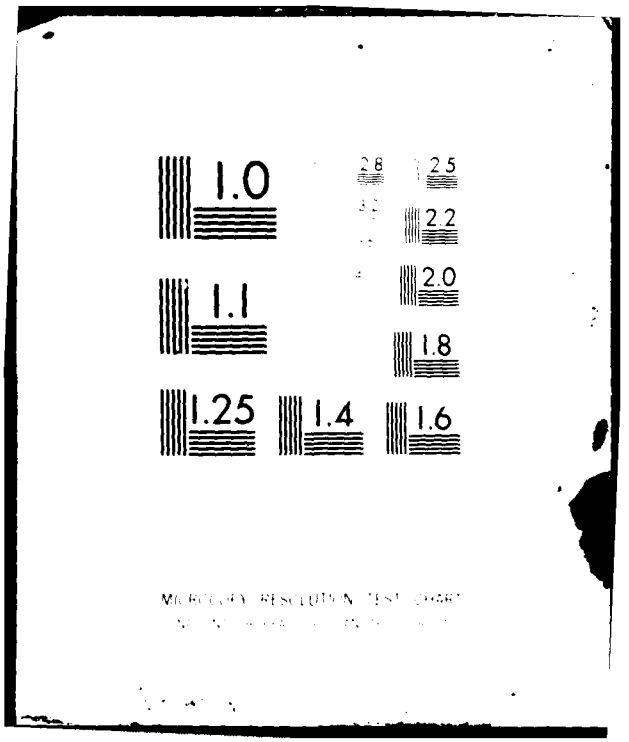
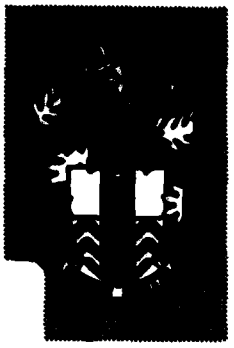
AD-A111 417 ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (ENGLAND) F/G 17/2
PROVIDING AN X25 INTERFACE TO THE RSRE PILOT PACKET SWITCHED NE--ETC(U)
AUG 81 R T EDWARDS
UNCLASSIFIED RSRE-MEMO-3402 DRIC-BR-82084 NL

END
DATE FILMED
03-82
DTIC


1.0
2.8
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6



RSRE

MEMORANDUM No. 3402

ROYAL SIGNALS & RADAR ESTABLISHMENT

PROVIDING AN X25 INTERFACE TO THE RSRE PILOT PACKET SWITCHED NETWORK

Author: R T Edwards

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
RSRE MALVERN,
WORCS.

RSRE MEMORANDUM No. 3402

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 3402

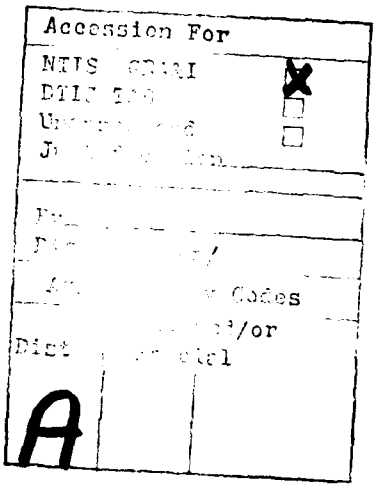
TITLE: PROVIDING AN X25 INTERFACE TO THE RSRE PILOT PACKET SWITCHED NETWORK

AUTHOR: R T Edwards

DATE: August 1981

SUMMARY

This paper describes the successful provision of an X25 interface to the Pilot Packet Switched Network (PPSN) developed at RSRE. The X25 protocol is assessed for its suitability as an interface to the PPSN and as an interface to military networks in general. The technique used to integrate the protocol into the network is described.



CONTENTS

1 INTRODUCTION

2 X25 IN THE MILITARY ENVIRONMENT
- 2.1 Multi-addressing
- 2.2 Precedence
- 2.3 Mobile Hosts
- 2.4 Security

3 INTEGRATING X25 INTO THE PPSN
- 3.1 Which X25 interface ?
- 3.2 Providing End-To-End Services for X25
- 3.3 Coupling the X25 Virtual Call Service to the End-To-End Service
- 3.4 Inter-operability Between X25 Hosts and Block Hosts
- 3.5 Adding an X25 Multi-Address Facility
- 3.6 Testing

4 SOME THOUGHTS ON THE X25 DESIGN

5 CONCLUSIONS AND COMMENTS

6 REFERENCES

DIAGRAMS

Fig. 1 PPSN Architecture
2 PPSN Protocol Structure Prior to X25
3 New PPSN Protocol Structure
4 Communications Packages in the Access Node
5 Multi-address Data Packet For X25 Virtual Calls

1 INTRODUCTION

The Pilot Packet Switched Network (PPSN) was developed to carry out experiments into the military use of packet switched networks. The points of interest include network access protocols, end-to-end protocols, routing, survivability and security (refs 1,2). Figure 1 shows the general network architecture.

