Communications Research Centre

AN INTERACTIVE IMAGE COMMUNICATIONS SYSTEM USING PACKET SWITCHED NETWORKS

by

W.T. LALONDE AND C.D. O'BRIEN

Department of Communications
Ministère des Communications

CRC TECHNICAL NOTE NO. 701

OTTAWA, DECEMBER 1980

COMMUNICATIONS RESEARCH CENTRE

DEPARTMENT OF COMMUNICATIONS
CANADA

AN INTERACTIVE IMAGE COMMUNICATIONS SYSTEM USING PACKET SWITCHED NETWORKS

by

W.T. Lalonde* and C.D. O'Brien**

(Technology and Systems Branch)

* W.T. Lalonde is presently with Norpak Ltd., Pakenham, Ontario

** C.D. O'Brien is presently with Communications Research Centre, Ottawa

CRC TECHNICAL NOTE NO. 701

December 1980

OTTAWA

TABLE OF CONTENTS

FOREWORD

The Image Communications Program at the Communications Research Centre of the Department of Communications has, for a number of years, been investigating methods of communicating at minimum bandwidth. One area under particular investigation has been the communications of image in a conferencing environment where the image maintained in front of each user is identical yet modifiable by any member of the conversation. This common visual space so maintained behaves as a common writing pad for each user where anyone may add to or erase from the common image.

Techniques have been developed in order to maintain a common visual space using minimum communications bandwidth. The basic premise has been that computing hardware is getting cheaper much faster than enhancements in communications plant, so that it is economic to place as much sophistication in terminals as possible so as to use minimum bandwidth.

Experiments have been conducted by H.G. Bown, C.D. O'Brien and W. Sawchuk at Communications Research Centre using narrow band communications lines, specifically voice grade telephone lines to establish a common visual space for a simple two node situation. Much of this work has been done in conjunction with the Royal Military College at Kingston. A direct extension to this work is to multi-node communications systems. Communications Research Centre has been actively pursuing research in this field for a number of years. A chairmanship protocol was developed by C.D. O'Brien and G. de Courson at CRC in order to accommodate a multi-node image communications network maintaining a common visual space. The work reported herein by W.T. Lalonde and other work by R. Schwab at R.M.C. has refined the chairmanship protocol for use over packet switched communications facilities.

W.T. Lalonde has performed this work for a minor thesis at Carleton University under the direction of Professor J.K. Cavers and C.D. O'Brien of CRC. The major refinements developed are Error Recovery, Logon/Logoff and Election procedures required to maintain the integrity of the protocol and the state specification of the protocol developed to verify its correctness. Mr. Lalonde has also implemented the core of the protocol, using the X.25 packet switched Datapac communications system, in order that CRC can further experiment with the operation of the system.

C.D. O'Brien

December 1980

AN INTERACTIVE IMAGE COMMUNICATIONS SYSTEM USING PACKET SWITCHED NETWORKS

by

W.T. Lalonde and C.D. O'Brien

ABSTRACT

The role of computer-to-computer communication has become essential in today's society. Establishment of two-way image communication, between remote locations, has until recently suffered from the high bandwidth requirement associated with visual data.

Over the past several years, scientists at the Communications Research Centre have been developing a bandwidth reduction coding scheme for the transmission of image data over various communication media. A two node network for interactive image communication, operating over 300 baud switched telephone lines, has been demonstrated at CRC in Ottawa. The system is based on the concept of a common visual space, wherein an identical image is presented to both users. The image may be modified by either participant, the results of which will be seen by both.

This report outlines a proposed protocol of operation for a multi-node image communication system, using a packet switched network (DATAPAC) as the medium of inter-node communication. The objective of this protocol is to maintain synchronization between all of the participating nodes, such that the concept of a common visual space is extended to multiple users.

1. INTRODUCTION

To most people today, the ability to communicate messages over long distances is something which is taken for granted.

From the early days of the Pony Express, people have been able to communicate with each other in spite of the distance between them. The telephone provided a level of inter-personal communication by which people could be thousands of miles away in an instant of time. Television has given us the ability to communicate images, which may be accessed by anyone, simply by turning on a switch.

These things have become an integral part of our day to day lives. The use of two-way interactive image communication is not so commonplace, although the concept is familiar to anyone who has seen a Hollywood science fiction production. The ability to interact visually, between remote locations, is the essence of the project which is to be described in this report.

The project involves the implementation of a network of computer controlled graphics terminals which will provide a visual counterpart to the telephone network. Used in conjunction with the telephone network, it will represent a new level of audio-visual communication between remote locations. Users will be able to call anyone else on the network, interact visually with the other terminal and then disconnect their call with no more effort than "hanging up the phone".

The implementation of such a system implies that each user must have a means of connecting and disconnecting network communications. This may be accomplished by the use of what may be called a "secretary" program. This would be used to initiate calls to the network, or recognize incoming calls, notifying the user that someone wishes to communicate (i.e., ring the bell). The procedure here is functionally the same as the tasks performed by the secretary of a busy executive. The "secretary" will arrange conference calls, screen out unwanted callers and generally set up a "meeting" between those users who wish to participate. Once the "secretary" is finished the communications link will be established and the participants are free to interact.

As with a telephone conversation, the participants must establish what is to be "discussed". The format of the "discussion" may follow one of several scenarios. Each participant may recall the results of a previous "meeting", which will then form the "baseline" for the current session. As we are dealing with computer to computer communication, this will have the form of a program common to all of the participants. One of the participants may wish to communicate some new information. By default, this may be communicated directly, alternatively, the user may wish to communicate an already prepared document in the form of a new program, which will then form the base for the current "meeting".

"To implement a data network that is universally available and shared by many users, it is necessary to have a protocol that is agreed upon by the users as well as the telecommunications carrier. Acceptance of a standard protocol enables users with a diversity of terminals to access the network and communicate with one another. The wider the acceptance of a protocol, the more effective is the movement of data over constantly expanding networks" [1].

Commercially available packet switched networks have established such a protocol. Networks such as Datapac, offered by the Trans-Canada Telephone System, have been designed to become the basis of future data network communications. This fact, and the claim by these packet switched networks of virtually error-free communication, make them the most attractive choice as the medium through which a visual communication network may be implemented.

Given the long range goals of a visual communication system, as they have been outlined, this report will present the proposed protocol of operation of such a network, and give details of the initial implementation of this system by the Department of Communications, at the Communications Research Centre in Ottawa.

1.1 IMAGE COMMUNICATION SYSTEM

The progress and development of interactive graphics can be traced from its first implementation by I.E. Sutherland, who used a CRT display and a lightpen to construct graphical images while a computer maintained the requisite data base, to the present day complex, dedicated systems with a variety of interactive devices and high resolution, full color displays [2].

Generally, systems such as these are used in a stand-alone or single user application. The image communication system proposed in this report will require the interconnection of several graphics terminals, which will then form the basis of a "common visual space". This concept of a common visual space provides that an identical image is presented to each of the users, currently in the network. This image need not remain static, any user may modify the image with the knowledge that these changes will be communicated to all of the other users. The common visual space concept need not be restricted to the computer graphics environment, a number of people interacting directly, using a common blackboard would accomplish the same result. However when the users are geographically disposed, the need for an image communications network becomes apparent.

The design of a multi-terminal system for interactive graphics can embody one of two configurations, a central computer connected to many non-intelligent terminals, or a distributed system with a number of equally capable terminals connected together. An appreciation of the current trend towards decreased memory and processor costs and the desire to provide a terminal which could fulfill the dual role of network or stand alone operation, has resulted in the approach taken at the Communications Research Centre for the communication of interactive graphical images. This was to provide independent processing power and data storage capacity at each of the nodes and work towards minimizing the amount of information that must be communicated among the nodes.

