to learn and is more concise than standard FORTRAN. The ability to access individual bits makes the DAP very useful for logical operations. Since each individual processor on DAP is relatively slow, to obtain maximum efficiency all 4096 processors must be used.

Numerical sea modelling on the DAP, using finite differences, has many advantages. The inclusion of drying is trivial, as is the representation of irregular coastlines and islands. Upstream finite differences are particularly suited to the DAP because the required logic is very easily handled. Physically it is possible to identify each of the DAP processors with an individual grid point in a two-dimensional sea model. Such an identification can simplify the coding and debugging of sea models. The extension of two-dimensional models to three is straightforward.

There are some disadvantages, however. If a model requires a grid larger than 64 x 64, the model grid must be divided up into subsections of 64 x 64 and the model calculations performed on each subsection in turn. Although the coding is more involved the efficiency of the DAP is not reduced, provided most of the processors in each subsection are utilised. The most inefficient situation would be a model requiring a grid of 65 x 65. In such a situation consideration should be given to reducing the model grid to 64 x 64. If this is not possible the grid could be increased to 126 x 126 (allowing for overlaps of subsections) without any computational overheads.

I/O is not possible from the DAP directly; data must first be converted back to the host memory store. This conversion carries a computational overhead, but it was shown that this would probably not be significant in most modelling situations, except perhaps in storm-surge modelling work'.

References

DAVIES, A.M. and OWEN, A.: 'Three dimensional numerical sea model using the Galerkin method with a polynomial basis set', *Appl. Math. Modelling*, 1979, 3, 421-428

FLATHER, R.A. and HEAPS, N.S.: 'Tidal computations for Morecambe Bay', Geophys. J. R. Astron. Soc., 1976, 42, 489-517

HEAPS, N.S.: 'On the numerical solution of the three-dimensional hydrodynamical equations for tides and storm surges', *Mem. Soc. R. Sc. Liege*, Ser. 6, 1972, 2, 143-180
143-180

3.12 Parsing English text on the DAP
D.E. Oldfield, University of Kent

The immediate aim of this work is to produce a DAP program which will abstract large and complicated documents, for example, Acts of Parliament. Oldfield is in fact using the 1968 Rent Act as a test bed. The broad strategy is to find what he calls kernel sentences, then to find the main part of the subject noun phrase and the main verb phrase in each sentence and finally to build up the abstract from this material,

As House does in his use of the DAP for compilation, Oldfield uses the machine in long-vector mode with the PEs considered as linked in a linear chain of 4096 elements. Test is input in blocks of 4096 items – words or punctuation symbols – and stored in a layer made up of consecutive planes, one item to each PE. A dictionary of function words is held in the store, these being the words most commonly used in the language, such as 'the', 'and', 'of', and which are almost independent of the subject matter of the text; information about its syntactic category is stored with each word. House uses a dictionary of about 110 function words. The occurrence of each function word in the block of text is established by broadcasting each word in turn to the whole block and noting which PEs, if any, contain the matching word. The syntactic categories of the remaining words of the text are then established by examining their positions in relation to the known function words. This involves shifting operations and comparisons with selected parts of the syntactic information in the function word dictionary.

The whole process is very much geared to DAP's parallel organisation and to its scope for logical operations. Oldfield reported that in his experiments with text from the 1968 Rent Act he had been able to identify correctly the main verb in about 80% of the sentences and to produce smoothly reading abstracts. The timing, which was completely independent of the content of the document, was 442 ms for each block of 4096 text items made up of 115 ms for dictionary lookup, 108 ms for syntactic processing and 219 ms for formatting for output. These times show that DAP is fast enough to deal with more complicated grammatical rules and hence to identify more kernel sentences. They show also that it would be practical to develop the system into an interactive syntax checker which could form the basis of an information retrieval system for large documents.

Oldfield commented that the entire program had been written in DAP-FORTRAN apart from the I/O routines which were written in 2900 FORTRAN. He had found no need to go to the assembly language APAL either for facilities or for speed.

4 Summary and conclusions

We must emphasise that the 12 applications of the DAP which we have described in this paper are in no sense artificial test problems but are in every sense real problems arising from real situations. The presenters are all active research workers whose concern is with the problems which they described and not primarily with the computer; they concern themselves with the DAP only because they see this as a powerful weapon in their attack on the problems.

The great breadth of the range of types of application reported at the symposium is very evident: from the heavily numerically oriented work of Chong on econometric modelling, Pawley on phase-change calculations and Ovadia and Owen on sea models to the entirely non numerical work of House on compilation and Oldfield on text analysis. There is great variation also in the extent to which the problems might at first sight be expected to map on to the DAP architecture. This architecture is clearly well suited to Monte Carlo calculations, in which a single process is to be applied to a large number of independent variables; to matrix operations; and to finite difference systems set up on a mesh. Furthermore, as in the case of Kirby's work in plasma physics, it turns out to be unexpectedly well suited to situations in which one has to use a non regular and varying mesh. But there is no immediately evident parallelism or regularity in the network systems studied by McQueen, the compilation work of House or the text analysis of Oldfield. And the value of the ability to work with numbers of any desired length is shown in the work on map data processing by Adams, on image processing by Arnot and – perhaps a rather esoteric application – on very large primes by Holmes.

The sheer weight of computational power that the DAP can deliver is equally evident from these applications. Comparisons between different computers, especially of advanced design, are notoriously difficult to make. Several of the speakers were able to report results of comparative runs on DAP and machines acknowledged to be in the highest class of computing power; as was to be expected the relative performance varied widely with the type and scale of the problem, but in all cases the advantage of DAP increased steadily as the scale and complexity of the problem increased. There were certainly cases in which DAP could be rated as the world's most powerful computer. Equally important, several speakers commented on the ease with which the potential power could be achieved in practice and on the high cost-effectiveness of DAP compared with other machines.