[The protocol levels referred to in this paper conform to the definition given in the CCITT X25 recommendation (ref 3). This identification of protocol levels has been formalised by the International Standards Organisation (ref 4) where a physical layer 1, a link layer 2 and a network layer 3 are supplemented by a transport layer 4 and three more layers which deal with the higher aspects of data communications, eg data presentation.]

At the start of the PPSN project (1976) it was decided to design from scratch whatever protocols were thought necessary for a packet network in the military environment. At that time there were no firm standards for packet network access and there were doubts as to whether new standards would suit the military requirement. Protocols were therefore designed to provide link (level 2) and packet (level 3) access to the network (ref 3). The emphasis in the design of these protocols was simplicity. This is considered to be an important factor in producing a highly reliable military network. A simple protocol not only results in reliable software implementations but can also greatly reduce the effort and cost of implementation and testing.

The PPSN provides at level 3 both virtual call and datagram services. Three protocols are separately identified at level 3 for dealing with host connection, virtual call and datagram services. They are collectively referred to as the Block protocol. A host using these protocols is known as a Block host operating over the Block network interface. Certain aspects of the protocol design are unusual. The virtual call and datagram protocols have a multi-addressing capablity. Multi-addressing is frequently required in military data communications and one of the purposes of the PPSN project was to investigate multi-addressing techniques for packet switching.

As time went by it became evident that the X25 protocol, as an international standard could be more important to a military packet switched network than access protocols customised for a military requirement. The reasons for this are easy to understand. Firstly there may be a need for military hosts to operate through public X25 data networks, using them perhaps as fall-back in wartime or simply for economic reasons in peacetime. On cost grounds military computer applications are, where possible, hosted by 'off the shelf' commercial computer hardware and software. In order to gain experience in the use of packet networks the military can minimise their costs by running their applications on host machines in the manufacturers communications environment - which will, most likely, include X25 network access.

It had been intended for some time that a network, based on the PPSN design and software, would be created by the military in order to gain experience in the use of packet networks. Thus in order to satisfy the military requirement for an X25 network it was decided that the PPSN would provide an X25 network interface in addition to the existing network interface. The provision of an X25 interface to the PPSN would also make it possible to investigate and demonstrate how particular military requirements such as multi-addressing could be accomodated within the X25 specification.

As a consequence of providing two different network interfaces it was decided that hosts connected by different access protocols should, if feasible, be able to communicate with each other.

At this time the PPSN already provided X25 level 2 link access to the network, to be used in conjunction with the in-house level 3 protocols. X25 level 2 is in fact provided by autonomous micro processor based hardware called 'line units' which plug into a PPSN node and communicate with the node using a simple data transfer protocol (ref 6).

This paper discusses the problems of providing an X25 level 3 interface to the network and describes their solutions. Knowledge of the X25 protocol is required to fully appreciate the information presented in this paper.

2 X25 IN THE MILITARY ENVIRONMENT

As stated in the introduction there is considerable interest by the military in the CCITT X25 recommendations for use on military networks. Some military networks curently under developement will provide a network interface based on X25.

Although the X25 recommendations allow a great deal of flexibility in the type of service provided there are some facilities commonly required on military networks which are not provided for in the recommendations, notably precedence, multi-addressing, security, connection of mobile hosts and multi-homing of hosts. The current problem is to define standards for the inclusion of these features in a military network environment which operates, as a subset, an X25 service conforming to the recommendations.

In looking to a solution the first point to make is that some of these military facilities may more logically be provided by a higher level protocol, eg multi-homing an application onto one or more networks using a transport (level 4) layer is probably more sensible than multi-homing a host using level 3. It is also worth asking the question as to whether some of these military facilities are really needed in future packet networks. Precedence, for example, is found necessary on congested low capacity message switched networks whose delay characteristics are very much worse than those of modern packet switched networks. Multi-addressing may offer a significant improvement in efficiency on existing message switched networks where bandwidth is at a premium. Perhaps these facilities are not necessary for future military packet networks.

While there is uncertainty over these issues it is worthwhile investigating how X25 can be enhanced to include military facilities while remaining compatible with CCITT recommendations.

2.1 Multi-addressing

In the military environment it is often the case that a message must be sent to more than one geographical location. Sometimes it is necessary to transmit a continuous flow of data to a number of sites which are perhaps replicating some military function, eg radar data to monitoring centres.

Special switching techniques can be used for multi-addressed data in a military network in order to minimise transmission link utilisation. Whether such a scheme is worthwhile will depend upon the amount of multi-addressed traffic in relation to the total traffic load and also the network

connectivity between source and sinks of multi-addressed data. Multi-addressing within the network obviously increases the complexity of the network nodes.

