Report Series
No 3

Terminal Compatibility

November 1977

The Butler Cox Foundation

Report Series No. 3

TERMINAL COMPATIBILITY

NOVEMBER 1977

A. Gunton

Butler Cox & Partners Limited
The Press Centre
11th Floor
76 Shoe Lane
London
EC4A 3JB

TABLE OF CONTENTS

I. INTRODUCTION

Virtually every large computer user is aware of the problem of terminal compatibility, either as a result of restrictions on the range of equipment which can be used, or in terms of real difficulties in tracing and correcting system faults arising from equipment incompatibility.

There are a number of reasons for incompatibility between the terminals in common use today and the computer systems with which they communicate. These are partly historical, arising from the way the terminal industry has developed; partly commercial, arising from practices adopted by suppliers to protect their market; and partly technological, arising from fundamental differences in the way that terminals operate.

But the situation confronting users appears to be becoming more confusing. A new generation of terminals is becoming available, new networking products are being announced by the mainframe suppliers, and the European PTTs are talking about — and in some cases implementing — plans for public data networks. Separately they offer advantages over their predecessors, but together they compound the compatibility problem and introduce new uncertainties. Further ahead there is the prospect of terminal applications which will add a new dimension to an already complex problem.

On the other hand, many may hope that cheap microprocessors will help to solve the incompatibility problem by increasing the sheer processing power of terminals. Others may point to the increased effectiveness of international authorities such as CCITT in establishing machine-independent standards as a positive sign. Clearly standards are the key to any lasting solution to the problem.

Terminals will be fundamental to the future business strategies of many users. Therefore data processing managers and specialists alike need to be aware of the full extent of the terminal compatibility problem and its implications for their development plans. The purpose of this report is to help them to such an awareness, by providing a clear picture of the root causes of incompatibility, assessing how far they are capable of solution and examining the practical options which are open to the user.

II. THE COMPATIBILITY PROBLEM

This section outlines the important aspects of terminal/computer interfacing and describes some typical symptoms of the compatibility problem.

A Elements Of The Terminal/Computer Interface

In the early stages of terminal use, it was enough to know the rules for physical connection of the terminal to the computer channel or, for those using telecommunications lines, to the line terminating equipment (normally a modem). In some cases, there might additionally have been some simple logic for error detection, such as the echoplex arrangement found on many teletype-compatible terminals.

With increasing line speeds came the introduction of buffered terminals using synchronous transmission techniques. Character-by-character transfer of data was replaced by exchange of message blocks, and more sophisticated means of signalling and error correction were built into the protocol governing the exchange of messages. It also became possible for a transmission line to be shared by a number of terminals. Again, logic built into the protocol, in this case poll/select logic, controlled access to the shared resource.

Most medium and high-speed terminals in use today use a protocol of this type. IBM's Binary Synchronous Communications (BSC) is dominant by virtue of IBM's dominance of the mainframe computer market, and all other major manufacturers offer an alternative, most deriving from the ISO and ECMA standards for what are termed 'basic mode' control procedures. Appendix 1 is a review of international standards for data communication.

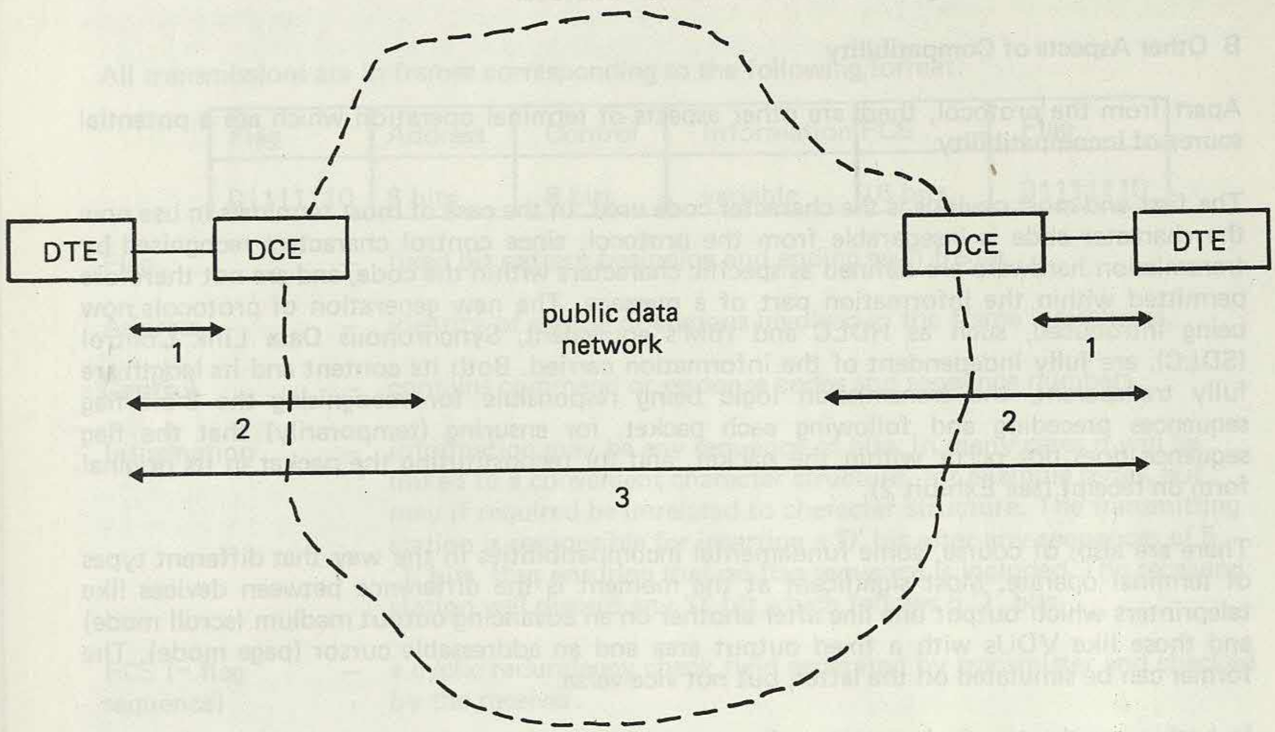
A further level of complexity is introduced in data networks using intelligent computers at the nodes. The nodal computers need a protocol for signalling to one another as they route information *through* the network. This is in addition to the protocol needed for signalling right *across* the network, between the communicating terminal and its host computer system. The three levels of protocol defined in the CCITT X.25 recommendation for public data networks, shown in Exhibit 1, illustrate the structure which has developed:

- At the lowest level, the electrical interface between the device and the network terminating equipment is defined. This corresponds to the V.24 interface which is widely used today for data terminals, but provides for additional signalling.
- The link protocol, at the next level, defines the format of message blocks now termed packets, exchanged between nodes. The link protocol is bit-oriented, enabling any code or indeed pure binary information to be transmitted. Although there are still minor differences, we anticipate that the X.25 link protocol will eventually coincide with ISO's High Level Data Link Control (HDLC) standard.
- At the highest level, procedures for establishing and clearing connections and for exchanging information between communicating devices are defined.

A protocol such as BSC or one conforming to the X.25 recommendation thus provides for the transport of information between a terminal and a computer system. This is not, of course, the whole story, since in most cases the application program within the computer system requires the information to be presented in a certain way, and the terminal requires device control information which may differ from one type of terminal to another and from one supplier to another.

Thus BSC is not a complete definition of a protocol — complete, that is, in the sense that it

Exhibit 1 Levels of protocol in CCITT's draft recommendation X.25



DTE = Data Terminal Equipment
DCE = Data Circuit-terminating Equipment

level 1 — The physical, electrical, functional and procedural characteristics to establish, maintain and disconnect the physical link between the DTE and the DCE.

level 2 — The link access procedure for data interchange across the link between the DTE and the DCE.

level 3 — The packet format and control procedures for the exchange of packets containing control information and user data between the DTE and the DCE.

could be given to an alternative terminal supplier as a full specification of requirements. To achieve this the device must also be defined. For example, 2780 BSC is used for remote job entry, 2260 BSC for visual displays, and so on. In addition to the line protocol these specify the rules governing the internal format of information blocks, and those governing the operation of the terminal device i.e. the application level protocol. All of this is required for a complete definition of a given terminal/computer interface.

B Other Aspects of Compatibility

Apart from the protocol, there are other aspects of terminal operation which are a potential source of incompatibility.

The first and most obvious is the character code used. In the case of most terminals in use now the character code is inseparable from the protocol, since control characters recognised by transmission hardware are defined as specific characters within the code, and are not therefore permitted within the information part of a message. The new generation of protocols now being introduced, such as HDLC and IBM's equivalent, Synchronous Data Link Control (SDLC), are fully independent of the information carried. Both its content and its length are fully transparent, the transmission logic being responsible for recognising the 8-bit flag sequences preceding and following each packet, for ensuring (temporarily) that the flag sequence does not occur within the packet, and for reconstituting the packet in its original form on receipt (see Exhibit 2).

There are also, of course, some fundamental incompatibilities in the way that different types of terminal operate. Most significant at the moment is the difference between devices like teleprinters which output one line after another on an advancing output medium (scroll mode) and those like VDUs with a fixed output area and an addressable cursor (page mode). The former can be simulated on the latter, but not vice versa.

In both cases the terminal must permit as many or more characters in each line as the application driving it expects, and in the case of page mode it must also permit as many or more lines in each page. Sometimes this will be a virtual line or page size, if the terminal maps between the terminal seen by the application and the physical terminal seen by the operator. The Burroughs TD700 VDU, for example, has a 256 character display but a 1024 character buffer, which the operator can inspect 256 characters at a time.

These are the major sources of incompatibility at the present time. It is conceivable that new applications and new technology will introduce more – a possibility which is discussed in Section IV.

C Some User Experiences

Occasionally users experience compatibility problems even when using standard equipment from a single supplier, for example when national modem standards have been overlooked, or when an unusual combination of hardware is being used. Because there is no division of maintenance responsibility, such problems generally tend to be shortlived. However, it is where more than one hardware supplier is involved that most compatibility problems arise, and these problems can often be difficult both to trace and to remedy. Three typical user experiences are described below.

1. **A Service Bureau**
   This bureau serves the needs both of its own corporation and of outside customers, the business being split approximately 50/50. The majority of its services are on a remote job entry (RJE) basis, using IBM mainframe systems. The RJE terminals are from a variety of suppliers, ranging in size from stand-alone devices consisting of input device and printer, up to medium-scale systems such as an IBM 370/145.

Exhibit 2 Frame structure and basic principles of bit-oriented protocols

All transmissions are in frames corresponding to the following format:

Flag	Address	Control	Information	FCS	Flag
01111110	8 bits	8 bits	variable	16 bits	01111110

Flag — fixed bit pattern beginning and ending each frame.

Address — identity of station or stations involved in the frame interchange.

Control — contains command or response codes and sequence numbers.

Information — information may be any sequence of bits. In many cases it will be linked to a convenient character structure, for example bytes, but may if required be unrelated to character structure. The transmitting station is responsible for inserting a '0' bit after any sequences of 5 '1' bits, thus ensuring that no flag sequence is included. The receiving station will discard any '0' bit which follows 5 '1' bits.

FCS (= flag sequence) — a cyclic redundancy check field generated by transmitter and checked by the receiver.

Scanning of the information field and generation of flag and FCS fields will normally be performed by hardware.

Some special supervisory frames will contain no information field.

From the beginning all the RJE terminals were operated as standard IBM terminals, and difficulties were experienced in making the non-IBM terminals 'look' right. In many cases this was because the suppliers were too optimistic about their ability to produce and maintain the emulation programs. Both the software effort and the machine power required were often under-estimated, first for 2780 emulation and then more recently for the more sophisticated HASP multi-leaving protocol. Often customers grew to live with the anomalies these contained, either by-passing them or adapting their procedures to suit. Errors were still being discovered years after installation.

There were no problems with the mainframe software, both because considerable in-house expertise had been built up and because a stable environment had been achieved.

Today the service bureau feels some concern over whether this situation can be maintained as IBM encourages its users to move on to its new communications software.

2. An Intelligent Terminal Network

This user chose an independent (but large) supplier of intelligent terminals for a network centred on its IBM mainframe. Initially the terminals captured data off-line on floppy discs which were then transported to the central site. Later they were linked on-line to the IBM system, using a remote batch interface. This phase took between 2 and 3 times longer than planned to install; mainly because of inadequate support from the supplier. The supplier had contracted to produce all the software to the user's specification, including a 2780 emulator, and seriously under-estimated the effort required. (The emulator still has one or two bugs).

One problem which took several weeks to resolve illustrates the difficulties of a multi-supplier system. Following delivery of a new operating system for the terminals, the system ran slower than before. Extensive software checks failed to reveal the problem. The error, which was data dependent and thus apparently intermittent, was eventually traced to the modem interface. The modems also were from an independent supplier.