Figure 1 illustrates the basic architecture of the image communication system. The development of these terminals has produced a graphics system which may be either single or dual processor based [3]. Figure 2 shows the functional configuration of a dual-processor based node. Although more expensive, the dual-processor configuration provides a system with much better response time and display file generation speed, since the computational load is split between two processors. One processor is dedicated to providing display capability and handling interaction interrupts, while the second processor is responsible for the execution of the applications program and the handling and communication of interactions.

The development of the dual-processor arrangement has facilitated the implementation of the interactive graphics network. The basic terminal may be driven by any computer which outputs information in a particular coded format. This set of coded output commands will then constitute the common "language" for graphical communication between all of the nodes on the network. The use of this graphics "language" may be considered comparable to the use of ASCII code as the standard of character oriented communication between computers. The design of the image communications network is independant of the actual "language" used by the terminals, the only restriction, is of course the requirement that all of the terminals be "speaking" the same "language", or have the facility to translate the dominant "language" into one which they are able to understand.

The "language" used in this implementation of an interactive image communication system is referred to as Graphical Task Instructions (GTI) [3]. When a user interaction occurs, through one of the interactive devices, such as a lightpen, a GTI is formed and communicated to an interaction handler (a software module). The interaction handler insures synchronization between the "HOST" and "SLAVE" processors. The host then interprets this interaction command and generates the appropriate drawing commands which are then sent to the display processor. A network protocol module is required to insure synchronization between all of the nodes in the network. This module is responsible for the communication of the interaction messages to the other nodes and for the acquisition of similar messages from the other nodes. Figure 3 diagrams the information flow resultant from a single interaction at one of the terminals. The small arrows illustrate the relatively small amount of information actually transmitted between the terminals, while the large arrows indicate the greater amount of information involved in modifying the image on the screen.

Thus the host processor may run an application program written in any high level language as long as the support software emits graphical messages in the current network "language" (in this case GTI). One such language called IMAGE was designed some time ago at the Communications Research Centre [4], in order to provide an application programmer with just such an ability to program interactions easily. This particular language has since been superseded by a commercially available counterpart under licence from CRC called

IGPL (Interactive Graphics Programming Language), which embodies certain enhancements but still retains the unique interaction control structures that simplify the programming of highly interactive graphics. This is the particular language used to write the applications programs used in this implementation of the image communications network.