To achieve savings in a packet network a multi-addressing facility must be provided by the level 3 protocols. A higher level host-host transport protocol could be used to distribute copies of data messages but could not achieve any saving in link utilisation.

There is very little choice in how multi-addressing is represented on a datagram network. Each datagram simply carries a header for each destination. The only difficulty is the routing problem of when to send copies of a datagram along different paths.

There are several techniques which can be used to achieve multi-addressing at the network interface. Having tried a number of them on the PPSN it was quickly realised that the simplest was the best.

For the datagram service data is multi-addressed in the same way as within the network, simply by having one header per destination attached to the data field.

Multi-addressing data in the virtual call service is achieved in a similar manner to the datagram service. This technique requires that a call be established between the multi-address source and every destination. The calls can be set up by either end. Singly addressed data packets can be sent on the calls in the normal manner. Whenever data must go to more than one destination a special multi-addressed data packet is sent, containing an indicator of the number of destinations and a virtual call data header for each destination. This technique puts no restriction on communication between the multi-address source and the destinations other than what is normally expected, ie flow control etc.

The most desirable quality of the above technique for use on an X25 interface is the fact that it does not require any alteration to the basic protocol. Section 3.5 describes how this technique, as currently in use on the PPSN, was applied to X25.

2.2 Precedence

Although the PPSN will allow several precedence levels within the network and at the network interface the usefulness of this facility has not been demonstrated. All packets within the network and datagrams at the network interface can carry a precedence level. The difficulty is in making effective use of it on a per packet basis.

The most likely use of precedence on datagrams is as a criterion for discarding traffic in a congested network rather than using precedence under normal operating conditions.

Precedence can be applied more easily at the network interface in virtual call flow control algorithms and in call establishment. Here it is only necessary to specify the precedence of a virtual call. This can easily be done in X25 using a network administration defined optional facility.

2.3 Mobile Hosts

Some military packet networks have the situation of hosts moving around the network. This creates the problem of how the hosts register themselves on the network and how the network maintains a location record for routing purposes.

In the PPSN there is, as part of the Block network interface, a Host Connection Protocol. This protocol allows a host to connect to and disconnect from the network in a controlled manner. At connection time the host may specify its network address and even negotiate certain operating conditions. If the host is allowed to become active on the network it is registered throughout the network using a network internal protocol.

An X25 host does not have the concept of mobility built into its connection procedure. It connects to and disconnects from the network at link level according to the level 2 recommendations. A further mechanism at level 3, the restart procedure, is executed before the host is allowed to operate on the network. There is no way that a host can provide the network with its identifier or do any preliminary negotiation.

It is possible to inform the network of the X25 host's identity by alternative means. The host, on completing the restart procedure, could attempt to establish a call to a special address on the network. The host's identity can then be supplied in the call request and other particulars can be communicated using the call request facilities field or even the data phase of the call. This first 'host connection' call could be handled completely by the access node or might be routed to a network control centre if, for example, authentication is done centrally.

It is thus possible to have mobile X25 hosts using special procedures compatible with the X25 recommendations.

2.4 Security

Security is of special interest in the PPSN project (ref 7). It has had a major influence on the design of the PPSN access node. It has not however encroached on the design of the network interface protocols. There is little evidence to suggest that a military X25 network interface should contain any reference to security, this being dealt with autonomously either outside the network and/or internal to the network.

If user authentication at the network interface is required as may be the case in a mobile host environment a mechanism using passwords might be implemented as part of the host connection procedure. As pointed out in section 2.3 a host connection procedure can be accomodated within the X25 recommendations.

3 INTEGRATING X25 LEVEL 3 INTO THE PPSN

The provision of an X25 interface to the PPSN has taken about six man months to complete although further effort is required to look into some design issues and to carry out rigorous testing and any updates or maintainance that may be required. In this section the basic problems encountered during this exercise are outlined and the design decisions explained.

The software package which provides the X25 DCE function in the PPSN occupies about 13K bytes of memory within a PPSN access node (PDP 11/34). The package is written in MACRO 11 as is all the software written for the PPSN.

3.1 Which X25 Interface ?

The first problem in providing an X25 interface to the network is to decide on the specification. The CCITT X25 recommendation attempts to define a complete end-to-end service as well as a comprehensive network interface, all in the one protocol. It now contains many optional facilities and there are many variations possible in the service provided.

In solving this problem an assumption is made that all commercially available X25 hosts in Britain will operate to the British Post Office's Packet Switching Service (PSS). Therefore it makes sense for the PPSN X25 specification to be compatible with this service.

It is not possible for PPSN to provide an X25 interface completely identical to the PSS but it should be similar enough for any commercially available X25 host to operate over it.

The PSS currently does not provide an X25 datagram service and the PPSN is similarly limited to providing only an X25 virtual call service.

3.2 Providing End-To-End Services For X25

The PPSN access nodes provide an end-to-end virtual call and datagram service across a datagram switching subnetwork. This service is provided by what will be refered to as the trans-network protocol (see figure 2).