3. Distributed Mini-Computers

This user operates a policy of distributed processing using DEC mini-computers. Terminals from two independent suppliers were chosen, mainly because of the willingness of the suppliers concerned to tailor the terminals to the user's specific requirements. The interface between the terminals and the DEC minis was relatively straightforward, using teletype-compatible protocol.

Interfacing with the Honeywell mainframe was less easy. The Datanet front-end processor expected all terminals on medium-speed lines (which these were) to use Honeywell's VIP/7700 protocol. A solution was found by 'short-circuiting' the VIP/7700 and teletype interface so that the latter drove both slow and medium speed lines. This change was restricted to the Datanet and was transparent to the mainframe. There were also minor problems with the electrical interface again resolved by means of ad hoc solutions.

The TTY protocol adopted is only capable of limited error checking, and systems have to be designed to take account of this limitation. For the same reason it is regarded as being only a temporary solution, its immediate value being that it is clean and cheap. It is likely that a similar pragmatic view will be taken of future requirements, problems being fixed as and when they arise.

III. THE TERMINAL INDUSTRY

The problem of compatibility which has just been described is not a static one. For a full appreciation it is necessary both to review the past and to speculate about the future. The past shows how standards tend to become established and reveals some mistakes, but changes are now beginning to take place which may cause the future pattern of events to be very different.

A A Brief History

The DP terminal industry can be said to have started with the Teletype — AT & T's unbuffered keyboard terminal. It and its imitators still hold about half the market for keyboard terminal devices. A later contribution came from IBM in the shape of the 2740. Most other teleprinter manufacturers duly aligned themselves with one or the other.

The late '60s and early '70s saw rapid growth in the use of screen devices, particularly IBM's 2260/65. VDUs had been available earlier, and impetus was gained principally from the introduction of page mode operation and, later, protected formatting and other technical improvements. IBM now holds over 30% of the market for non-intelligent VDUs.

Since 1974 printing and display devices have become available based on micro-processors. These are often programmable and with local mass storage such as cassette tape or floppy disc. The reduced cost of processors is also encouraging the use of clustered devices such as the ICL 7502 and the IBM 3790, in which the processor operates both as device and communications controller, and includes some applications software.

Apart from special-purpose devices such as financial and point-of-sale terminals, the other major market for terminals is in remote job entry (RJE). Here IBM was again the early market leader with the 2780, but since RJE terminal operation is so closely associated with the mainframe operating system, many manufacturers set their own unique standards. Many of these such as CDC, Univac, ICL and General Electric subsequently became targets for emulation by the independent suppliers, along with the inevitable IBM. Many of the other mainframe manufacturers' RJE terminals also emulate IBM as well as supporting their own protocol.

A further word about the independent suppliers is necessary to complete the picture. Their appeal is not only cost — often 15% cheaper than the standard terminal — but also a willingness to adapt their product to meet special requirements, and to offer innovative features. To take teleprinter terminals as an example, features introduced by independent suppliers include two-colour and reverse printing, switch-selectable communications speeds, mass storage capability, printing speeds up to 120 cps, and operator-changeable character sets.

B Codes And Protocols

The pattern of use of codes and protocols at the moment does of course mainly reflect the historical development of the terminal industry, rather than any recent trends. Nonetheless it is useful to begin with a summary of the current situation before considering what is changing or what is likely to change.

Three protocols are sufficiently widely used to warrant describing in detail:

1. **Teletype (TTY)**
   This protocol is emulated by 60% of teleprinters available from independents and is used by about 50% of all installed teleprinters. It is also used by many scroll-mode VDUs such as those marketed by Hazeltine.

It uses the ASCII 7-bit code, which has control and graphic (or printable) code subsets. ASCII is also used with other protocols and is the dominant code for keyboard terminals.

Transmission is asynchronous normally at 300 bps (30 ch/s), but higher and lower speeds are also used. Character length is 10 or 11 bits – start, 7 code, parity (odd, even or more), 2 stop bits up to 10 ch/s, 1 over 10 ch/s.

It is not supported by IBM, but by most timesharing companies and by all major computer suppliers apart from IBM.

2. **Correspondence/2741**
Correspondence is a 6-bit code with variants for point-to-point (with media control) and computer communications (without). It is used by IBM magnetic card typewriters.

The 2741 protocol uses a 9-bit character – start, 6 code, parity, stop.

Transmission is at 134.5 bps ($\simeq$ 15 ch/s), half duplex.

It is supported by IBM and by some timesharing companies, and is used by about one quarter of all installed teleprinters.

3. **Binary Synchronous Communications (BSC)**
This is the standard protocol for higher-speed IBM terminals and is emulated by most of the other mainframe manufacturers.

It uses an 8-bit code set which can either be EBCDIC or 7-bit ASCII plus parity. Compared with 2741 protocol, it offers improved error handling using longitudinal as well as character parity check. It also has a transparent mode for handling binary data.

BSC is the common framework for a number of device dependent protocols, in which aspects such as buffer length and time-out handling vary. Notable among these are

– 2780 for RJE, using the transparent mode
– 2260 for VDUs with page-mode operation
– 3270 for VDUs, which adds protected format

Other manufacturers offer similar protocols based on ISO and ECMA standards, such as ICL's XBM, Honeywell's VIP/7700, Univac's Uniscope.

As far as character codes alone are concerned, the pattern is clear and well-defined – a large majority of teleprinters and VDUs use ASCII code, while IBM's dominance of the world computer market accounts for a similar dominance of EDCDIC code for RJE applications.

C Industry Trends

With the announcement of Synchronous Data Link Control (SDLC) in 1974 IBM signalled its intention to replace its BSC-based terminals with a new range based on a new full duplex protocol.

SDLC is a bit-oriented protocol with a fixed format frame surrounding a variable length information field, as for HDLC. The information content is device-dependent, so 3270 SDLC will replace 3270 BSC, 3767 SDLC will replace 2741 and so on. Other major manufacturers like Univac, Burroughs and Honeywell have announced similar protocols.

The situation has all the appearances of continuing the haphazard development just described, with IBM's dominance challenged by its major competitors each of which speaks IBM's language but expects its customers to learn its own private dialect.

However, there are differences. Most importantly, the standards bodies like CCITT, ISO, and ANSI have an opportunity to lay down the rules in time to influence events, rather than as a belated attempt to straighten out well-established confusion as has been the case previously. The conflicts between CCITT's X.25, ISO's HDLC and the expected ANSI standard Advanced Data Communications Control Procedure (ADCCP) are minor. SDLC also is very similar – it is a subset of HDLC and, according to the chairman of the ANSI sub-committee, ADCCP terminals will operate in an SDLC environment although not vice-versa. Neither X.25 nor ADCCP are directly comparable with SDLC, however, because they are much wider in scope. The crucial comparison is between X.25 (and, when details are known, ADCCP also) and IBM's Systems Network Architecture (SNA), which incorporates SDLC. SNA is elaborated in the next section of this report.

The opportunity open to the standards bodies, if they can sink their differences, arises from the changed position of IBM as an innovator. Earlier, the market leaders were able to set de facto protocol standards which their competitors were forced by market pressures to emulate. Since then an inertia has built-up, partly because of IBM's very success in establishing standards which are now widely accepted, partly no doubt because users are dubious about the benefits offered by new communications products. There is a clear reluctance on the part of BSC users to move on to SDLC, and SNA has obtained few converts thus far, particularly in Europe. This inertia is not so great for non-IBM users, who may feel that they would lose less by abandoning a minority protocol. Several of IBM's competitors have announced their intention to conform to HDLC/X.25 standards, which may allow these to become established, at least in Europe, before IBM has manoeuvred its reluctant forces into position.

However, it does not do justice to this new generation of pr tocols to portray them merely as another round in the manufacturers' market strategy, or in the standards battle (see Exhibit 3). They are far more versatile than the protocols which they are intended to replace. As has already been indicated, they are bit-oriented and fully independent of the method used to code the information; they are capable of both half and full duplex operation; they can cope with both primary-secondary and primary-primary operation (i.e. one in which either of the communicating parties may initiate and control an exchange of information rather than the master-slave relationship usual now – see Exhibit 4), and error handling facilities are improved.

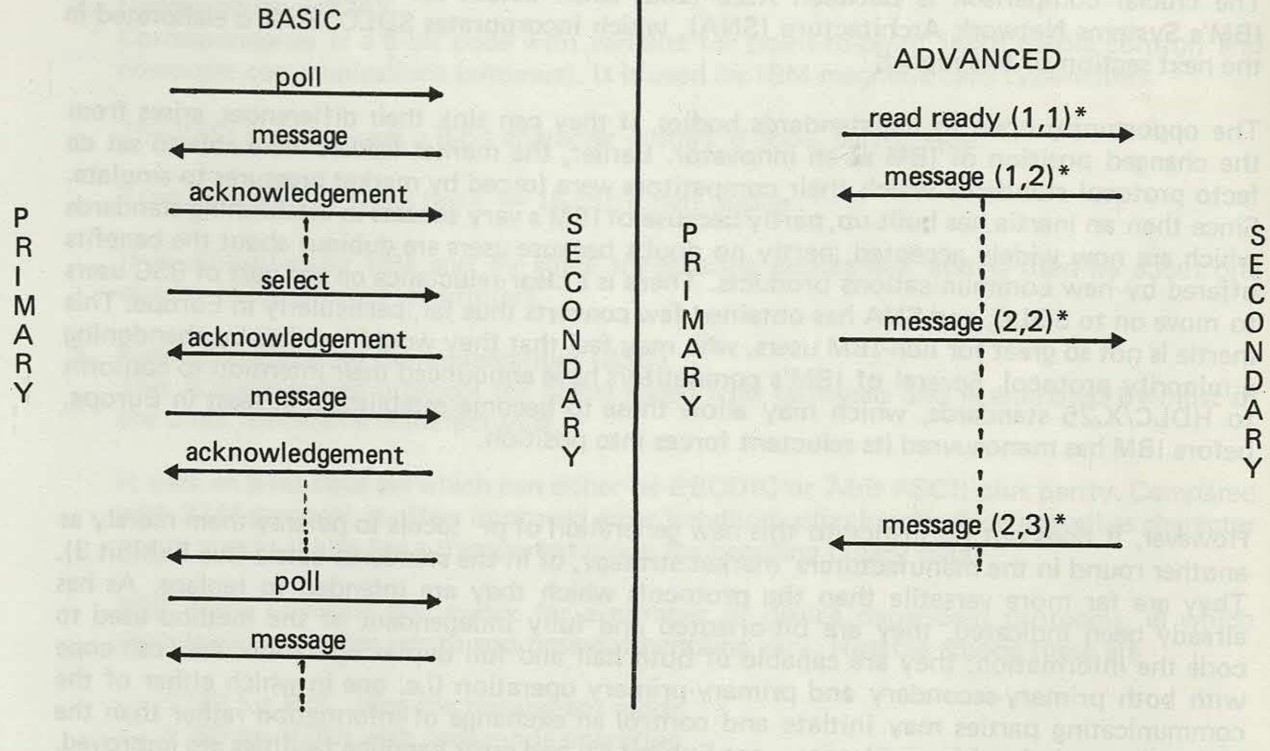
In several respects then they acknowledge the trend towards more intelligent terminals, producing more complex patterns of communication at higher transmission speeds. They also anticipate the introduction of public switched data networks.

The terminals themselves will provide higher printing speeds, increased buffer and mass storage capacity and enhanced programmability. Line speeds also will increase – 50% of terminals will probably be operating at 2400 bps and above by 1980. Well over half of these will operate at 4800 bps and above.