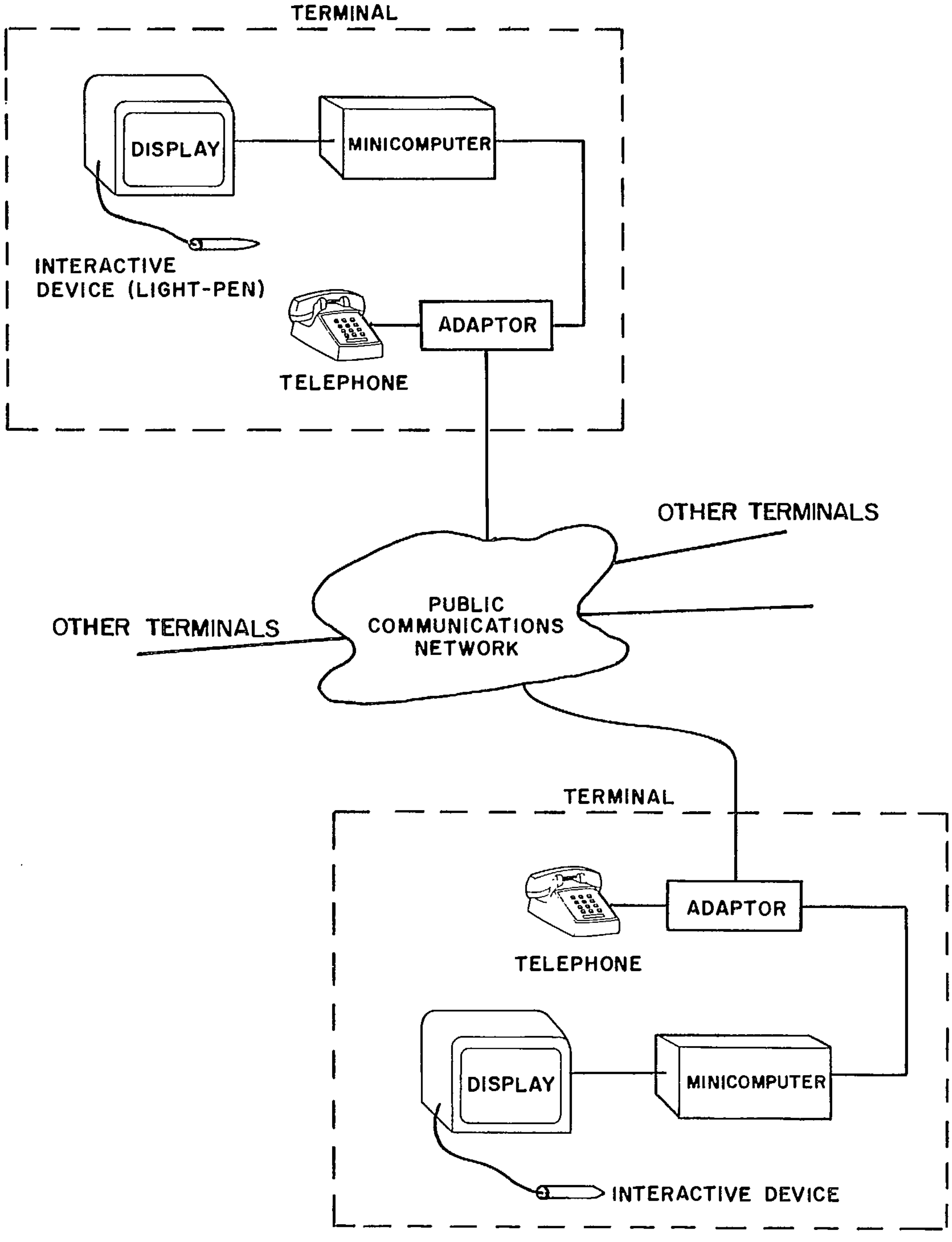


Figure 1. Basic Architecture of an Image Communication Network

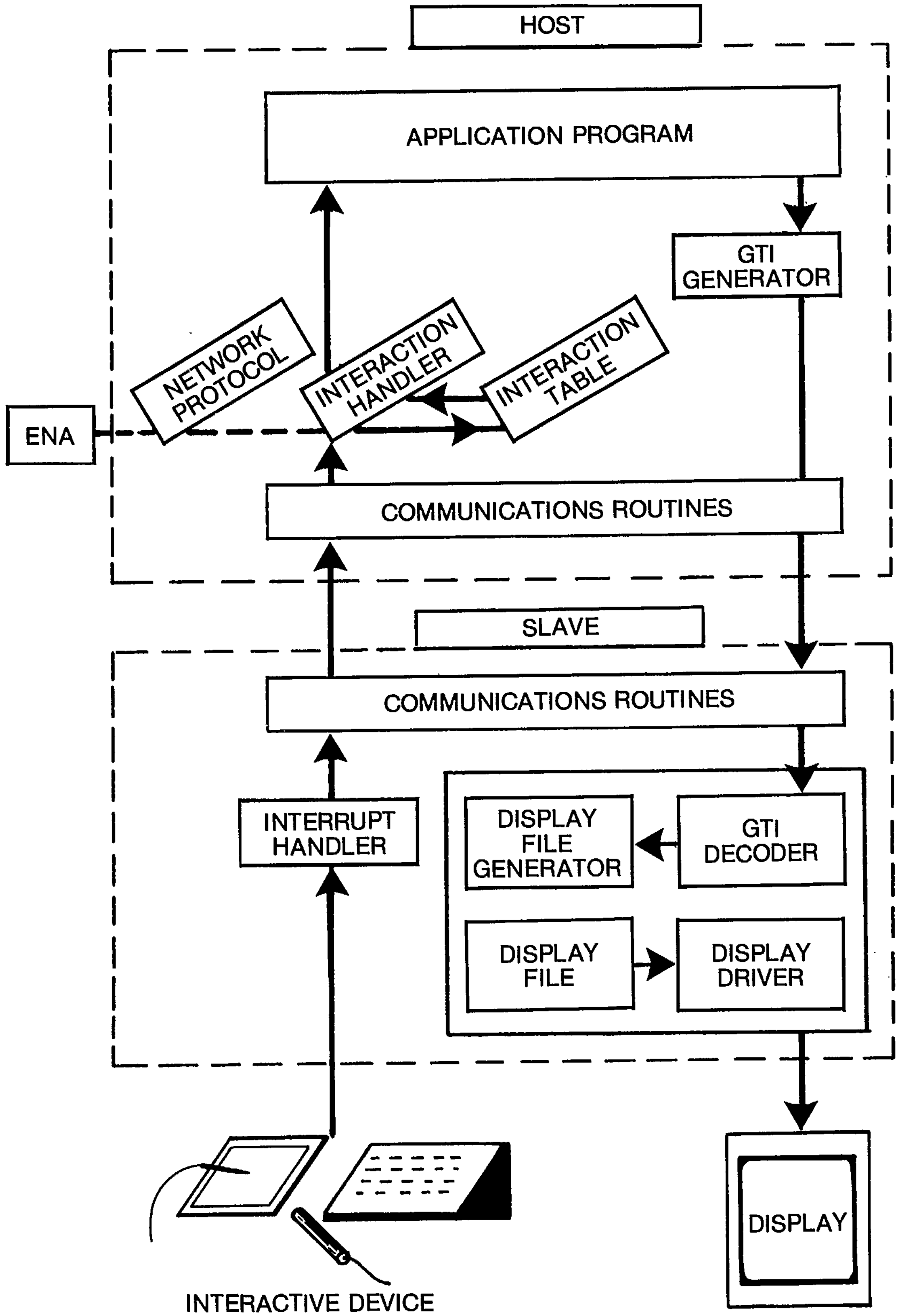


Figure 2. Functional Configuration of a Dual Processor Based Node

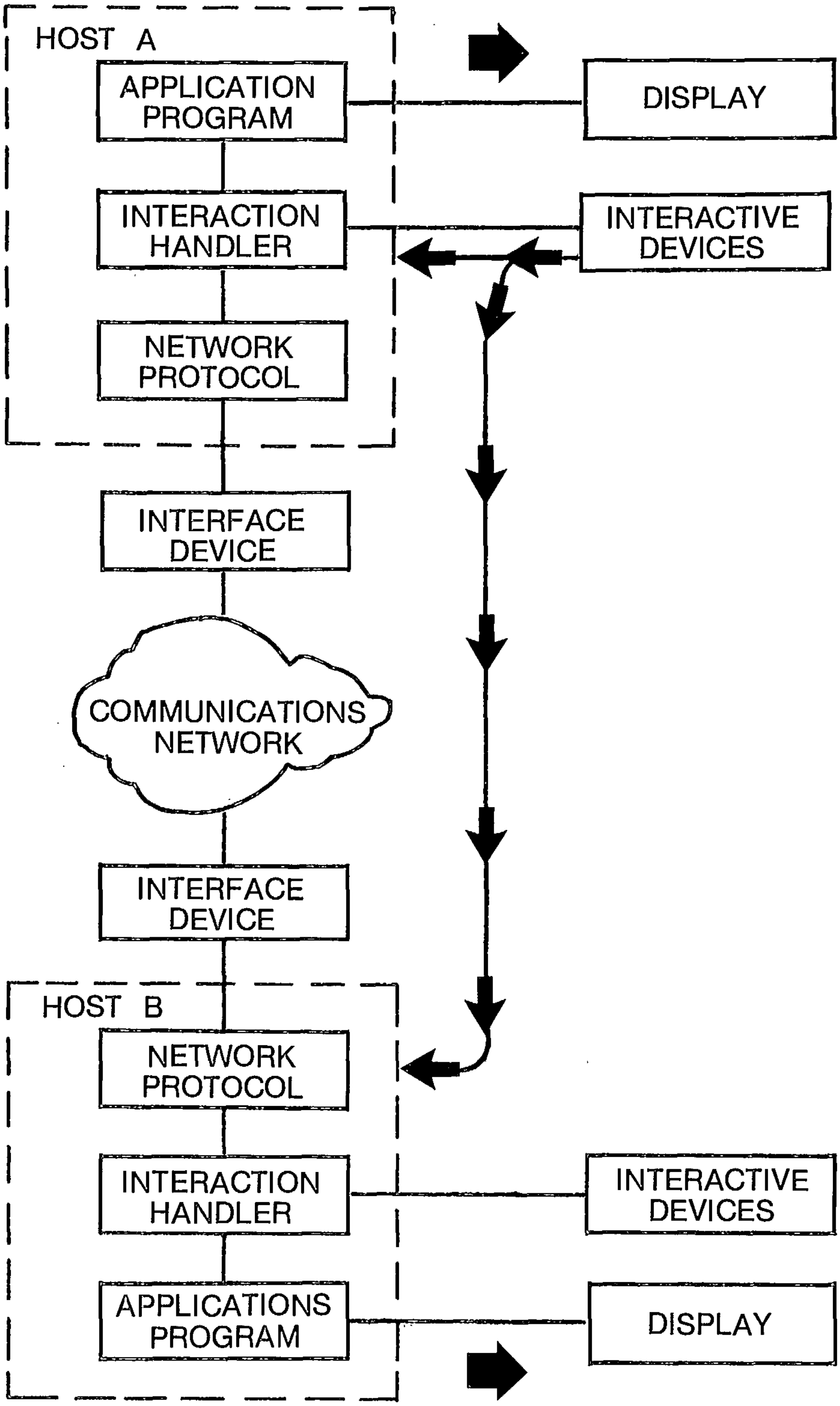


Figure 3. Interaction Information Flow

An integral part of the interaction handler when running in the network configuration, is the generation of a transaction table (interaction table). This will then be used when a participant wishes to join an existing network "conversation" after it has begun. With the interaction table the new user may then be brought up to the current level of activity. This same table will provide an "audit trail" for analysing a communication session that has already occurred. Possibly the most important function of the interaction table will be its use in error recovery procedures, whereby the nodes may re-establish synchronization.

This section has presented some of the basic requirements of the terminals to be used in the implementation of an interactive image communications network. Although some of the details refer to the specific implementation to be outlined in this report, the major functional blocks, a graphics applications program, an interaction handler, an interaction table, a display processor and a network communications protocol will be required by any terminal wishing to participate in an effective system.

The levels of protocol required in the implementation of an image communications system are shown in Figure 4.