Apart from network-user data there is a considerable amount of X25 information which must be conveyed end-to-end, eg call set-up and clear down messages, reset messages and interrupt messages. The existing end-to-end service has call set up and clear down messages but does not cater for such X25 facilities as negotiation at call set up, reset and interrupt. The problem is how to convey this X25 end-to-end information across the network using the trans-network protocol.

There are two solutions to this problem -

a) upgrade the existing trans-network protocol to provide an adequate service for X25.
b) provide a separately defined end-to-end protocol to support the new services. This would operate above the existing trans-network protocol.

The second solution was selected for two reasons. Firstly it has a minimum effect on the existing network implementation. Secondly it suited the security requirement of the network.

Figure 3 shows the new protocol in context with the other network protocols. The new protocol operates a simple virtual call service. It is specifically designed to carry all X25 end-to-end information from the basic protocol such as the reset and interrupt procedure to user facilities such as 'more data' and 'data qualifier' flags. The information is conveyed in messages having a format independent of X25. These messages are carried as data by the existing trans-network protocol. The two end-to-end protocols operate together where they share common protocol functions, eg the new protocol call request message is carried in the data field of the existing trans-network call request packet. This is done not just for efficiency but simplifies the network service overall. The concept of the new level 3 end-to-end protocol is similar

to that of the level 4 'Yellow Book Transport Protocol' - operating as a supplement to the services underneath. For further details of all PPSN internal protocols please refer to reference 8.

3.3 Coupling The X25 Virtual Call Service to the End-To-End Service

The existing trans-network protocol combined with the new end-to-end protocol provide the services necessary for conveying X25 traffic over the network. The job of translating from the X25 access protocol to the network internal protocols is carried out in an access node by a software module known as a Host Support Package. This module is very similar in design to the Block Host Support Package used to provide a Block interface to the network. Figure 4 shows the X25 Host Support Package in context with the other communications packages of the access node. An access node usually provides just one type of host support but it can be built to provide both types of network interface if different types of host are to be connected to it. Further information on the design can be obtained from reference 9.

Because the new end-to-end protocol is tailor made for X25 there are few problems associated with the protocol translation. The only incompatibility of any importance is flow control.

PPSN virtual call flow control operates essentially as a token scheme. The possession of a token at a transmitter represents permission to transmit one packet. Tokens are in fact conveyed across the network by a variable size window scheme in the trans-network protocol. Tokens may be issued in any number at any time during a call and are accumulative.

The ability of the access nodes to call forward data in any quantity cannot be fully realised at the X25 interface because of the fixed window scheme. The X25 level 3 flow control mechanism is a fixed sized window scheme relying on Data, Receiver Ready and Receiver Not Ready packets to convey flow control information. The flow control information can be interpreted as two messages -

1) 'You can send me packets with sequence numbers X to X+W-1'
2) 'Do not send me anything'

X is known as the low window edge. W, the window size, is usually 2 but is negotiable on a per call basis. At call establishment a receiver (DTE or DCE) commits itself to receiving a number of packets equivalent to the agreed receive window size. The commitment can only be reduced from this maximum by either a) using the RNR procedure to effect acknowledgements and halt the flow or b) holding back acknowledgements. Control of flow by either method can be overriden by the reset procedure. A reset is designed to escape from flow control lock ups caused by a fault in the design or implementation of the communicating systems. The receiver's maximum commitment therefore extends to the situations after a reset as well as after call establishment.

Depending on the use or misuse of the reset procedure a receiver may find its control on flow after call establishment somewhat inadequate. This is especially true in the situation where a receiver has a very erratic incoming traffic flow. Because the receiver cannot predict the usage of established calls it may allow a large number of calls with large windows in order to make full use of the available link capacity. It is however open to congestion because of its overcommitment and may find it necessary to rely on flow control at link level. An example is where a receiver supports a number of permanent virtual circuits, most of which are dormant at any one time but all

require high throughput when in use and thus require large windows. The receiver has no effective mechanism at level 3 to avoid congestion if all circuits suddenly attempt to deliver a windows worth of packets.

A more useful flow control scheme in this situation is one using a variable window size on a call. This requires the window size as well as the low window edge to be sent on the call. With this mechanism a receiver can allow large windows when safe to do so and close down the window during a call without the need to withhold acknowledgements or stop the flow altogether. In practice the window size could be conveyed by RR packets to avoid extra header in data packets.

After call establishment and after the reset procedure the PPSN is not necessarily prepared to accept a full windows worth of data packets. Issuing RNR packets will not necessarily prevent the transmission of data and the subsequent flow control violation.

There are two ways around this problem. One way is to ensure that the PPSN token flow control mechanism caters for X25 by accomodating the full X25 window at call establishment and after a reset. The second way is to isolate the X25 flow control from network flow control by buffering data at the point where protocol translation is carried out on the virtual call (in the Host Support Package).