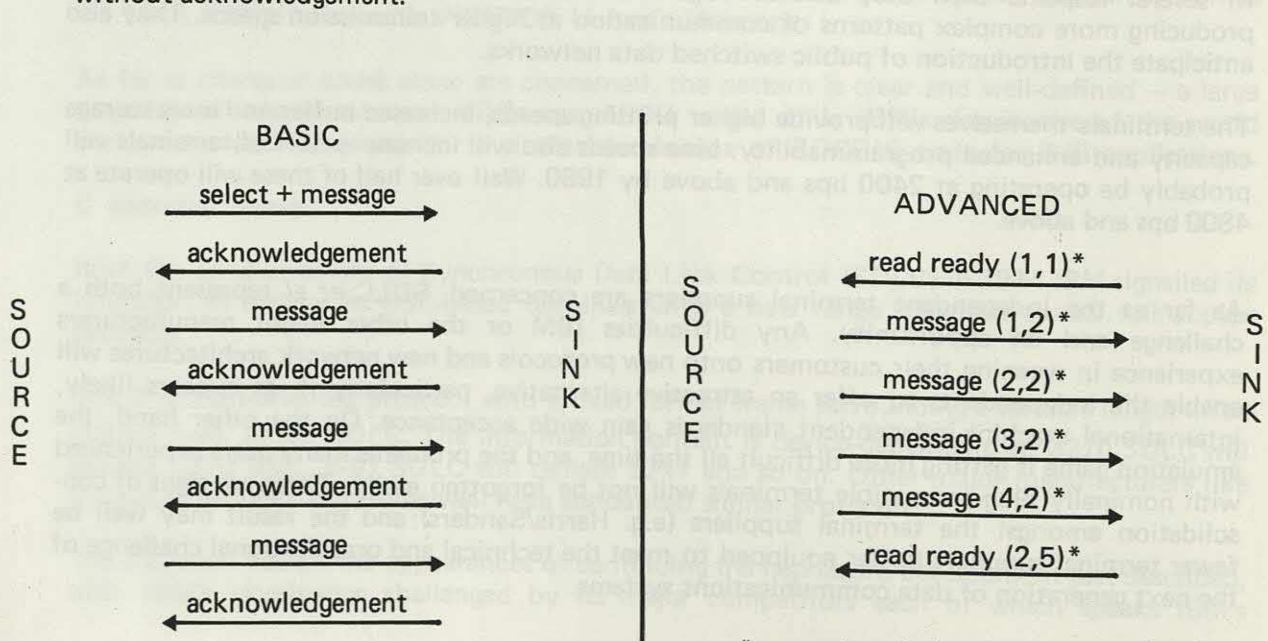
As far as the independent terminal suppliers are concerned, SDLC *et al* represent both a challenge and an opportunity. Any difficulties IBM or the other major manufacturers experience in weaning their customers onto new protocols and new network architectures will enable the independents to offer an attractive alternative, particularly if, as appears likely, international machine-independent standards gain wide acceptance. On the other hand, the emulation game is getting more difficult all the time, and the problems many users experienced with nominally plug-compatible terminals will not be forgotten easily. There are signs of consolidation amongst the terminal suppliers (e.g. Harris/Sanders) and the result may well be fewer terminal suppliers better equipped to meet the technical and organisational challenge of the next generation of data communications systems.

Exhibit 3 Basic mode and 'advanced' control procedures — two key differences

1. *Suitable for high traffic volumes*
   eg acknowledgements are embedded within the information streams.



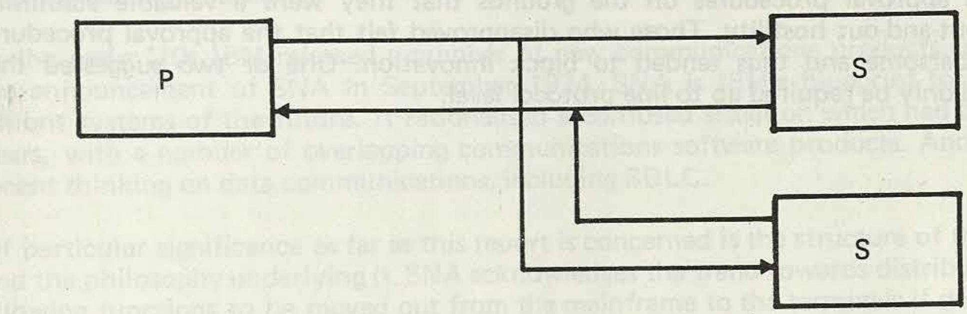
2. *Adaptability to traffic pattern*
   eg many messages may be transmitted without acknowledgement.



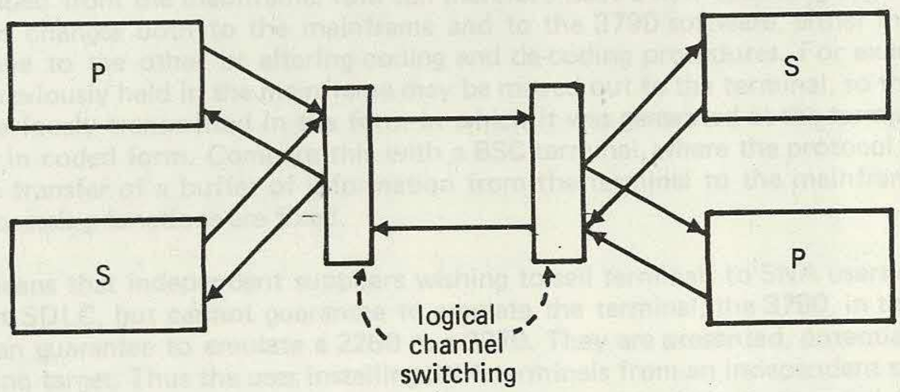
* numbers in brackets are send and receive sequence numbers held in the message header

EXHIBIT 4 Classes of procedure defined for HDLC

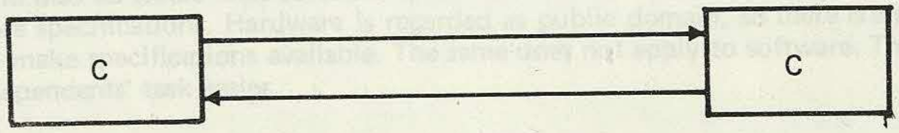
UNBALANCED — one primary controlling one or more secondary stations



SYMMETRICAL — two primary and secondary pairings back to back



COMBINED



Primary stations (P) send commands, receive responses and are responsible for error recovery.

Secondary stations (S) receive commands, send responses and participate in error recovery.

Combined stations (C) send and receive both commands and responses and have equal responsibility for error recovery.

D The Suppliers' Views

We interviewed a number of terminal suppliers in the UK with differing positions in the market, ranging from the established mainframe manufacturers including IBM to the independents and the telecommunications companies. Their views on users' terminal needs varied according to their particular viewpoint, which is only to be expected.

With the exception of IBM all acknowledged the need to be aware of what the other manufacturers – and particularly IBM – were doing. Some acknowledgement was also made to international standards like HDLC, but for most the realities of the market clearly came first. They will conform with international standards if they can, but not if this involves a competitive risk.

Most also commented on the regulating role of the Post Office. Views ranged from acceptance of Post Office approval procedures on the grounds that they were a valuable stabilising influence, to out-and-out hostility. Those who disapproved felt that the approval procedures were too cumbersome and thus tended to block innovation. One or two suggested that approval should only be required up to line protocol level.

IV. INFLUENCES

There are now a number of powerful influences bearing on the terminal compatibility scene. IBM is promoting SNA strongly, and it may help them to protect their terminals from emulation by independent suppliers. Many European PTTs have announced their intention to support X.25, which in some ways conflicts with SNA. At the same time new terminal-based applications are beginning to emerge – for example word processing and teletext terminals.

In this section we examine the nature of these forces and speculate about the extent of their influences.

A The Meaning of SNA

In the early '70s IBM released a number of new communications products culminating with the announcement of SNA in September 1974. SNA is IBM's blueprint for data communications systems of the future. It rationalised a confused situation which had built up over the years, with a number of overlapping communications software products. And it incorporates recent thinking on data communications, including SDLC.

Of particular significance as far as this report is concerned is the structure of the SNA network and the philosophy underlying it. SNA acknowledges the trend towards distributed intelligence, allowing functions to be moved out from the mainframe to the terminals if desired. The effect of this is that the terminal-computer interface has become a software interface and this is more easily changed.

Take as example a network of 3790 systems processing accounting information for a financial institution. The applications software which carries out this processing in the 3790 is down-line loaded from the mainframe. IBM can therefore issue a new operating system release which includes changes both to the mainframe and to the 3790 software, either moving functions from one to the other or altering coding and de-coding procedures. For example, a look-up table previously held in the mainframe may be moved out to the terminal, so that a field which was previously transmitted in the form in which it was generated at the terminal can be transmitted in coded form. Compare this with a BSC terminal, where the protocol merely provides for the transfer of a buffer of information from the terminal to the mainframe or vice versa, and processing functions are fixed.

This means that independent suppliers wishing to sell terminals to SNA users can guarantee to support SDLC, but cannot guarantee to emulate the terminal, the 3790, in the same way that they can guarantee to emulate a 2260 or a 3270. They are presented, potentially at least, with a moving target. Thus the user installing SNA terminals from an independent supplier, or come to that, from one of IBM's mainframe competitors, risks being prevented from taking new operating system releases from IBM, and the supplier is faced with a continuing maintenance problem.

It should also be noted that software specifications cannot be obtained by legal means, as can hardware specifications. Hardware is regarded as public domain, so there is a statutory obligation to make specifications available. The same does not apply to software. This will not make the independents' task easier.

B The Competitive Response

Apart from DEC, whose DECNET has been available for some time, little detail is available about the other computer manufacturers' new data communications products. Several have made announcements of policy and general intent (Univac, Burroughs, Honeywell) but are no doubt waiting for the X.25/HDLC/ADCCP dust to settle before committing themselves too far. There is also a feeling that any elaborate network scheme is in advance of most users' needs at this stage – this is clearly the ICL view.

IBM's major competitors (which must now include the mini-manufacturers) have never been able to adopt such an aggressive stance as IBM in their terminal support policies. Burroughs, for example, while designing applications software specifically for its own terminal devices, claims that its Network Definition Language allows virtually any terminal to be supported. To a greater or lesser extent, this is true of the others also. Not surprisingly, however, plenty of reasons emerge for using the standard equipment when there is a possibility of the terminal contract going elsewhere. It would indeed be surprising if any computer manufacturers missed an opportunity to strengthen their ability to hold onto their customers, with data communications equipment forming an increasingly large part of sales.

That said, it is easy to overplay the change that is taking place as a result of developments in network architecture and the increasing intelligence of terminals. It will always cost a supplier money and effort to introduce arbitrary changes designed to confound its competitors. IBM apart, it is likely that the cost will prove higher than the gains. Increased user sophistication and militance is also likely to be a deterrent. And even IBM will not remain completely immune from the influence of public networks which will impose their own common standards.

C The Impact Of Public Networks

Up to now the PTTs have largely only provided the means for users to transmit bits from point to point. This limited the scope of the standards that were essential to ensure satisfactory use of public facilities. In essence all that is required is an appoved modem and a standard modem interface. The rest is the concern of the users and the equipment supplier.

The planned public data networks do not only carry bits, however. At the least they carry packets (with a datagram protocol), and they may provide for complete call sequences (as in virtual circuit schemes). Thus additional standards are a pre-requisite for satisfactory operation of the network, at message level and, sometimes, at dialogue or network level. Hence the work of CCITT which has produced the X.25 recommendation, and the considerable impact this has had – many of the European PTTs and several of the major suppliers have announced their intention to support it.

But X.25 is not only a machine-independent standard which may be imposed on the computer industry from outside. It also conflicts with the centralised control structure of SNA and networking schemes like it. The SNA network is hierarchical, and nothing moves without the prior consent of the Network Control Program resident in the 3705 front-end, which issues polling messages, establishes logical connections between terminal and application, and so on. On the other hand, traffic flow on a public network is controlled by the network itself, which limits the ability of the host mainframe to direct matters and forces the terminals to operate with a higher degree of autonomy.

It would be rash to predict what IBM's response will be to the challenge of X.25, but there can be little doubt that they will have to come to terms with it somehow. Whatever degree of compatibility is achieved between SNA and X.25 it must not be forgotten that this is only part of the problem. Full compatibility which enables terminals to be replaced at will requires agreement on the information content of messages as well, which X.25 cannot influence.

D New Applications

Terminals are of course used for other than data processing applications. In due course it is conceivable that one and the same terminal may be used both for data processing and, say, electronic mail. It is certain also that data processing terminals will share future public data networks with terminals involved in other tasks. Hence the possible influence of terminal applications emerging now on the way data processing terminals operate. The three areas of terminal use most likely to interact with data processing are facsimile, word processing, electronic mail, and teletext. These are reviewed briefly below.

1. **Facsimile (fax)**
   Standards for facsimile transmission have developed in a similar way to those for data processing — the market leaders have set standards which the smaller manufacturers are beginning to follow. It is from digital facsimile that any possible influence on data processing is likely to come, and it is also in this area that most developments are taking place. These include the introduction of fax transmitters designed to send alphanumeric data entered via data terminals and of switched fax networks such as GRAPHNET, which accepts input from computer terminals or fax machines. Some terminals are now capable of automatic operation and thus can run unattended using off-peak network capacity.

   It is probable that any influence exerted will be by data processing on fax rather than vice versa. For example, it is easy to see how the standard fax call sequence could be accomodated by a virtual circuit packet-switching protocol such as X.25 (see Exhibit 5).