The human protocol dictates the use of the common visual space. The rules which apply here will have to be established in advance and understood by all terminals intending to participate in the network. For example, a user must realize that if in an attempt to communicate information to the other terminals, some other interaction appears on his screen, he simply has to try again to communicate his idea. Etiquette must of course prevail, a user should not communicate "senseless" interactions simply to maintain control of the common visual space. The specific rules will depend upon the actual implementation.

The application language protocol is a function of the specific language used to generate the graphical code instructions which will create the image communication scenarios, (IGPL in this case).

The interaction handler protocol is required to maintain synchronization within the individual terminals graphics environment.

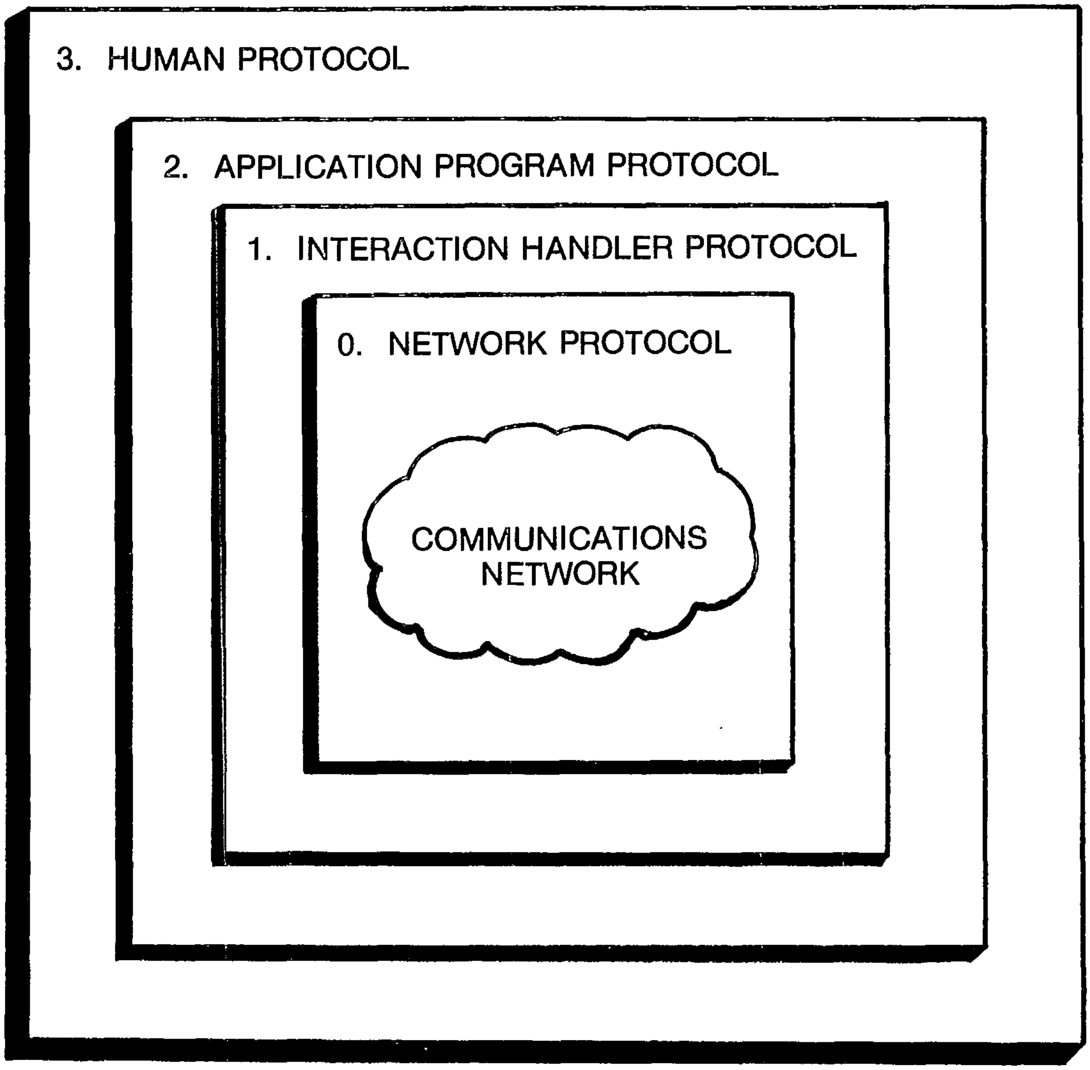
The final level, the network protocol, maintains synchronization of the entire network. This protocol should, at this level be independant of the type of network medium used to implement the image communication system.

1.2 PACKET SWITCHED NETWORKS

"Computer communication networks using packet-switching technology provide for the interconnection of data-processing equipment of any kind. Such systems, sometimes simply referred to as computer networks, may be viewed as multi-macroprocessors whenever the goals of resource-sharing are achieved. With the large-scale emergence of mini and microcomputers, it is now possible to envision building general or special purpose multimini and multimicrocomputers to be operated in a non-centralized manner. The need for automatic resource sharing arises here as in a similar way it does for multi-macroprocessor systems" [5].

The implementation of the concept of a common visual space in an interactive image communications system may indeed be considered a multi-macroprocess. The simultaneous operation of many individual graphics terminals sharing a single resource, in the form of a common visual space, is a situation which is well suited to the message oriented concepts of a packet-switched network.

In contrast to the more common line-switched techniques, whereby the message route must be set up prior to transmission, packet switched networks move a message through the network based on a destination address, contained within the header portion of a packet. The means to achieving the end result (successful transmission) is transparent to the user, the message simply enters the network and arrives at its destination. The sophisticated protocol of the packet switched network takes care of such problems as data integrity, retransmissions, collisions and deadlock on the transmission link. Thus the overall communication between the nodes on a packet switched network is reduced to an END to END protocol.



LEVEL	PROCEDURE	INFORMATION UNIT
3	COMMON VISUAL SPACE	IDEA
2	APPLICATIONS PROGRAM	GRAPHICAL PRIMITIVES
1	INTERACTION HANDLER	GRAPHICAL CODE INSTRUCTIONS
0	NETWORK PROTOCOL	MESSAGES

Figure 4. Levels of Protocol for an Interactive Image Communication System

Commercially available packet switched digital communication networks support a standard interface protocol. This protocol is called X.25 and has been ratified by CCITT (International Telegraph and Telephone Consultative Committee) as an international standard for accessing public packet switched networks. Each terminal on the network must have a software or hardware package which understands the X.25 protocol and interfaces with the host computer.

The levels of protocol required to achieve the specified END to END performance are indicated in Figure 5. There are two possible methods of providing an X.25 interface: having the X.25 implementation imbedded in an operating system, utilizing the capabilities of the host computer, or building a separate X.25 interface device, with some computing power of its own.

Each method of implementation has its own relative merits. Considering the implementation based on a protocol imbedded in an operating system we have the following trade-offs.

ADVANTAGES	DISADVANTAGES
— hardware facilities are available immediately from the host computer.	— support of the communications protocol by the host CPU may unduly degrade CPU performance.
— the implementation is easier to write by using high level languages.	— the operating system must have a foreground/background or multitasking capability to support the interrupt structure of the X.25 protocol.
— the cost for a single implementation (primarily development costs) may be higher, but duplicating costs will be very low.	— a special clock or other timing facility is necessary to support X.25.