The first method is possible when each end of the call is an X25 host but cannot be done when the hosts participating in the call are using different access protocols (see section 3.4). The second method is generally a good solution when dealing with protocol translation but can cause lock up problems caused by a shortage of buffering. A compromise on the first method is to accomodate an X25 window at the start of a call and if a flow control violation occurs following a reset, the call could be reset or even cleared.

The first method is currently in use but it is to be replaced by the method of queuing at the point of protocol translation.

3.4 Inter-operability Between X25 Hosts and Block Protocol Hosts

The PPSN now provides two network interfaces, the new X25 interface and the old but still useful Block interface. It is desirable that a host operating Block protocol should be able to communicate with a host operating X25 protocol. This has been achieved with a fair degree of success.

The translation of both access protocols into the standard end-to-end protocols provides the basis for communication. As in any protocol translation, where a particular service does not continue through the protocols or is too incompatible, a failure in translation will occur and the best recovery mechanism must be employed. This occurs in the PPSN where X25 resets and interrupts have no counterpart in Block protocol. The recovery policy can either be failsafe, ie register an error and terminate the communication, or an attempt can sometimes be made to maintain communication by falsly emulating the service.

An example of false emulation occurs with interrupts and resets. A Block protocol Host Support Package does not deliver interrupts but does return an interrupt confirmation. It is possible to deliver interrupts to a Block host as data but this will not be enacted until the use of interrupts by users indicates that this policy is worthwhile. A Block protocol Host Support

Package responds to a reset by returning a reset confirmation, no other action being possible or necessary.

A few other incompatibilities exist in the end-to-end service provided by the two access protocols. For example Block protocol defines an 8 bit binary host process identifier (0-255) whereas X25 only allows a 2 digit number (0-99). Where translation fails with process identifiers communication is prevented.

Another problem arises because Block hosts do not recognise any form of maximum packet size negotiation and operate to the PPSN standard of 255 data bytes. The network, as a policy, does not carry out packet fragmentation and prevents communication between an X25 and Block host if the X25 host insists on a maximum packet size less than 256 data bytes. It is still undecided as to whether to introduce some indication of maximum allowed packet size into Block protocol. An alternative solution is to reduce the PPSN maximum packet size to 128 data bytes. This has been considered for other reasons. The network can always insist that an X25 host use this maximum packet size and communicaion could then be guaranteed.

One of the more annoying problems is caused by the X25 fast select facility. Originally the PPSN Block protocol allowed data to be carried in call request and clear request messages. This was later changed to be compatible with the emerging X25 standard which did not allow data in these messages. Now X25 call request, call connect and clear request packets can carry a limited amount of data in the fast select mode. Three solutions to this incompatibility are possible; a) deliver the set-up and clear-down data in PPSN data blocks, b) reject fast select X25 calls to Block hosts or c) change the Block protocol back to what it was. Although the third solution is not usually available in translating between protocols it is being considered as a possible final choice. Currently the problem is dealt with by rejecting fast select calls to Block hosts.

3.5 Adding An X25 Multi-Address Facility

As described in section 2.1 multi-addressing is achieved on virtual calls by creating a multi-address data packet. Only hosts wishing to transmit multi-address data need be aware of this packet. The network always delivers standard data packets so that a host without a multi-addressing capability can participate as a receiver of multi-addressed data even when more than one of the destinations is on the host.

Figure 5 shows the multi-address version of the standard X25 data packet. The first byte of the packet is coded 255 decimal to uniquely identify it from other packet types. This code should remain unique as it would normally indicate an illegal channel group number in other packet types.

The mechanism of handling a multi-addressed data packet is fairly straightforward. The packet is processed for each call specified in the header in much the same way as with a normal data packet. If a violation occurs on one of the calls it is generally desirable that the violation only affects that call. This involves responding to the violation on that call and removing the call header before continuing to process the packet. Certain violations, eg packet too big might affect all calls.

The X25 host generates a multi-addressed data packet quite easily and maintains the virtual calls in the usual manner. Its policing problems and exception handling will occur at the interface to level 4 which is also

required to support multi-addressing in some way - preferably the same way, ie with multi-addressed data messages.

3.6 Testing

The overall task of testing the X25 PPSN can be subdivided into the following -

1. testing the correct function of the X25 services and facilities
2. testing the correct handling of exception conditions
3. testing performance
4. testing interoperability with Block hosts

It is true to say that a great deal of effort is required to carry out the above tests thoroughly. Automating the testing process is essential in the long term for rechecking modified software. This in itself requires more programming effort than the X25 DCE implementation. Experience has shown that even supposedly thorough testing by one group of people does not show up faults discovered by another group. Independent testing by people not concerned with the system implementation is certainly very useful. The experiences of Horton and Thomas (ref 10) may be found interesting. Testing, as listed above, has been carried out to a reasonable level of confidence. Due to the experimental nature of the project and the fact that there is no prospect of the PPSN providing a service in the military environment it was not considered sensible to invest a great deal of effort on testing.