2. **Word Processing/Electronic Mail**
   Word processing requires an extended character code compared with that normally used for data processing — to accommodate lower case characters and various control functions. EBCDIC in transparent mode is suitable, and work has been carried out by ANSI on code extensions for 7-bit ASCII. This may lead to a range of standard 8-bit code sets for different purposes, including word processing. 8-bit ASCII could be used in conjunction with a bit-oriented protocol and would be relatively easy for byte-oriented devices to digest.

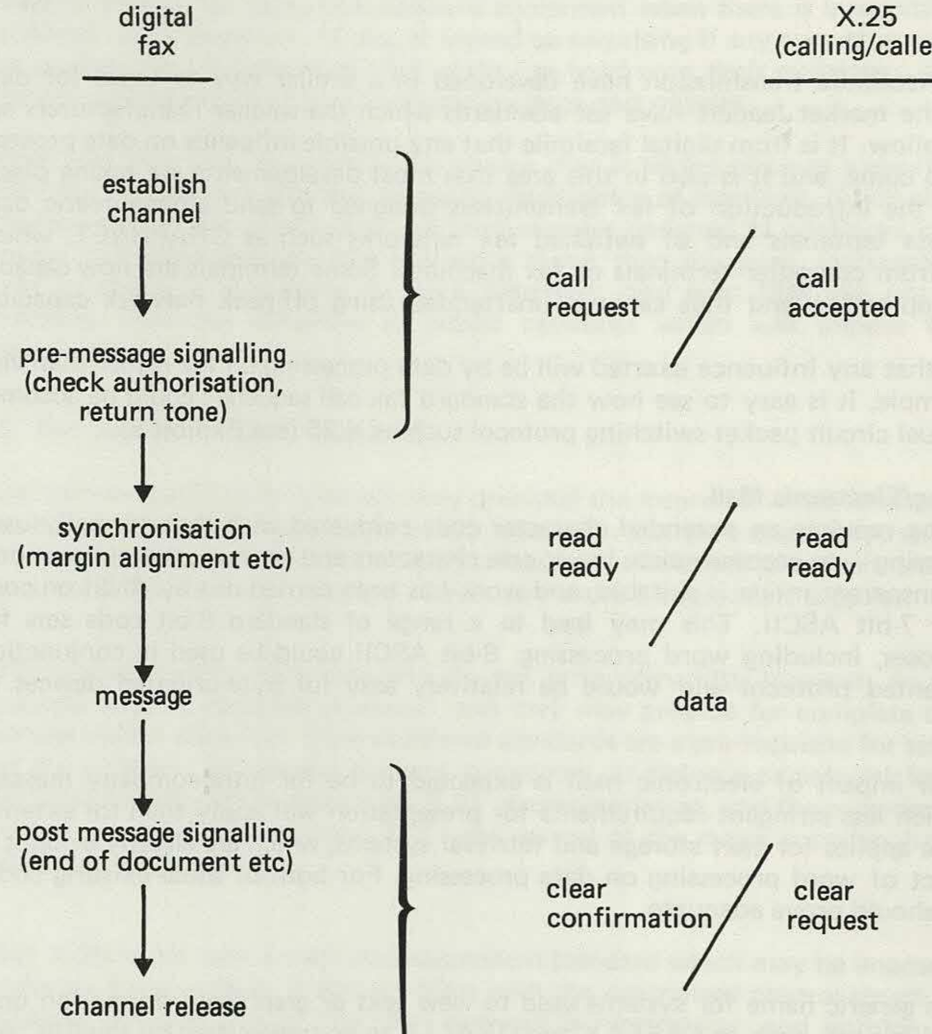
   The first major impact of electronic mail is expected to be for intra-company message traffic, for which less stringent requirements for presentation will apply than for external mail. The same applies for text storage and retrieval systems, which are already evident as the first impact of word processing on data processing. For both of these existing codes and protocols should prove adequate.

3. **Teletext**
   Teletext is the generic name for systems used to view text or graphical information on a domestic television set, such as CEEFAX and ORACLE to be transmitted by the BBC and ITA over the air, and Viewdata by the British Post Office over the voice network. They have a common page format of 24 lines of 40 characters each, and hence can be accommodated within the industry standard VDU screen size of 24 lines by 80 characters. Special Viewdata phones are planned for business use, with two-way message communication via a key-pad or a full keyboard. This might form a possible route to electronic mail.

   The influence of teletext or, more probably, Viewdata on data processing is difficult to evaluate. To a large extent Viewdata networks are likely to operate parallel to and independently of data networks. They may even form the basis for private networks outside the Post Office system altogether.

   On the other hand, Viewdata could easily become such a pervasive communications

EXHIBIT 5 Comparison of typical facsimile and X.25 protocols



medium that its influence might be unavoidable. People accustomed to using Viewdata in their homes might well find it easier if their terminals at work operated in a similar way. Mass-produced Viewdata terminals will also be attractively cheap. Viewdata of course would be a complete protocol, not just a transport medium for any application, and thus is a possible formula for full compatibility between dissimilar devices. The applications for which it is suited will be few but probably common to most organisations.

E The Future Role Of The PTTs

Discussion of Viewdata leads directly to consideration of the future role of the PTTs. Viewdata was originally seen by the Post Office as a good way of generating traffic on the voice network outside peak hours, on the assumption that it would be used mainly in the home. If it were to be successful for business use also, it would draw the British Post Office, reluctantly or otherwise, into the data terminal business. This might give the BPO more impetus in its involvement in the setting of high level standards.

The key question is whether this is a role which the BPO or its counterparts in other countries are equipped to play (assuming that they wish to play it at all). Many users feel that the PTTs should limit themselves to the task which they understand, namely the transport of data. Protocols should be left to those who understand them — the computer and terminal suppliers. So far the result of this division of responsibilities has been a situation which, if not chaotic, is at least confused. If continued, it could lead to fragmentation of the market for future public data transmission services, which would appear to be in the interest neither of the PTTs nor of the users.

V. POSSIBLE TECHNICAL SOLUTIONS

Not all suppliers have a vested interest in incompatibility, and there are also supra-national and research organisations for whom a solution to the problem is of either academic or political interest, or both. Two possible solutions are reviewed in this section. One uses the increased power of micro-processor based terminals to make them more flexible; the other is an attempt to find a solution to the underlying problem by means of the so-called Virtual Terminal.

A Multiple Emulation

Cheaper processors make it possible to incorporate far more logic in terminals at only a marginal increase in cost. One way this additional power is being used to increase flexibility is by programming the terminal to emulate more than one protocol.

Initially the approach of the suppliers was relatively unsophisticated — the change from one protocol to another was made by loading a new program, or by setting a switch. Two examples of devices now available illustrate the current state of the art.

1. **Kongsberg FBT**
   The Norwegian firm Kongsberg have developed what they call a 'flexible user terminal'. It is designed to run against a number of mainframe systems in either remote job entry or interactive mode simultaneously. For example, it can run as an IBM HASP workstation and as a UNIVAC NTR terminal at the same time as it is driving interactive VDUs using IBM 3270 and Uniscope 200 protocols. It has its own command language, so that the user can operate the terminal in broadly the same way whatever mainframe he is dealing with, but there is also a transparent or native mode. This enables the user to adopt the mainframes command set when this is convenient.

   The terminal has been delivered to several large users in Scandinavia.

2. **Megadata 700/UETS**
   Megadata's 700/UETS (universal emulating terminal system) is a customer-specified system which has been delivered to NASA in Houston. A few are also on trial at Renault in France. The system has 20K of memory of which 12K is used for resident software, including emulation routines for up to 5 terminals. The NASA system handles Hazeltine 2000, Uniscope 200, IBM 2265 and IBM 3270 protocols.

   There are also numerous thumbwheel and switch settings for synch/asynch, odd/even parity etc.

   The price to NASA, including interfaces for a floppy disc and Uniscope printer, was $7700.

Both these terminals represent a pragmatic attempt to solve the problem of terminal incompatibility by attacking the symptoms — differences in protocol. They are likely to be effective in a given, stable environment, but will be vulnerable to the mainframe suppliers' policy of planned obsolescence for their own terminal devices, which may casuse the protocols to change. Section IV A explained how changes of this nature are becoming easier to implement with the network structures now being introduced. Also, protocols are becoming more complex, which increases the danger of incomplete or sub-optimum emulation. Perhaps the increased experience of the established terminal manufacturers, gained in coping with the present generation of protocols, will counter-balance the new difficulties.

One feature of Kongsberg's FBT which is of particular interest is the standard command language. Unlike the protocol emulation, this is machine-independent and thus contributes to an enduring solution to the problem, such as the Virtual Terminal is intended to be.

B The Virtual Terminal

For resource-sharing networks such as ARPANET and the planned European Informatics Network (EIN) to be successful, it was necessary to define a standard way of accessing the network services. This had to be satisfactory for the range of dissimilar devices which the network's potential users will possess. The need for a common standard led first of all to standard high level protocols, such as for example ARPANET's File Transfer Protocol, and later to the Virtual Terminal.

In the words of Derek Barber, Director of the Executive Body of EIN, "The Virtual Terminal is a set of commands and responses for a hypothetical terminal". All applications are programmed to "see" this hypothetical terminal, as, where the range of applications demands it, one of a number of hypothetical terminals. Where the actual terminals gain access to the communications system, mapping routines translate between the Virtual Terminal (VT) language and that of the particular terminals in use. These mapping routines may reside in the communications hardware — a cluster controller or a communications computer — or in the terminals themselves. The rapidly increasing intelligence of terminals will soon make the latter an economic arrangement.

To be more than another terminal protocol like 3270 or 2780, the VT protocol must represent the general functions performed by the terminal, rather than the physical characteristics or preferences of a particular device or a particular software package. Only if it is machine-independent can it introduce much needed stability into an environment where technological advance is promoting rapid change in network topology, in terminal design and mode of operation, and in system structure.

The paper from which the quotation given earlier was taken summarises the important issues in the design of a Virtual Terminal. This paper is attached as Appendix 2 to this report. It also lists a number of references for those wishing to investigate VT protocol design further.

One or two additional points are worth emphasising. Firstly, the design of a protocol will have two objectives:

— *To standardise the interface between the applications programs and the VT.*
This should help the applications programmer by minimising variations between similar applications.

— *To enable different terminals to access the same application.*
It will be the responsibility of designers of terminal processors to map between the VT protocol and the characteristics of their real terminals, which will vary markedly from one type of terminal to another. Once a satisfactory VT protocol has been devised, it can be presented to terminal suppliers as the interface to match.

The value of the VT lies not only in the fact that it may resolve protocol differences, but also that it decouples the application and the terminals, allowing each to evolve separately. Assuming that a satisfactory technical solution can be found, it therefore leads both to easier interchange of information and also to increased portability for applications software.

There has been a rapid advance in understanding of the concept and its application in recent years, although there are still differences of opinion on the best form for it to take. As far as the ordinary data processing user is concerned, two questions remain to be resolved.

1. **The cost of implementing a VT in the normal corporate data processing environment.**
   It is relatively easy to graft a VT protocol onto a packet-switching network, since the extra program logic only brings about a marginal increase in cost for the node hardware. The types of terminal for which VT protocols have been devised are also relatively straightforward teleprinter devices. VDUs and multi-unit devices such as RJE terminals will certainly be more difficult to cope with. On the other hand, the examples of multiple emulation terminals quoted earlier show how easy it is becoming to crush processing problems with sheer power.

2. **The impetus for development of practical data processing protocols.**
   It is difficult to visualise a VT protocol broad enough to satisfy the needs of the wide range of present, let alone future, terminal applications. The highest common factor of all applications would no doubt be so little as to be of no practical value. Clearly, there must be enough commonality of interest to provide the basis for a meaningful VT protocol. How then is a sufficient commonality of interest to be recognised? And who will have the incentive and also the skill to turn the commonality of interest into a practical VT protocol?

VI. REVIEW OF USER OPTIONS

In the foregoing sections of this report we have described the nature of the terminal compatibility problem, its historical background, and the factors likely to be influential in the near future.

Most big computer users today are planning extended — and often complex — communications networks of computers and terminals which will be of fundamental importance to their business strategies in future. Against the background of compatibility issues described in this report, what should terminal users be planning to do? This section describes three broad alternative courses of action.

A Supplier Loyalty

The simplest option open to users is to standardise on a single supplier for all data processing equipment. This ensures that system and maintenance responsibility is not shared (further than is inevitable that is, since the PTT will always play a part).

The disadvantage of this approach is limitation of choice. Of the existing computer manufacturers only the five leading mainframe manufacturers offer anything approaching a complete range of equipment. Of those only IBM and possibly Burroughs are likely to remain credible as sole suppliers as the market widens to include computerised office equipment.

Limitation of choice will be experienced either as a cost or a convenience penalty, or both. Terminals from the independents are normally between 5% and 20% cheaper than the standard equipment. For special purpose devices the margin is sometimes much wider. The extra cost of standard devices will sometimes be offset by a lower software cost, however, as the standard software will provide wider support for the standard terminals.