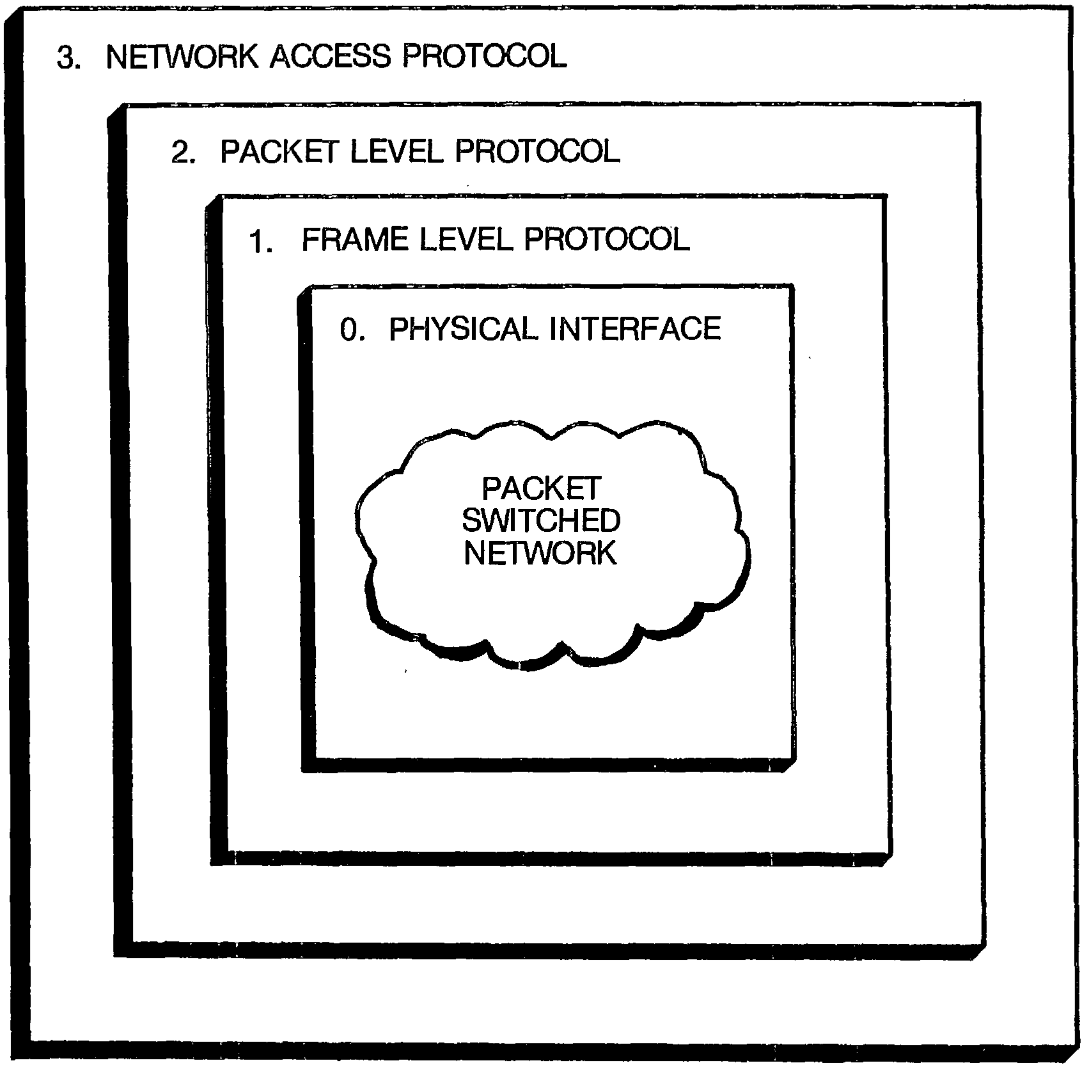
Only a manufacturer of an operating system may reasonably implement an X.25 protocol imbedded in the operating system, because maintenance or updates to the operating system would change the details of the imbedded protocol and thus require maintenance or update to the X.25 implementation.

The use of a separate "X.25 machine" has advantages which are generally opposite to those listed above. The major advantage of a dedicated interface machine is that it is portable, thus once it has been built it may be interfaced to several different host computers and can be maintained independently from them. The frame checking and other time consuming tasks performed by a separate X.25 interface machine are tasks which do not need to be performed by the host computer and therefore cannot degrade performance.

Thus, in light of the long range goals of the image communications network, specifically terminal independence [3], the choice of a separate interface machine would seem to be the logical one for the network implementation.

This was indeed the choice made for this initial implementation of the network. A commercially available device, manufactured by ENA Datasystems, was selected and a "front-end" interface adapted to it to conform with the hardware requirements of the particular host computers used in this implementation, (DEC PDP/11).

The use of this separate interface device then reduces the network communications to a simple message oriented protocol as required by the image communication network specifications.



LEVEL	PROCEDURE	INFORMATION UNIT
3	USER APPLICATION INTERFACE	MESSAGE
2	CALL SET UP, CLEARING DATA TRANSFER	PACKET
1	ERROR CORRECTION	FRAME
0	CONNECTION TO A TRANSMISSION FACILITY	BIT

Figure 5. Levels of Protocol to Provide End to End Message Synchronization

2. NETWORK DESIGN SPECIFICATIONS

From the introductory specification of the requirements of an image communication system, the function of the network protocol to be outlined in this section seems clear. This function is simply to maintain synchronization of the network by providing an interface between the individual graphics terminals and the packet switched network, providing a protocol of operation which will support the concept of a common visual space.

Figure 6 illustrates all the levels of protocol required to produce an interactive image communication network.

Levels 5–7 are a function of the individual terminals used in the implementation of the network. It is only necessary that this level be able to produce and receive coded graphical messages which are of the same format recognized by the network as a whole. Internal synchronization procedures within this level should be such that they are transparent to the network.