Two different X25 DTE implementations have exercised the network to date and it is hoped to attach others in the near future.

One of the X25 DTE implementations used to exercise the PPSN was derived from the X25 DCE implementation used in the PPSN. This DTE software package, known as the X25 Host Communications Package, was incorporated into the existing suite of test host software to provide X25 network access for test applications previously used to exercise the Block interface. Much of the testing has therefore been accomplished using the existing PPSN test host software. In this way multi-addressing has been demonstrated between X25 hosts and between a mixture of X25 and Block hosts.

Rigorous testing of the X25 facilities and exception handling has yet to be done. A more cost effective approach to rigorous testing of the PPSN is to obtain or borrow a testing facility from another organisation, perhaps already contracted to deal with the business of testing military networks.

4 SOME THOUGHTS ON THE X25 PROTOCOL

In carrying out the work of providing an X25 interface to the PPSN, a thorough understanding of the protocol is necessarily aquired. It is thought useful to include in this paper some general impressions of the X25 protocol gained during the implementation and formed in the light of experience accumulated during the lifetime of the project.

The X25 protocol is based on well established techniques and, in its basic form, is straightforward. However, as a result of requests from many networking communities, the protocol has grown and is still growing in complexity. The evolution of the protocol is achieved by the use of protocol options. This is a convenient way of enhancing the protocol without having to declare existing implementations non standard. If taken too far there is a

danger that many X25 DTE and DCE implementations may become incompatible. Too many options on network interface facilities may prevent a DTE, designed for one network, operating on another. Too many options on network end-to-end facilities may prevent communication between DTEs on the same network or on different networks which are interconnected.

A protocol which is evolving to include new features but trying to remain compatible with previous specifications can become messy, especially with respect to message format. X25 is already showing signs of this.

One example of what might be thought unnecessarily complicated is the data packet and the facility known as packet sequence numbering. X25 offers two data packet formats. The standard is a super compact three byte header which allows modulo 8 sequencing. There is an option of having a four byte header which allows modulo 128 sequencing. Perhaps it would have been preferable to standardise on modulo 128 sequencing rather than offer the choice of modulo 8 or 128. Surely no-one would begrudge the extra header byte in the data and flow control packets, especially if it means a greater degree of standardisation. If a six byte data packet header could have been agreed as not extravagant, there may have been advantages in standardising on modulo 256 sequencing. The control flags (M,Q,D) which are stuck in strange places could have been collected together into a control byte leaving room for more flags as the need arose. The packet descriptor byte would then be dedicated to containing only the packet descriptor.

The criticism of format may be considered petty but it is believed that simplicity and clarity of packet formats are important for the protocol to evolve sucessfully and be easily implemented and maintained.

One optional facility which complicates the protocol specification more than any other is the fast select facility. The fast select option solves the common problem of whether to allow a data field in certain control packets. Some say no on principle, others say yes on grounds of efficiency. X25, in allowing both methods, satisfies everybody while adding significantly to the complexity of the X25 specification. It would be so much simpler to accept one mode of operation. It should not cause anybody much of a problem whichever method was specified; the problems are caused by having a choice.

An example of what was thought unnecessary is the virtual call reset procedure. It is an added complication that will not prove to be justified. The PPSN Block protocol does not include a reset, the policy being that if a flow control lockup occurs the call is cleared. A lockup will only occur if there is a fault in the protocol or in the machine implementations and this is more likely to be rectified if calls are being cleared down.

A similar opinion is held for the interrupt procedure. This procedure is designed to communicate a small amount of data (1 byte) to a participant in a call outside the normal data flow control. It is a mechanism devised before the use and characteristics of such a procedure could be properly determined and again does not seem to justify the extra complication in the protocol and the implementations.

The retransmission facility, using the reject mechanism, seems to be of little use on an X25 network interface. The design is unsymetrical, retransmission taking place only in the direction DCE to DTE. This feature puts quite a burden on networks because packets must be stored awaiting acknowledgement. Luckily it is a network option. Unluckily it is standard on PSS and must

likewise be on PPSN. The retransmission facility might be useful where the level 2 services cannot be trusted but then you would need a symetrical service. Possibly this facility is there to cater for hosts whose flow control and buffering capability is inadequate.

Since the X25 protocol appears to cater for various public and private network environments, looked at in this light, it doesn't really go far enough. There would have been many more applications for X25 if it had been designed as a general communications protocol without the concept of DCE and DTE. This could have been achieved quite easily with just a basic change to the channel multiplexing scheme and a few other minor modifications.

Many criticisms of the X25 recommendations have been expressed wordwide. Some have been satisfied by further enhancing the protocol. Only a few personal criticisms have been mentioned above. All relate to complexity, brought about by trying to create a protocol which means all things to all people.