Convenience will be sacrificed because it will not be possible to match applications requirements as closely as with an unrestricted choice. Independent suppliers have been responsible for a large number of technical innovations — although by no means all of them. Possibly more important than this is the willingness of the smaller suppliers to provide a fully bespoke product, rather than requiring a choice from a standard catalogue, however wide.

The changing structure of data communication systems may prove to be a powerful argument in favour of single-source computer systems, if it strengthens the ability of suppliers to hold their customers captive. Only IBM is strong enough to formulate and implement an effective policy of this nature. Opposition will come from the PTTs who will require greater control over data communications protocols to operate public data networks. They may also see centralised, hierarchical networks as a threat to the development of widespread interchange of information using electronic means.

B Ad Hoc Problem Solving

Users not prepared to accept restrictions on their choice of terminals will, unless they are fortunate, be compelled to develop some special expertise. Without it they are vulnerable in two important respects:

— When errors occur which are difficult to isolate, they will have no frame of reference against which to evaluate conflicting diagnoses which the suppliers may provide. Often they will have no diagnostic tools with which to test these out.

- When they wish to introduce new equipment or new software, they may not be able to determine in advance what the effects will be.

As a result they will rely on the ability of their technical staff to invent solutions to technical and design problems as they arise. It may be that an external source of expertise like a terminal supplier or a systems house can be found to take care of the initial design and development work when installing a system based on non-standard terminals. But, as explained earlier, the ability of third parties such as these to find enduring solutions to compatibility problems is threatened by the mainframe suppliers' networking products.

Unless these solutions are set in the context of widely accepted international standards such as X.25, they can easily lead users up a blind alley.

This does not appear such a danger for distributed processing systems. Because these tend to separate terminal handling from complex applications processing, it becomes easier to introduce terminal-dependent changes without a major impact on the applications system as a whole, and vice versa. DEC's networking product DECNET illustrates this. Unlike SNA, DECNET does not provide for communication between terminal and computer, but between computer and computer. Terminal handling remains the responsibility of the local processor to which the terminal is attached and is thus independent of information flow between the processors.

C A Standardisation Policy

Standardisation on a limited range of protocols is a middle way between rigid adherence to a single source and the ad hoc approach just outlined. Such a policy could take one of two distinct forms:

1. Virtual Terminal

The design and development effort involved in devising a satisfactory VT protocol (or range of protocols) is likely to limit its application. Users able to justify the effort might be restricted to large dispersed or multi-national groups wishing to promote standard systems throughout the organisation.

The procedure for introducing a VT protocol might consist first of all in programming applications and existing terminals or terminal processors to meet the VT protocol specification. Subsequently the VT protocol could be presented to potential suppliers of new terminals as the interface to be met. At this stage it would be necessary to muster enough buying power to interest the suppliers concerned in investing the necessary effort.

A group of users in France has pooled its resources for just such a purpose. The INFOREP organisation, led by Louis Pouzin of the state research organisation IRIA, now represents a number of powerful users. INFOREP was set up to attack the problems of multi-supplier systems. They have focussed their attention both on contracts and on design concepts. The VT protocol which they have designed is now being included in members' requests for tender and has also been presented to the national standards body.

While it is possible to solve the problems of interfacing terminals with a VT protocol in this way, the same cannot so easily be achieved with the applications. If the VT protocol conflicts with the methods adopted by the communications routines of the computer supplier, it will be necessary either to amend or to by-pass them. Such an undertaking cannot be taken lightly, and must be weighed against the benefits obtained by de-coupling the applications from the real terminals.

2. Choice Of Existing Protocols

A more realistic approach for many users will be to standardise on protocols which they are now using or planning to adopt. Where mainframes from more than one supplier are in use, this might mean choosing a protocol fully supported by each of these suppliers, and requiring terminal suppliers to emulate them all. Thus for a user with IBM and ICL equipment, HASP and 2903 protocols might be chosen for RJE, and 3270 and 7181 protocols for interactive working. Applications programs on the different systems would continue to be written in their native protocols.

This policy is retrograde in the sense that it embraces existing machine-dependent protocols, rather than attempting to break out of the vicious circle which causes applications to chase terminals and vice versa. It does not guarantee users against the need to modify their applications programs as the computer suppliers' communications software and protocols change. But if the standard protocols and the terminal suppliers are chosen wisely a relatively comfortable passage can be relied on, since the mainframe suppliers are bound to provide reasonable continuity to retain their customers.

D The Need For A Policy

However for many users the first question will not be which option to take but whether a definite policy on terminal compatibility is necessary at all at this stage. Two reasons might be advanced for not taking a definite line:

1. **It is difficult to justify.**
   Many users find it extremely difficult to predict their terminal requirements more than a short time ahead. On top of this they have to justify a restrictive purchasing policy and, for the VT option, a design effort, on the basis of intangible benefits. It can easily look as if management services are trying to make it easy for themselves at the users' expense.

   Failure to establish a definite policy probably means adopting the ad hoc problem solving approach by default. This means that difficulties will have to be dealt with one by one for each terminal-based project, which might well look no better for management services than grasping the nettle at the outset.

2. **The problems may soon go away.**
   There are no clear grounds for optimism that the compatibility problem will soon resolve itself. As Exhibit 6 shows, each positive sign can be balanced neatly with a negative one. What conclusion one reaches is largely a matter of judgment, based on relatively inconclusive evidence so far. The plain fact is that there is no *obvious* source of machine independent standards of the necessary scope:

   - the computer suppliers are restricted by marketing considerations
   - the independents do not have enough influence
   - the PTTs are not fully conversant with the problem
   - others, such as research organisations and standards bodies, tend to be remote from market realities.

   There is room for disagreement about this simplified picture of the main protagonists, but it is difficult to see a clear case for any one of them as saviour in the near term.

The micro-processor has not been included here because it will achieve nothing without human skills to support it. It might become the tool with which compatibility problems eventually are solved, but unless the nature of those problems is understood and unless appropriate solutions are devised which enable devices to communicate without enclosing them in a strait-jacket of rules, the micro-processor will merely add to the confusion.

Exhibit 6 Factors for and against the compatibility problem improving

For	Against
Increased influence of PTTs in establishing machine independent standards	Vested interests of established suppliers
New protocols more versatile	Programmable terminals not so easily emulated
Cheap microprocessors enable terminals to overcome incompatibilities	New systems skills needed to exploit increased intelligence
Protocol requirements are becoming better known	Transmission technology is advancing and may invalidate current techniques
Suppliers are becoming fewer and better able to supply total systems	The scope of communications-based systems is widening
Protocols are stabilising round those used by the major suppliers	New applications are bringing new suppliers into the market.

VII. CONCLUSION

This report has described the root causes of terminal incompatibility, estimated how far they are capable of solution, and examined the practical options which are open to users.

The simplest option open to users is to standardise on a single supplier for all equipment. The disadvantage of this approach is limitation of choice. For users who are considering applications beyond data processing alone, the choice is effectively narrowed to one supplier — IBM.

An alternative option for users not prepared to accept restrictions on their choice of equipment is that of ad hoc problem solving, which entails a high level of technical expertise, and a high degree of risk because of the difficulty in finding enduring solutions.

Standardisation on a limited range of equipment suppliers — and hence protocols — is a practical middle path between these options. It can be achieved in two ways. The first is by making use of the elegant Virtual Terminal approach. However the effort involved in implementing the concept at the application level is likely to be considerable, thus probably restricting it to very large terminal users.

The second way — and probably a more realistic alternative for most users — is to standardise just on the protocols which are currently in use or planned. There is a danger that applications programs will need to be modified as suppliers' communications software and protocols change in the future, but the risk can be minimised by making a wise initial choice.

Users reluctant to establish a clear policy on terminals should be aware that there is no clear solution to the incompatibility problem in sight at this time. Cheap processors are able to relieve the symptoms of incompatibility, but this respite will only be temporary. An enduring solution can only come from measures which attack the problem at its roots. Those who stand to gain from a solution — users and possibly the PTTs — do not appear to have the skills needed; those with the skills do not have the incentive or the influence.

Appendix 1

A REVIEW OF INTERNATIONAL STANDARDS FOR DATA COMMUNICATIONS

Four international organisations have contributed to data communications standards:

1. **International Organisation for Standardisation (ISO).**
   Most national standards bodies are represented in ISO, including the American National Standards Institute (ANSI) and the British Standards Institute (BSI). ISO's Basic Mode Control Procedures have formed a rather loose basis for many of the half duplex protocols in use today. More recent work has centred on High Level Data Link Control (HDLC). HDLC consists of a set of standards for data transmission which define a frame structure for several different classes of procedure. These classes include both the master/slave mode of operation in common use today, and more sophisticated alternatives. A basic repertoire is defined for each class with a number of optional functional extensions. In total, HDLC adds up to a comprehensive framework for many types of data communications. Elements of HDLC were first approved in 1975 and the main line of the standard is now fully defined. Work is continuing on the more advanced aspects of the procedures.

2. **European Computer Manufacturers' Association (ECMA).**
   ECMA is an industry association in which all the major computer manufacturers trading in Europe are represented. It has issued a series of standards for information representation on various devices including those used for data communication. Standards are normally referred to by their number – ECMA 16 is the standard for Basic Mode Control Procedures corresponding to the ISO standard mentioned above. ECMA does not appear to be mounting an independent effort to formulate a counterpart to HDLC.

3. **Consultative Committee for International Telephone and Telegraph (CCITT)**
   CCITT is the international standards committee for telephony. Its V series of standards for data communication, such as V.24 for the modem interface, are effective throughout Western Europe (which does *not* mean that national variations cannot occur). The more recent X series of recommendations are for public data networks. The best known of these is X.25 "Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for terminals operating in the packet mode on public data networks." There is also X.21 which is intended to supplant V.24.

   An indicator of the importance attached to X.25 is the fact that it progressed from draft to acceptance in record time – less than a year.

4. **American National Standards Institute (ANSI)**
   ANSI has been responsible for a number of widely-used computing standards, the most relevant in this context being the ASCII character code. An ANSI subcommittee is now working on Advanced Data Communications Control Procedures (ADCCP), which is in its sixth or seventh draft. Several of the major computer manufacturers have promised to support ADCCP when it finally emerges.

This review would not be complete without a reference to the standards created by IBM by virtue of its market leadership. Binary Synchronous Communications (BSC) is now the most widely used protocol for medium-speed terminals, and Extended Binary Coded Decimal Interchange Code (EBCDIC) is the most widely used transmission code for RJE applications. IBM's more recent protocol, Synchronous Data Link Control (SDLC), has similarities with HDLC but has yet to make a major impact on users.

Appendix 2

THE ROLE AND NATURE OF A VIRTUAL TERMINAL

D L A Barber
Director, Executive Body
European Informatics Network

Recently there has been a growing activity directed towards the definition of standards to facilitate the handling of a variety of terminals in a data network. The Virtual Terminal approach, favoured for some research networks, can seem an over-complex solution unless the proposals are seen in the wider context of a general method for information exchange between computer systems. This note examines the principles underlying the Virtual Terminal and tries to indicate the merit of this approach. The ideas described are not new; a few are so obvious as to be common knowledge, some come from early work on standards internal to particular computer systems, and the remainder appear in current working papers circulating in EIN, Euronet, IFIP/INWG, and in the literature (see list of references). For this reason no attempt is made to give a detailed description of any particular Virtual Terminal specification.

1. Introduction

The advent of data networks makes possible the easy physical interconnection of many of today's existing computer systems. Typically such systems comprise a few groups of terminals, connected with one or more large computers providing them with a wide range of services. The designers of each type of system will already have solved the problems of sharing resources, handling different types of terminal, providing effective command languages and so on. In the nature of things the solutions chosen will be broadly similar, but there will be arbitrary differences in detail that make it difficult, if not impossible, to exploit a network interconnection, so that terminals of one system can access services of another.

There are of course some properties of computer systems that must differ between systems, otherwise one system has no advantages over another for any particular application. However this need not rule out the introduction of some measure of commonality for all systems so that users are aided by standard methods of access and interaction when engaged in the more mundane tasks.

The desire for commonality has led to the idea of standard protocols which attempt to offer a uniform mode of working suitable for use by all of the computer systems connected to a network. Of course each system must support the standard protocols in addition to, or instead of, its own, but this is clearly preferable to having each system support all of the protocols of all other systems.

For clarity of definition and ease of implementation several standard protocols are envisaged arranged in layers, or levels, to form a hierarchy with each protocol performing a particular task. In general each protocol uses the facilities or primitives provided by a lower level and offers a set of primitives for the next level above. For example a communications link inevitably introduces errors, but a line-handling protocol can incorporate an error correction mechanism that allows it to offer an error free connection for a higher level protocol.