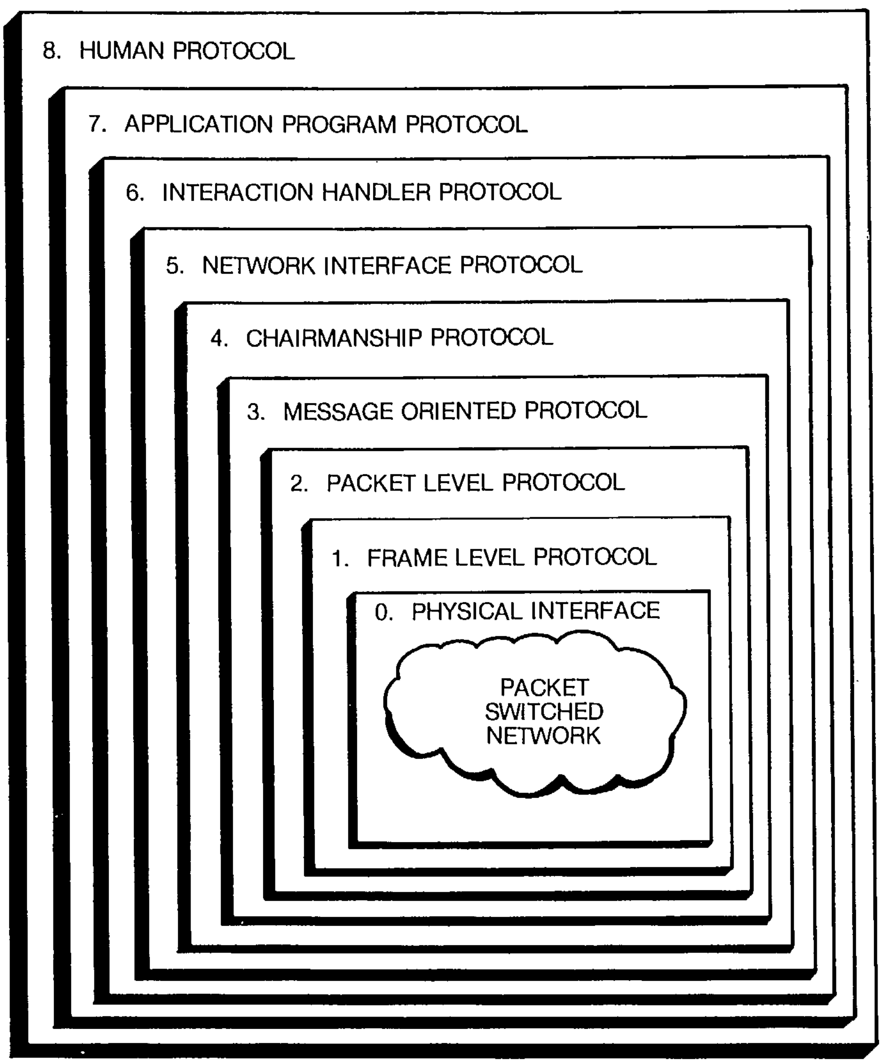


Figure 6. Levels of Protocol for an Interactive Image Communication Network

Levels 0–3 are provided by the separate interface device and as such do not appear in the specifications of the network requirements. The function of these levels as a group is simply to insure an error-free transmission environment. As far as the network protocol is concerned the medium of communication is transparent and this bottom level need only be considered as a means of transmitting and receiving messages.

Level 4 is the actual network control protocol, which has been called a chairmanship protocol. Levels 5 and 3 overlap into the overall network protocol such that the chairmanship protocol is able to interface with the other levels above and below it.

Following is the design specification for the network protocol module required to implement an interactive image communication network.

2.1 END TO END PROTOCOL

The network as presented to each of the individual terminals in the image communications system, is required to provide an END to END protocol to control the flow of interaction data between the nodes of the system.

As discussed in the introductory section, this system only operates if absolute synchronization is maintained throughout the network. Order is important, all interactions must be processed by all terminals in exactly the same order. To preserve this order, a terminal cannot simply process an interaction at its own node and then communicate this fact to the other nodes. There is always the possibility that some other terminal is processing a different interaction at the same time, thus resulting in a confusion of order.

Synchronization is ensured by the use of an END to END protocol, which informs each terminal of the status of each of the others. If a collision occurs in protocol control messages, it is resolved by giving priority to one of the interaction messages, such that synchronization is maintained.

The objectives of an END to END protocol are:

- after an interaction, only carry out the corresponding action when the data has been sent and received correctly by the other nodes,
- only send data when the receiving node is ready to accept it,
- in the case of an unsuccessful attempt at network communication, abort the interaction and restore the transmitting node to its state before the interaction. (i.e., ignore the interaction if it can not be communicated),
- arbitrate in case of contention over control of the communications link,
- provide timeout functions, or some other facility to prevent deadlocks,
- ensure data integrity, by providing or making use of, lower level protocols for data error checking and retransmission of data.

Thus these are considered to be the design specifications for the network protocol as they appear at level 5 of the overall protocol structure.

Following is a state diagram representation of this END to END protocol, (Figure 7).

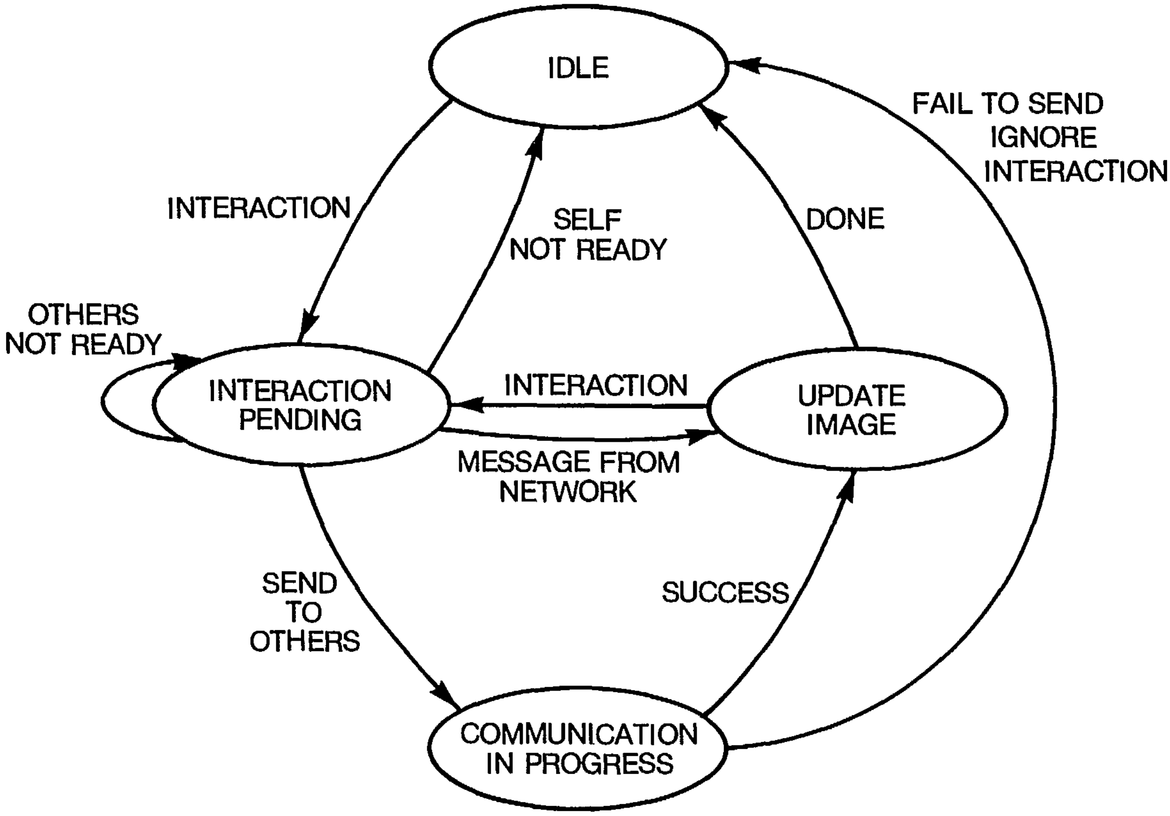


Figure 7. State Diagram of End to End Protocol

2.2 CHAIRMANSHIP PROTOCOL

To accomplish the END to END protocol required to maintain synchronization of the nodes in the graphics network the following protocol was proposed.

This protocol has been called a chairmanship protocol in that it operates in a similar manner to the maintenance of order by a chairman in a human meeting. In the normal operation of the visual communication system, one terminal performs several interactions in a short time span before "giving up the floor" to another terminal. Because of the nature of the dialogue, it appears best to arbitrate conflicts which might cause the system to lose synchronization in a manner similar to the dialogue flow. That is, allow the terminal to finish what he has to say before interrupting him.

An END to END protocol which operates in a similar manner to a formally chaired meeting between people is described below.

The Speaker –

A "chairman" (also termed the master) is in control of the dialogue flow and is responsible for designating which node may "speak"

Only one node may "speak" (that is, sent out interaction messages) at one time. The "chairman" specifies this by "giving the floor" to a specific node.

The node which "has the floor" may send out interaction messages and assume all other terminals are listening and responding. Since the packet switched network takes care of data integrity at a lower level, no acknowledgements are required. Thus network delays do not limit the response of a particular terminal node. The terminal may handle another interaction immediately after transmitting an interaction message, even while the first message is "in the network" being transmitted to the other nodes. A stream of such messages could be flowing over the network resulting in the other nodes of the system following a rapid series of interactions with a single fixed time delay.