To be fair the X25 recommendations have only suffered the usual problem of international standards - too many points of view. Many people must consider the various features of X25 to be perfectly reasonable and even useful! At least the standard is being adopted to a degree where further criticism no longer serves a purpose. The X25 recommendations, as a standard, is infinitely better than none at all.

5 CONCLUSIONS AND COMMENTS

In studying and implementing the CCITT X25 recommendations one cannot help but come to the conclusion that the X25 procedures and packet formats are unnecessarily complicated for its purpose as a standard packet network interface. Too many compromises have been made in order to have the recommendation unanimously approved.

There is no doubt that the users of packet networks will benefit greatly by the existance of the X25 standard. There is no reason why the military environment should be an exception. It is certainly possible to make use of X25 in most military environments and add military features without conflicting with the basic protocol.

The implementors of packet networks and hosts may see X25 as a solution to all their problems or may look on it as just another communications obstacle to overcome. Certainly the creation of a reliable X25 device (DTE or DCE) is no mean feat, particularly in the military and certain commercial environments where extreme reliability must be demonstrated by very thorough testing.

The provision of an X25 interface to the PPSN has proved very worthwhile both in terms of experience gained and future usefulness.

The implementation of an X25 DCE has been a valuable exercise in gaining a thorough appreciation of the protocol, highlighting particular communication problems and testing solutions to problems. It has been demonstrated that the technique used for multi-addressed traffic on the PPSN can equally be applied to the X25 protocol. The multi-address service is a very small supplement to the protocol and does not require the normal X25 service to be altered in any way.

The decision to attempt to provide communications between X25 hosts and Block hosts turned out to be most beneficial. This requirement was a major influence

on the overall redesign of the network and resulted in a more modular and generalised approach to protocol design and implementation than would otherwise have been attained.

The PPSN is now a much more useful tool for protocol testing and demonstration purposes. One of its main applications, an operational role in a military copy of the PPSN, has unfortunately been cancelled due to economic restraint. However, in its role as a packet network test bed, PPSN will be used in support of a number of military network projects. There is also a service role for the network in support of work being done in house on internetworking.

6 REFERENCES

1 P H Masterman, "The RSRE Pilot Packet Switched Network", Data Networks-Developements and Uses, Online Ltd.

2 "PPSN Overview" available from T4 RSRE.

3 "CCITT Recommendation X25", CCITT Geneva.

4 "Reference Model of Open Systems Interconnection", ISO/TC97/SC 16 N 227.

5 "PPSN host to Network Communication Interface" available from T4 RSRE.

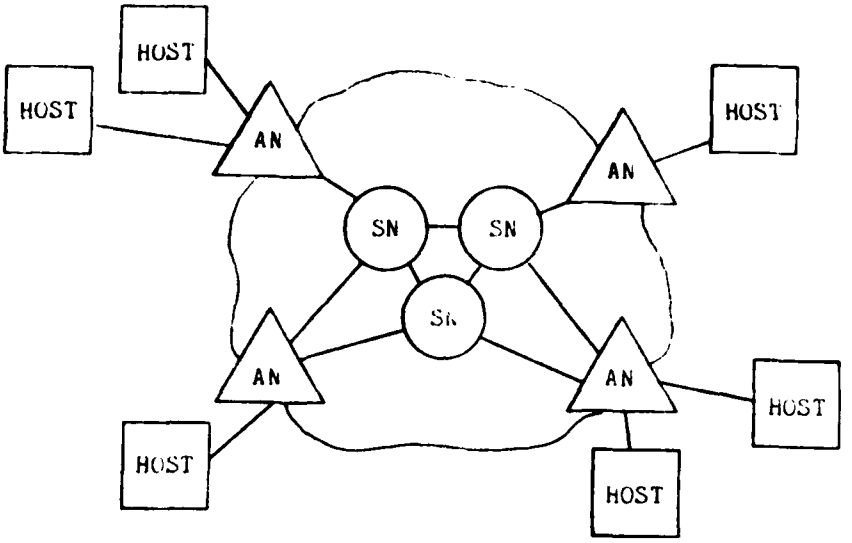
6 A F Martin and J K Parks, "Intelligent X25 level 2 line units for packet switching", Data Networks-Developements and Uses, Online Ltd.

7 D Barnes, "Computer Security in the RSRE PPSN", Data Networks-Developements and Uses, Online Ltd.

8 "PPSN Network Protocols and Packet Formats" available from T4 RSRE.

9 "PFC Design" available from T4 RSRE.

10 J R Horton and J S Thomas, "Developing and testing an X25 interface", Data networks-Developements and Uses, Online Ltd.