The layered protocol approach is a 'bottom up' one which usually begins by building up from the data link level. It has the danger that the final edifice may not match the end users requirements, and so a complementary 'top down' analysis is also needed. Unfortunately, because it is rarely possible to identify the 'top' or, indeed, the end user, such an analysis is seldom made. A useful compromise is to start somewhere in the middle, and with a data network the most obvious point is the users terminal. For this reason the problem of handling terminals in a reasonably coherent manner is assuming vital importance now that public data networks are in prospect.

2. The Terminal Processor

A major obstacle to the development of computer systems has always been the incompatibility of the various types of data terminal. Even terminals of a similar nature differ in detailed design, while those using different basic principles such as printing devices and visual displays seem quite foreign to each other. Yet terminals intended for the presentation of textual information to people certainly have common features, and it has long been the aim of system designers to exploit these features to simplify the tasks of terminal handling.

An additional factor when a data network is interposed between terminals and a computer is the delay that may occur in passing data through the network. This makes it difficult to operate terminals in some of the ways common in established computer services. For example with some systems the depression of a key at a terminal causes the immediate display of a symbol followed by the despatch of the corresponding character code to the computer. With others the key depression despatches a code, and the computer replies with a similar code to initiate the local display of a symbol. The latter method (often called echoing or echoplex) offers a degree of error detection but the introduction of significant delay can be annoying to users.

Fortunately the trend with modern systems has been to dissociate the terminal handling tasks from the main computer and perform them in a separate terminal processor. This makes it easy, when a network is introduced, to site the terminal processor near to the terminals, so that the network delays occur between the main computer and the terminal processor, rather than between the terminals and the associated terminal handling software. Almost invariably in today's private data networks, the interaction between the main computer and the terminal processor is by the exchange of blocks or packets, while the terminal processor often interacts with the terminal by an exchange of characters. This allows, for example, the terminal processor to echo characters on behalf of the main computer in order to offer an echoplex facility.

A further and most important benefit derives from the introduction of the terminal processor, for it can be used to shield the main computer from some of the variations to terminal types. This paves the way for a terminal processor to communicate with several different main computer systems using a uniform method of interaction for a variety of terminals. The agreement of standards for such interactions is a vital step that must be made if the new public data networks are to be exploited to advantage, with most of the attached terminals being able to interact with most computer systems.

A special problem arises with a network such as EIN, where existing computer systems, each supporting its own terminals, are to be joined together. Certainly each system already has its own method of handling a variety of terminals, but these differ so there is a need for a translation mechanism between systems. One possibility is to treat each pair of systems separately by introducing ad hoc solutions, the other is to define a standard method for all systems. The most general approach is to develop a standard process-to-process communication mechanism. This is followed in the design of modern operating systems, and has the advantage that agreement on a network interprocess standard would offer a flexible and powerful method of communication for many purposes. It could then be used to link distributed terminal handling processes which represent each type of terminal. This is, however, a difficult approach and it seems more realistic, at the present time, to make each system appear as a terminal processor, with a standard interface to all other systems.

3. Public Networks

With the new public networks which use the packet switching principle, the terminal processor may readily interact with a remote computer by an exchange of packets, as is usual in the private and experimental networks now in operation. If, however, the terminal processor is to interact with the terminals by exchange of characters, it must perform a packet assembly and disassembly function on their behalf. In doing so, it forms a matching interface between packet handling computer systems and character handling terminals. The nature of this interface has led to much current debate, for it is not clear whether extensive data processing facilities could or should be included as part of the terminal handling facilities in a public network or, indeed, whether a public network should necessarily handle terminals at all.

Using the nomenclature adopted by the CCITT (International Telegraph and Telephone Consultative Committee) the computer system becomes a PDTE (packet mode data terminal equipment); the terminal is a CDTE (character mode data terminal equipment) and the terminal processor is a PAD (packet assembler disassembler). The connections of terminals is therefore as illustrated in Figure 1, which shows a data network with two PDTEs, A and B, and two PADs, X and Y, each handling several CDTEs. There is one interface between the network and the PDTEs; (generally this will be to CCITT recommendation X25) and another between the PAD and the CDTEs, this will vary depending on whether connection is through leased lines, the switched telephone network, or some other medium.

Consideration of Figure 1 raises some interesting questions. For example, the interface between the network and the PAD seems to be identical with that between the network and the PDTE, so presumably the PAD is also a PDTE. If so, is the PAD part of, or external to, the network? This is not easy to answer for there will be advantages for many users if the network can handle, and even supply, their terminals, but there will always be new or special types of terminal that warrant a special PAD – probably owned by the user himself. Either way, assuming that the necessary connections can be established, there are a number of possible types of interconnection and these are shown in Figure 2.

If, as suggested by Figure 2, the PDTE communicates in a uniform way with the PAD, then two PDTEs should be able to communicate with each other in the same way. Also, two PADS should be able to intercommunicate in order to provide facilities for interaction between dissimilar terminals connected to them. If these assumptions were valid, the problems of linking dissimilar computers and terminals would be largely solved. So there is a potentially considerable advantage to be gained by agreeing a standard for the interaction between a PDTE and a PAD. There are two basic approaches to the definition of such a standard:one of these seeks to parameterise the features of different classes of terminals, the other attempts to define a network Virtual Terminal.

4. The Parametric Approach

The parametric approach is most attractive when the prime aim is to handle existing types of terminals. These are classified into categories such as printing units, display devices, graphics plotters etc, possibly with a further division into groups of terminals with sufficiently similar characteristics to allow them to be handled by similar software. An initial interaction is required between terminal and computer to establish the category and group to which a terminal belongs, so the computer must be aware of all the types of terminals it may encounter.

To allow existing private networks to be transferred to a public network without any significant changes, the PAD almost certainly must incorporate a transparent mode of working whereby signals are passed straight through it, allowing a terminal and an associated computer service to interact as if no PAD were present. Built upon the transparent mode, the PAD may incorporate any number of extra features to relieve the associated PDTE of some of the tasks of terminal support. The nature of these features is the subject of much current debate, depending on whether the aim is to minimise the task of handling a few types of simple terminal, mainly for existing systems, or whether a more ambitious attempt is to be made to rationalise the handling of terminals generally.

To be really useful the PAD functions must be standardised so that a PDTE supplying services can rely upon the existence of a common interface with all PADs; however, there is still likely to be a need for considerable involvement of the PDTE in the handling of terminals unless the functions of the PAD are extensive; furthermore, the interactions with terminals are often in the form of an exchange of character codes, and it may be inefficient to pass these through the packet network to the PDTE, which is by definition intended to communicate with packets rather than characters. Above all, because the aim is to simplify the handling of terminals, it is unlikely that the capability for PAD to PAD communication will be provided, and even less likely that the Service Computers (as PDTEs) may usefully employ the standard PAD functions to interact directly together. Some of these disadvantages may be avoided by using the Virtual Terminal approach.

5. The Virtual Terminal Concept

In essence the Virtual Terminal is a set of commands and responses for a hypothetical terminal, defined for use by any PDTE when offering a service. The PAD incorporates software to make it appear like this hypothetical or Virtual Terminal to the PDTEs with which it exchanges packets (rather than characters as in the parametric approach). Potentially, this should permit a more efficient operation of the PDTE, which need be concerned with only the one type of packet terminal. The packets from the PDTE manipulate the Virtual Terminal while the PAD handles the real terminals by exchanging characters with them in order to map commands and data between these real terminals and the standard Virtual Terminal.

The Virtual Terminal concept has an important bearing on the ability of a PDTE to communicate freely through the network, for it need not be aware of which type of real terminal it is dealing with during an interaction. Some hold the view that an even more important advantage is that the Virtual Terminal, in principle, may be used as an interface between two PDTEs, so that they may exchange information in a uniform way; although, as shown later, its lack of symmetry may make this difficult. Some also believe it may serve as the interface between two PADs allowing, again in principle, any real terminal to communicate with any other real terminal regardless of type. For these reasons a considerable effort has been made to define Virtual Terminals for some of the research networks.

Of course, many of the proposals seem over complex for the purpose of handling a few terminals. However the Virtual Terminal can be regarded as a crucial step in the more general development of protocols to improve the compatibility between different computer systems, and an assessment of any particular proposal needs to take into account this wider issue. It must also be remembered that there are very many existing systems which offer the ability to communicate between different types of terminal, so there are no technically difficult problems involved. The big problem is to agree on a standard way to carry out the necessary transformations that all of the subscribers on a network are prepared to adopt.

6. The Shared Data Structure

When people interact they often appear to build a common model which encompasses their area of agreement. Sometimes this is a written document, sometimes a mental image; but the essential feature of an interaction seems to be the joint model with which they agree or disagree, and which continues to evolve until all parties are satisfied.

Intuitively, something similar seems appropriate for an interaction between computers, where a common data structure might serve as a translation medium between dissimilar systems. Put in the context of the Virtual Terminal one might imagine a standard text structure into which messages from one computer are mapped before despatch, and from which they are transformed as necessary on arrival at another computer.

However, the internal data structures may vary markedly between different computer systems and are less well known than are the characteristics of terminals. It is therefore more useful to consider two PADs which use a Virtual Terminal for the mutual interaction necessary to allow their real terminals to communicate together. In this case, a message from one terminal would be transformed by the first PAD into the standard data structure and then transmitted to the second PAD where it would be further transformed to be suitable for another type of terminal. This is illustrated in Figure 3, which shows a teletype communicating with a VDU.

On examination of Figure 3 it is immediately obvious that its implication of a disembodied data structure located at an unspecified point between the two PADs is unrealistic. In fact, either, both, or neither of the PADs may internally allocate a storage area equivalent to the standard data structure in a way that depends on how the interactions with a user at a real terminal are handled.

For example, one system might employ a two dimensional array to represent, say, the screen of a visual display, and to arrange for this array to be updated as a user edits the displayed information by moving a cursor about the screen. The array then forms a kind of 'window' into the file which eventually contains the complete record of the current transactions of the user. Possibly the file might be structured as a sequence of arrays or 'screens'; but this need not be the case, for any convenient mapping between the screen and the user's file might be chosen to suit the system designer's requirements.

In contrast, another system might offer a line based editor which constrained users to indicate the line and character positions where changes were required. In this case a single line buffer could serve as a one dimensional array, so that the 'window' into the user's file would seem to be only a line. Even so, this might not reflect the internal system where a two dimensional array could still be used to ensure rapid handling of a group of adjacent lines, although the user would be unaware of this.

The above brief discussion of how real terminals might be managed indicates the wide variety of ways that data may be structured within a computer system; this underlines the difficulties of mapping data from one system to another, and of selecting a common data structure to suit all systems. Ideally, a criterion for selection is required that is independent of any particular computer system; it must therefore be based on some persuasive arguments about ways that interactions may best be done, and might also attempt to take account of the ways that people commonly communicate with each other, and the kinds of data structure that they find convenient to use.

7. The Virtual Terminal Data Structure

The data structure chosen for a Virtual Terminal must act as a transformation medium or 'window' between the two parties to an interaction. In the general case, each party will have a file representing his own view of the behaviour of the Virtual Terminal, which he is manipulating in an attempt to correlate his file with that of the other party. This is illustrated by Figure 4. The fact that one or both of the files may be stored in a memory or displayed on an output device need not alter the basic principles that are involved.

When a file is mapped via the Virtual Terminal window, a different file is obtained. This suggests that the data structure for the Virtual Terminal really should be closely related to a standard Virtual File protocol. It is not the purpose of this paper to discuss file transfer protocols, but one key problem is recovery after failure to a previous check point. A well structured Virtual Terminal could be helpful, so it seems plausible to use a multi-dimensional array as is shown in Figure 5. By analogy with accepted usage the array comprises a Volume of Pages, each having Lines of Characters. Volumes are held on shelves in Racks; Racks are arranged in Bays, etc. The address of a character is therefore hierarchical in the form of bay, rack, volume, page, line, character, and its value.

	B	R	V	P	L	C	
....	b	r	v	p	l	c	VALUE

This is, of course, capable of indefinite extension by defining Bays to be on Floors, Floors in Offices and so on; preferably by choosing different initial letters for the terms used for the address fields.

The question immediately arises about the size of the address field for each component of the array. The use of eight bit fields would give 255 possible values (assuming zero is generally reserved for control purposes). This allows characters to be arranged in positions 1 to 255 along lines arranged in pages of from 1 to 255 lines. However, real pages with 255 lines of 255 characters are not common, real volumes often have more than 255 pages, and 255 shelves high is difficult to reach, so the analogy with familiar usage needs to be stretched. As the file is a virtual one the higher order field structure is not too significant, because real storage space needs to be involved. The choice is quite arbitrary, but it is useful for everyone to agree so that ambiguity in addressing is avoided. Fortunately the 8 bit by 8 bit page/line array seems now to be well accepted as a basis for the Virtual Terminal.