The Listener –

The other nodes of the system "listen" to the node which "has the floor" and processes its interaction messages.

If a participant interacts with an input device on a "listening" node, a request is sent to the chairman to "have the floor". If this request is denied the interaction is ignored. Requests to "have the floor" are acknowledged by a specific message from the "chairman" to all nodes describing who "has the floor". Reception of any messages other than an acknowledgement that the terminal was "granted the floor" is considered a rejection of the request.

The "chairman" may be appointed by one of several algorithms. In the simplest case the "chairman" is appointed at the commencement of the conversation ("meeting") and remains fixed for the duration. This has the disadvantage of making the network asymmetrical by adding an additional load to one node specifically. This scheme would also require a special case of the general protocol to allow the "chairman" to "leave the meeting". These problems may be overcome by allocating the "chairmanship" dynamically as described below.

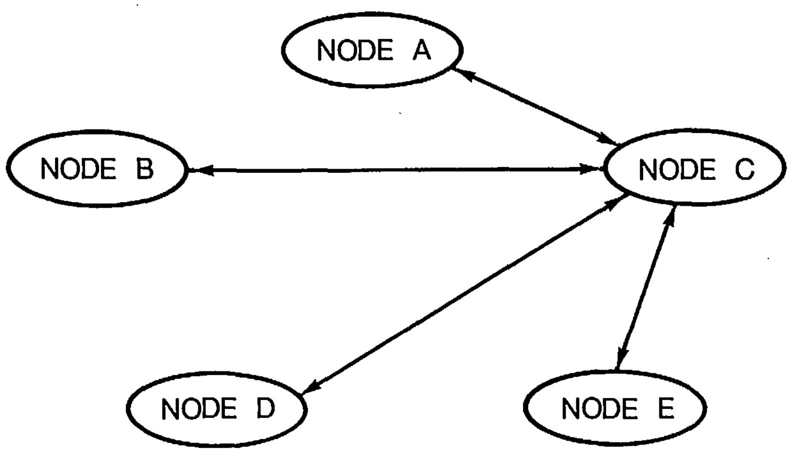
When a terminal is "granted the floor" to "speak" he is also granted the "chair". In other words the "chairman has the floor". This prevents problems from occurring due to communication delays between the "chairman" and the "speaker". The "chairman" knows when the "speaker" has finished communicating as they are one in the same terminal.

With each node appropriately "ranked", the "chair" is assigned initially to the node with the highest "rank", however any other node may become "chairman" simply by requesting the "chair".

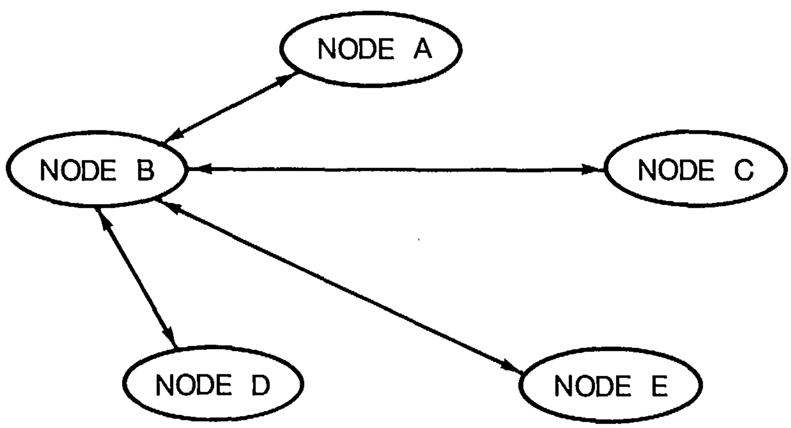
The "chairman" may then "leave the meeting" at any time, an "election" process will assign the "chair" to one of the other nodes (it matters not whether he has anything to say at the present time).

From these specifications we can see that at any given instant of time we have what is classically called a "star" type network, where one node is at the centre, in control of all of the communications. All nodes must communicate through this central node to modify the common visual space. However, due to the dynamic nature of the allocation of the control of the network (i.e., chairmanship), at another instant of time we have a physically different "star" network. The overall view of a network of this type with N nodes would appear to be C(N,2) point-to-point virtual circuits, (i.e., 3 links for 3 nodes, 6 links for 4 nodes, etc.). This has been shown graphically in Figure 8.

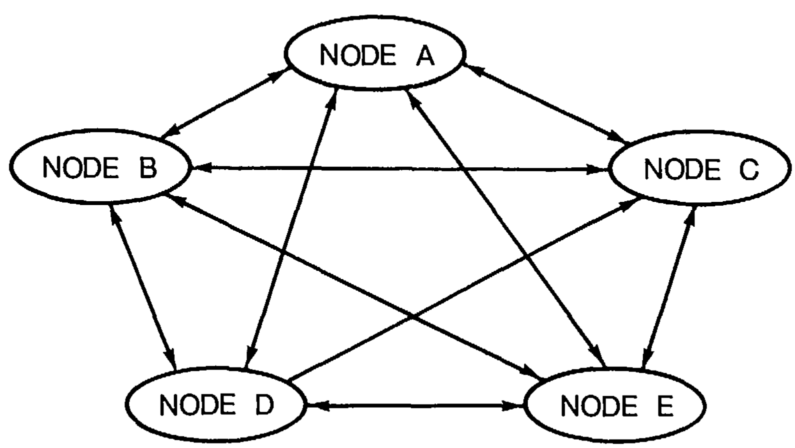
The chairmanship protocol is thus used to provide the synchronization and aribtration facilities required by the END to END protocol. Vital to the implementation of this protocol is the definition of two overall network state variables, CHAIRSTATE and RUNSTATE. When RUNSTATE is in the NORMAL condition for the communication of graphical messages, the state of CHAIRSTATE will govern the flow of interaction messages. When RUNSTATE is in any state other than NORMAL or ELECTION, the status of CHAIRSTATE will remain locked in either MASTER or NOTMASTER, depending upon the conditions at the time RUNSTATE changed. Before returning to the NORMAL RUNSTATE from any of the other states, the network must pass through the ELECTION state, whereby the current status of the network is confirmed. It is here where " roll call" is taken whereby each of the participants must "stand up and be counted" in order to



STAR NETWORK TIME=T_1



STAR NETWORK TIME=T_2



APPARENT NETWORK WITH DYNAMIC "CHAIRMANSHIP" ALLOCATION

Figure 8. Multi-Node Network Configurations

confirm the actual number of active members currently in the "meeting". Once the population has been "enumerated", the "election" may proceed to reassign the "chair" to the highest "ranking" member currently in the "meeting". When this has been accomplished the overall network returns to the NORMAL RUNSTATE, where interactions may then again be communicated.

Following are the finite state descriptions of CHAIRSTATE and RUNSTATE, (Figures 9 and 10).

2.3 EXTENSIONS TO THE BASIC PROTOCOL

As mentioned, the chairmanship protocol will provide for network synchronization under NORMAL operation. However, allowing dynamic allocation the "chairmanship" when RUNSTATE is not in the NORMAL node would generate an excessive amount of communications overhead and surely lead to loss of synchronization. For this reason the following "special case" protocols are defined for the non-normal states of operation, (i.e., RESTART, LOGON, LOGOFF and ELECTION).