AN = Access Node
SN = Switching Node

Figure 1 - PPSN Architecture

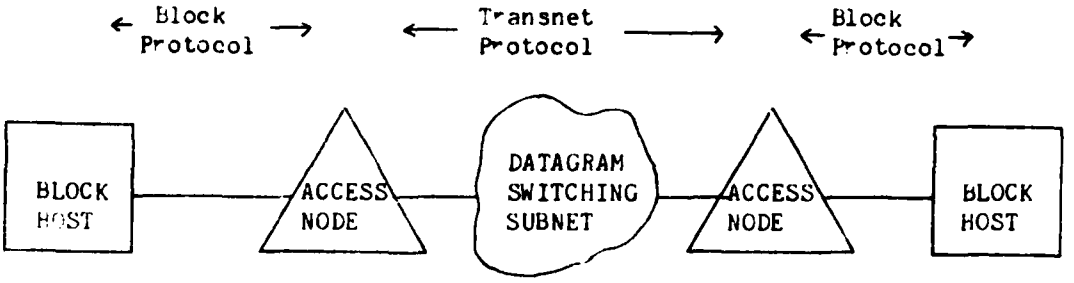


Figure 2 - PPSN Protocol Structure Prior to X25

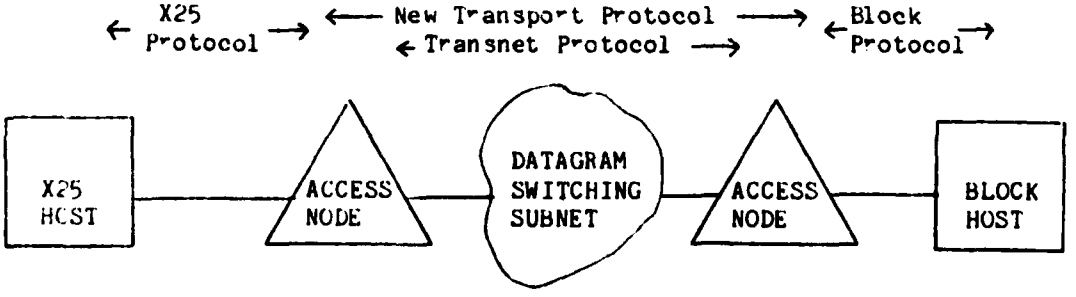
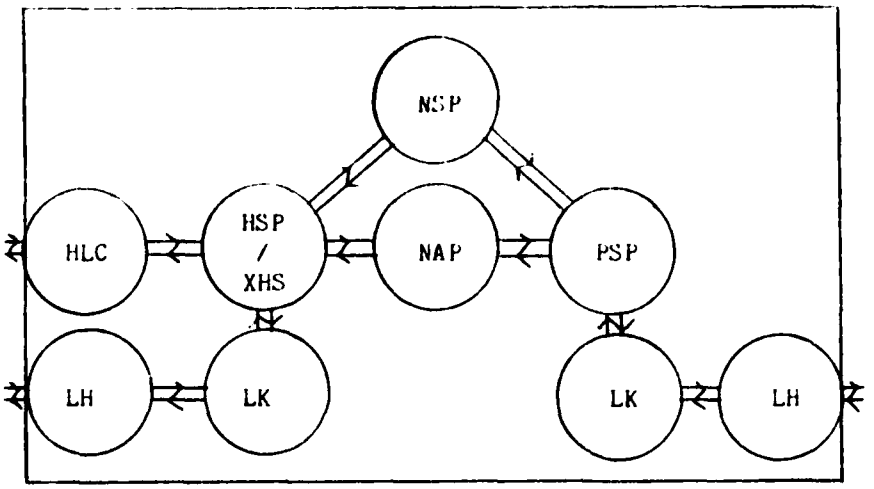


Figure 3 - New PPSN Protocol Structure



HLC - Simplified Link Controller (for in-house link protocol)
HSP - Block Host Support Package (handles Block protocol)
LH - X25 level 2 line handler (used in network as well as at interface)
LK - X25 level 2 link controller
NAP - Network Access Package (operates transnetwork protocol)
NSP - Network Support Package (handles network control and monitoring)
PSP - Packet Switching Package (handlers local switching functions).
XHS - X25 Host Support Package (handles X25 level 3)

Figure 4 - Communications Software Packages in Access Node

```
==========================================
I  Multi-address Packet Ident (255.)    I
I----------------------------------------I
I        Number of Destinations          I
I----------------------------------------I
I           1st Destination              I
I--                                    --I
I               Standard                 I
I--                                    --I
I                Header                  I
I----------------------------------------I
I           2cd Destination              I
I--                                    --I
I               Standard                 I
I--                                    --I
I                Header                  I
I----------------------------------------I
I                                        I
.                etc                     .
.                                        .
I                                        I
==========================================
I                User                    I
.                Data                    .
.                Field                   .
I                                        I
==========================================
```

Figure 5 - Multi-address Data Packet For X25 Virtual Calls


ATE
LME