8. The Relationship with Real Terminals

The relationship between the line and page size of the Virtual Terminal and that of the real terminal is the stage at which the arguments about the Virtual Terminal usually begin. It seems attractive to use an ISO standard paper size, say A4, on which about 64 lines of 64 characters can be printed. Apparently four such physical pages could then be mapped onto a virtual page as four quarter pages. But this is not possible, because in addition to printing characters, there are non-printing control characters which take up no space on the physical page, but make the virtual page bigger than the physical page. For simplicity it is better to map one physical page to one virtual page for this may also allow eventual extension to graphics terminals. Usually there will be two possible modes of mapping to a real terminal; one with control codes suppressed – for a finished copy, and the other with some of them included – to facilitate interactions during editing.

The next question is how to mix printing, non-printing and control characters in the Virtual Terminal array. Because the array has to be transferred sequentially over a serial link through the network and must be reasonably easy to parse, it is probably best to arrange characters in the order they are generated. This means any line may contain both types and usually the Virtual Line will be longer than the printed or displayed one. The problem of how to mix control signals and information cannot, however, be dismissed so readily in general, for if the Virtual Terminal idea is extended to include graphical information, control signals might be better arranged in a special part of the page. The use of the Virtual Terminal for graphics will be considered later.

One final difficulty is the varying physical page size of real terminals. Sometimes this can be overcome in a useful way, for text can be acceptable even when reprinted in a different format. At other times such mapping is a nonsense; for example, a large table of results or diagrams can rarely be mapped on a small sheet of paper, and even real books sometimes have pull-out diagrams. But the advantages of the Virtual Terminal approach are numerous, and the relatively minor cases when it fails are not very valid arguments against its adoption, because some ad-hoc solution can always be found such as printing on two real pages and sticking them together afterwards. These problems are not relevant to the design of the Virtual Terminal but are important in deciding how to interface it to a real terminal.

9. The Virtual Terminal Protocol

To use the Virtual Terminal it is necessary to define ways of manipulating, and interacting with, the data structure so that a given series of commands may be expected to achieve a unique change in its contents. And because the purpose of the data structure is to serve as a medium of exchange between two interacting parties, both must be able to make such changes. This demands a method of interlocking or synchronising their accesses to the data structure in order to prevent clashes. The procedures governing these interactions with the Virtual Terminal are referred to as the Virtual Terminal Protocol.

The interlocking of the two parties is achieved by allowing them alternate access to the data structure; once a party has been given access it retains control until it hands over voluntarily to the other party. An exceptional case arises when an interrupt or request-for-control signal is generated by the passive party to regain access. Who starts first, and what happens following an interrupt, will be determined by a higher level interaction protocol, possibly part of a command or control language. But the basic interrupt mechanism has to be provided in the Virtual Terminal.

Some difficult problems can arise when two parties interact, by exchanging messages which are subject to unknown, variable delays. However, these problems have been treated extensively in the context of generalised process-to-process interaction protocols and will not be discussed further here. In contrast with this more general case of process intercommunication the Virtual Terminal is not symmetrical, for it deals in characters on one side and packets on the other. Therefore the ways of manipulation need not be the same for both of the interacting parties, and this is reflected in the definition of the Virtual Terminal Protocol.

10. Conclusions

This paper has attempted to present the broad principles of terminal handling in networks, and the arguments in favour of the Virtual Terminal approach. To progress further with the discussion it is necessary to become more specific about the components and characteristics of the Virtual Terminal, and for this it is appropriate to consult the current working papers listed in the references. These papers give particular proposals for a Virtual Terminal and a Virtual Terminal protocol. Some of these proposals are based on sound argument. Some are on less firm ground, yet others are quite arbitrary. Clearly there are still considerable areas of disagreement between expert opinion, equally clearly the understanding of the principles involved is advancing rapidly. This is important because the Virtual Terminal approach seems the only way to achieve the state where most of the terminals in a computer network can freely interact with most of the computing systems.

A discussion of the more specific aspects of the Virtual Terminal will be the subject of a later paper, but it is hoped that this paper will have proved useful as an introduction to a topic of growing concern to network operators and users alike.

FIGURE 1 ELEMENTS IN A DATA NETWORK

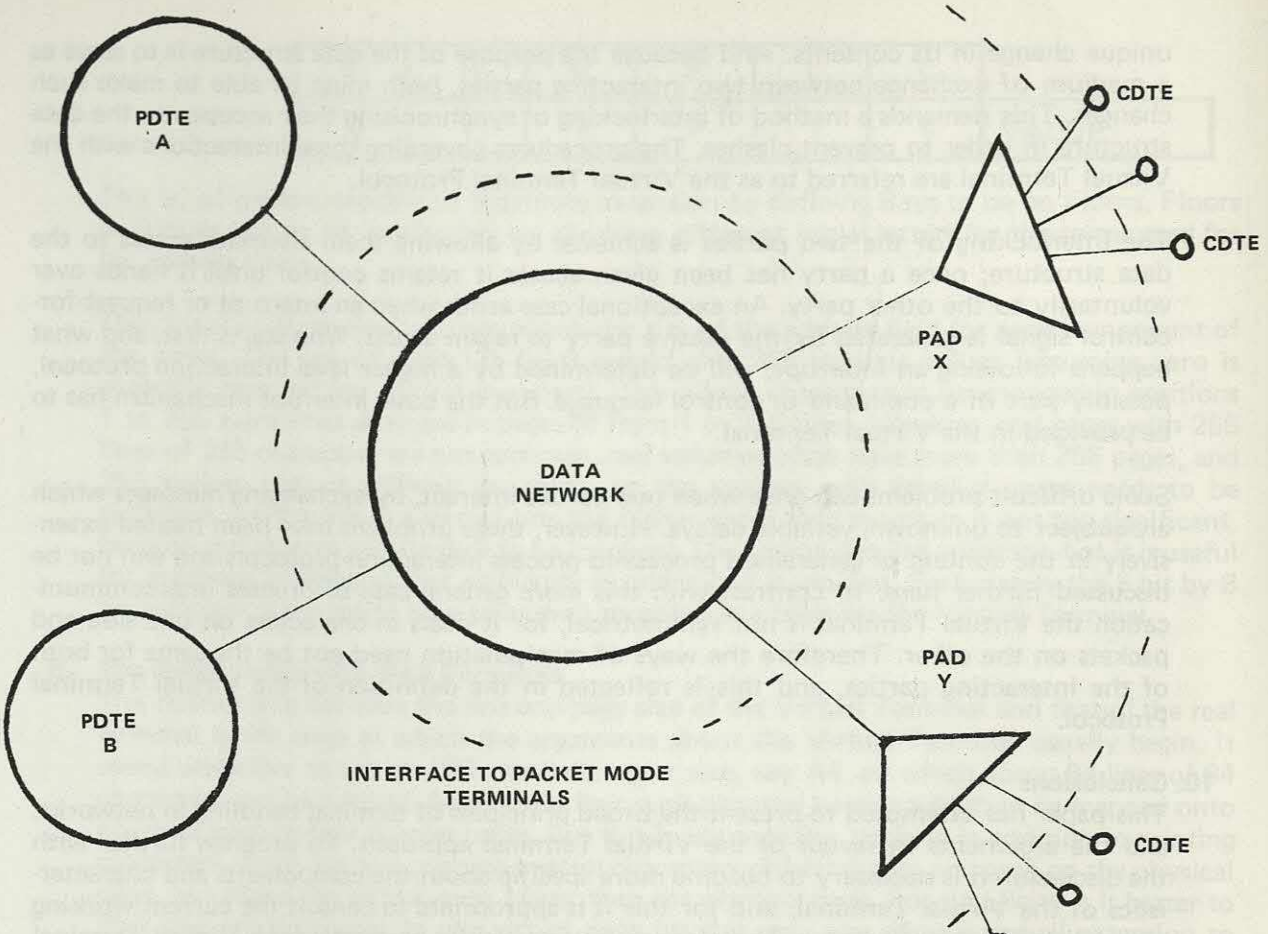


FIGURE 2

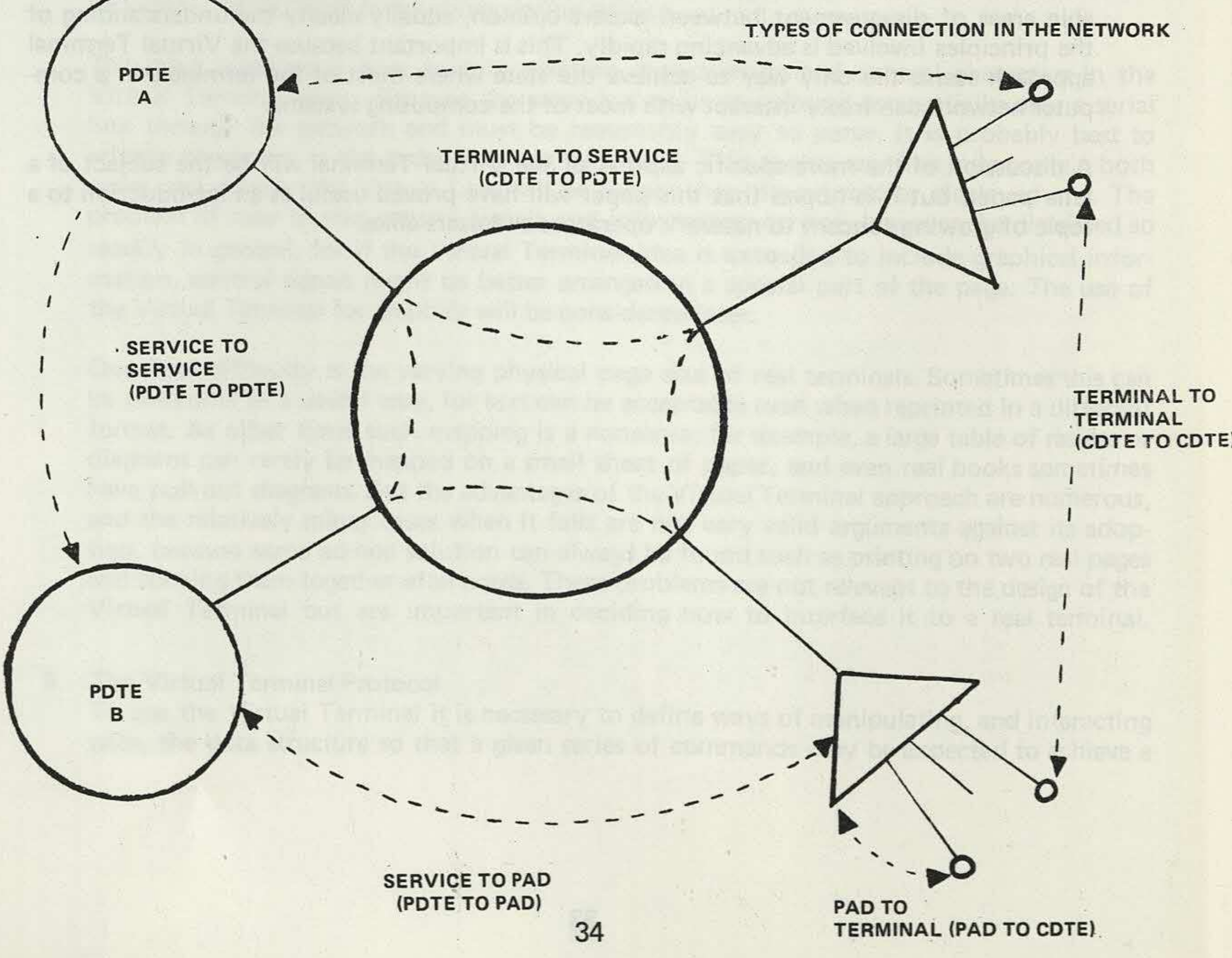


FIGURE 3

THE VIRTUAL TERMINAL BETWEEN TWO PADs

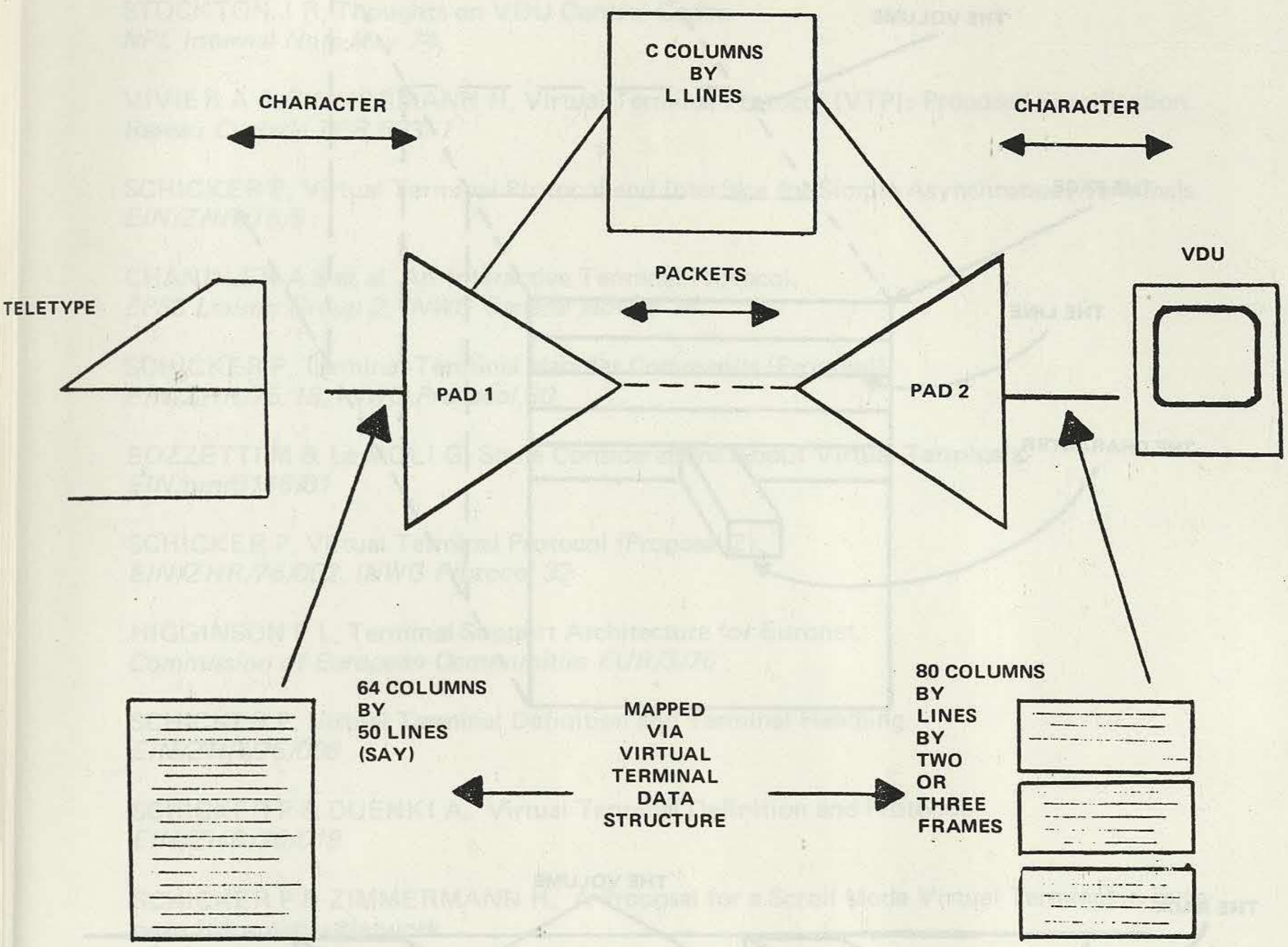


FIGURE 4

FILE MAPPING VIA A VIRTUAL TERMINAL

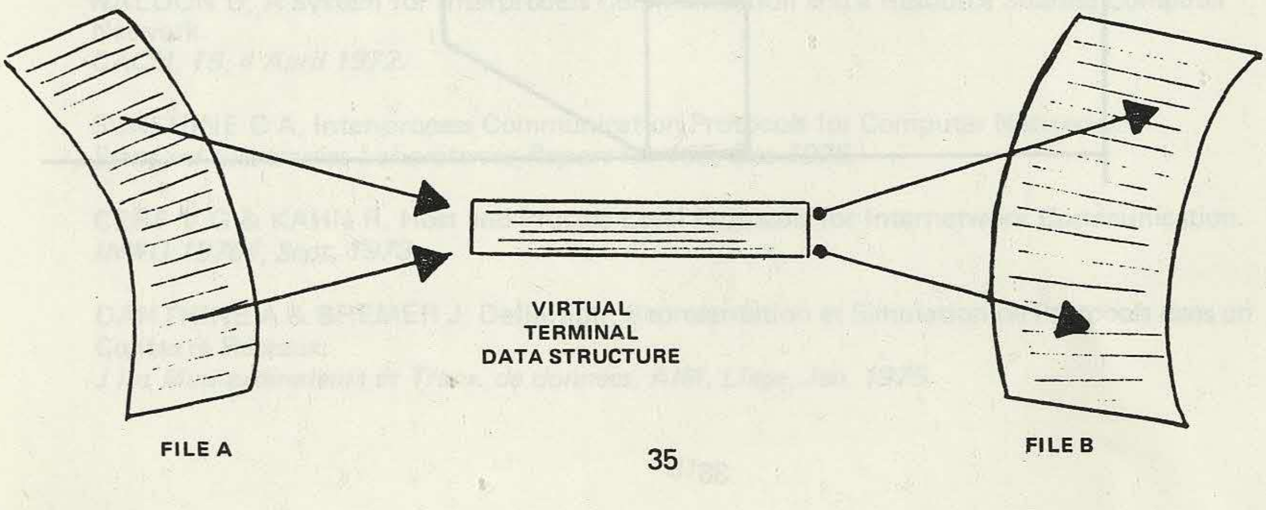
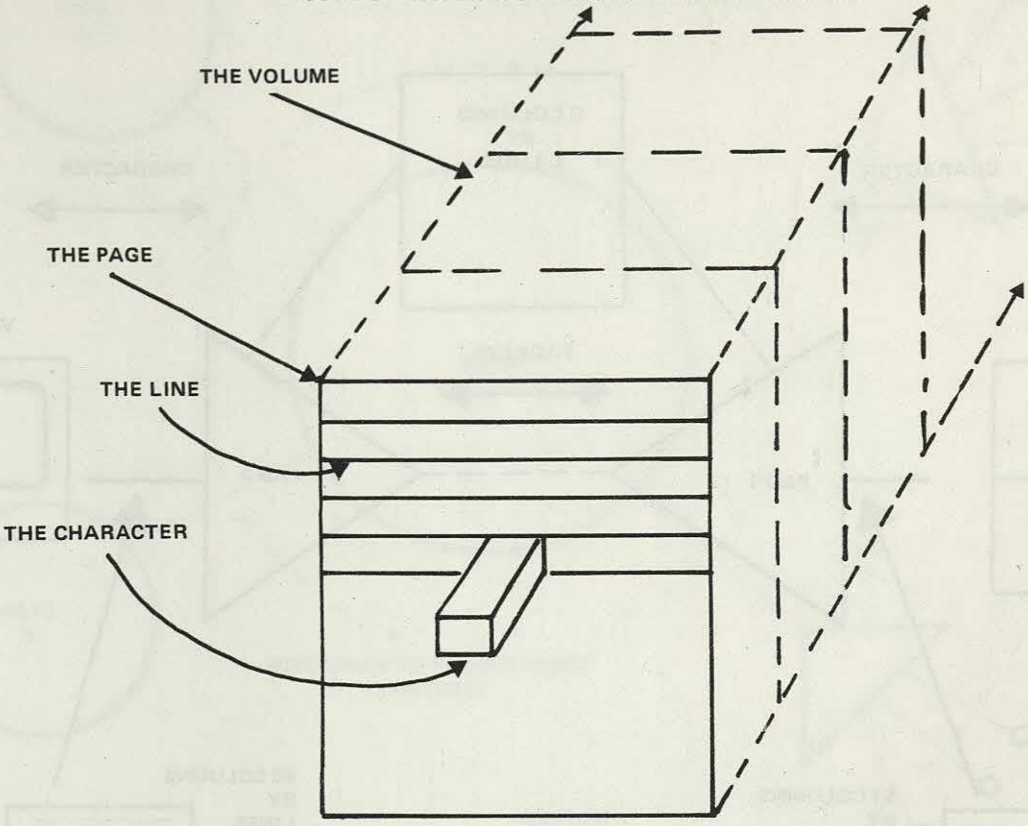
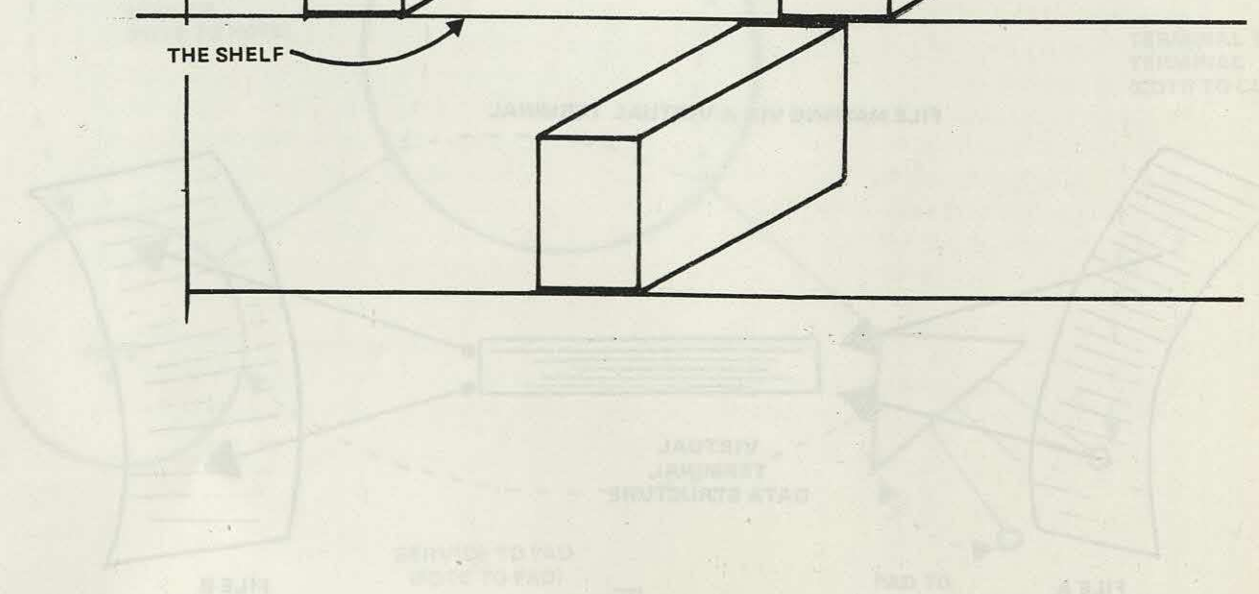


FIGURE 5 PREFERRED DATA STRUCTURE FOR A VTP





REFERENCES

Virtual Terminal Protocol

STOCKTON J R, Thoughts on VDU Control Codes.
NPL Internal Note May 74.

VIVIER A & ZIMMERMANN H, Virtual Terminal Protocol (VTP): Proposed Specification.
Reseau Cyclade TER 503. 1

SCHICKER P, Virtual Terminal Protocol and Interface for Simple Asynchronous Terminals.
EIN/ZHR/75/5

CHANDLER A S et al. An Interactive Terminal Protocol.
EPSS Liaison Group 2, INWG General Note # 74.

SCHICKER P, Terminal-Terminal Handler Commands (Proposal).
EIN/ZHR/75/18, INWG Protocol 30

BOZZETTI M & Le MOLI G, Some Considerations About Virtual Terminals.
EIN/pm/0115/01

SCHICKER P, Virtual Terminal Protocol (Proposal 2).
EIN/ZHR/76/002, INWG Protocol 32

HIGGINSON P L, Terminal Support Architecture for Euronet.
Commission of European Communities EUR/3/76

SCHICKER P, Virtual Terminal Definition and Terminal Handling.
EIN/ZHR/76/008

SCHICKER P & DUENKI A, Virtual Terminal Definition and Protocol.
EIN/ZHR/76/019

SCHICKER P & ZIMMERMANN H, A Proposal for a Scroll Mode Virtual Terminal in European Informatics Network.
EIN/CCG/76/001

SCHICKER P, Terminal-Terminal Handler Commands (Proposal).
EIN/ZHR/75/18

Process to Process Communication

AKKOYUNLU E et al. Inter-process Communication Facilities for Network Operating Systems.
"Computer", June 1974, pp.46-55.

WALDON D, A System for Interprocess Communication and a Resource Sharing Computer Network.
CACM, 15, 4 April 1972.

SUNSHINE C A, Inter-process Communication Protocols for Computer Networks.
Stanford Electronics Laboratories Report No 105, Dec 1975.

CERF V G & KAHN R, Host and Process Level Protocols for Internetwork Communication.
INWG 18764, Sept. 1973.

DANTHINE A & BREMER J, Definition, Representation et Simulation de Protocols dans un Contexte Reseaux.
J Int Mini-ordinateurs et Trans. de donnees, AIM, Liege, Jan. 1975.