2.3.1 ERROR RECOVERY

The assumption by the chairman that all of the nodes in the network are listening and responding to his communications, is vital to the operation of this protocol. The fact that the chairman does not wait for acknowledgements of his communications may imply that errors in receiving information by the listeners may go unnoticed by the chairman. This can be overcome by having the chairman listen only for negative acknowledgements. These would come in the form of requests to be master with a message requesting the chairman RESTART the meeting. The chairman will not, under these circumstances relinquish the "floor", since the listener requesting the RESTART will not have the most up to date copy of "the minutes of the meeting" (i.e., the interaction table). Because the chairman may not be exactly sure when the requesting member of the meeting stopped listening, he might "read the complete dialogue transcript". This must be done in order to insure synchronization of all of the nodes in the network. This complete replay of the network

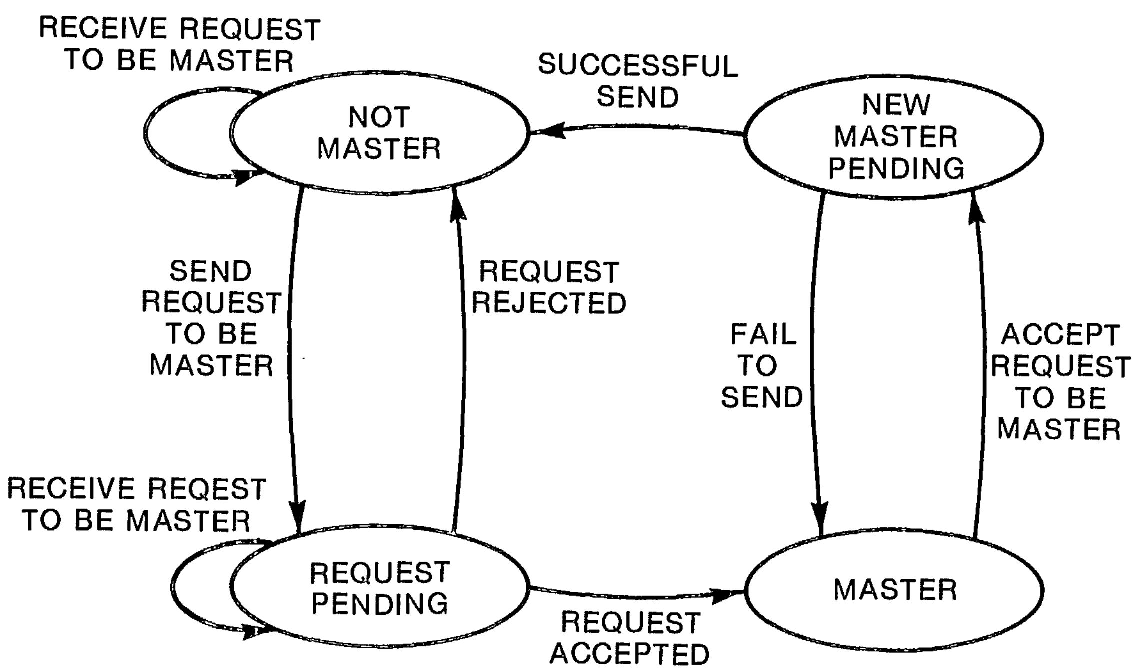


Figure 9. State Diagram of CHAIRSTATE

conversation should not however pose a major problem in terms of overall system response. This is based on the same assumption that produced a protocol which does not require acknowledgements to be sent for every communication. This assumption is that the packet switched network is for all intents and purposes error-free and the probability of having to restart due to a loss of information is very low. (DATAPAC documentation suggests an undetected error rate of one in every 5×10^8 packets.)

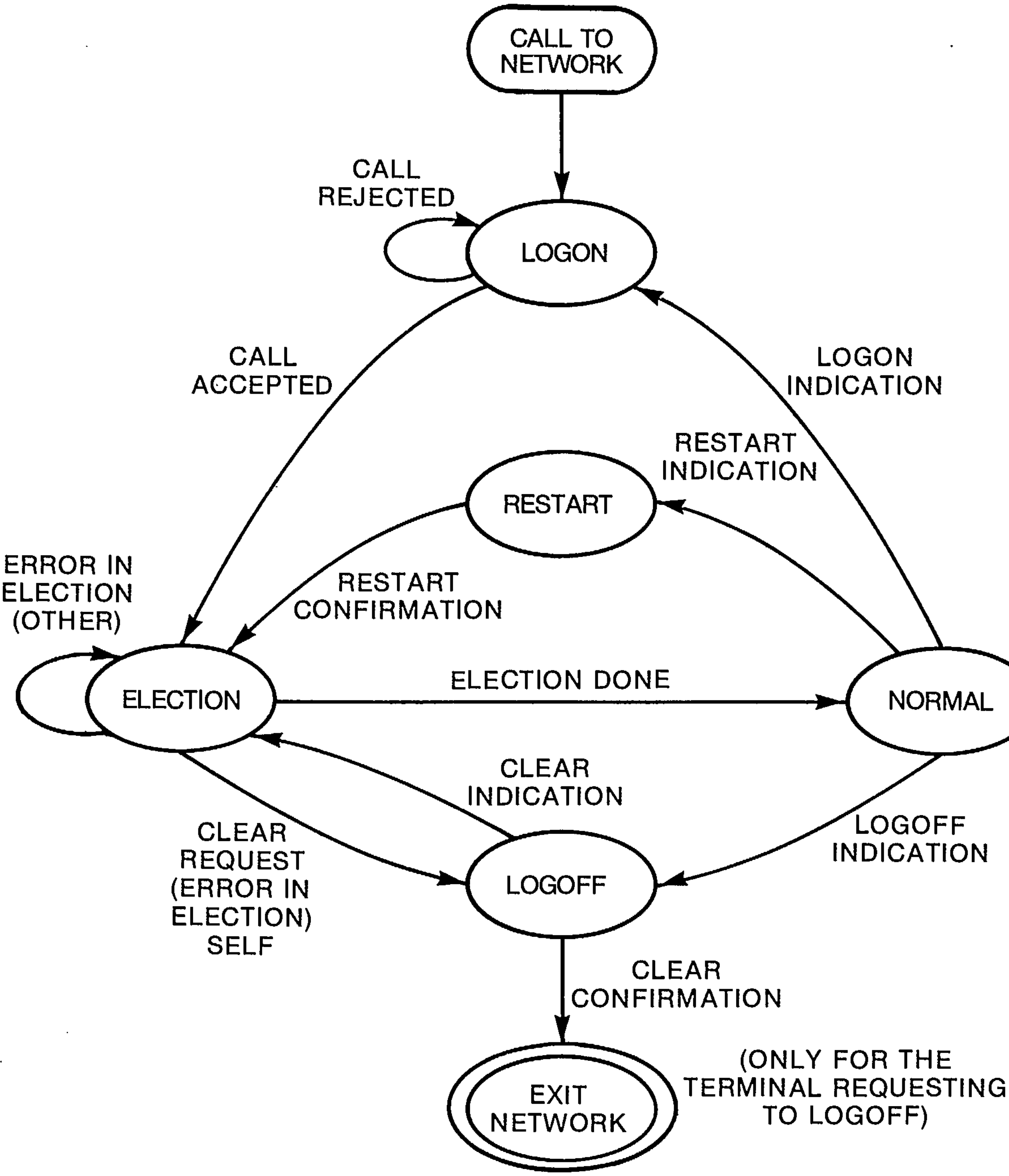


Figure 10. State Diagram of RUNSTATE

This RESTART condition may actually afford the chairman an opportunity to "edit" the records to reflect the current status of the meeting. Throughout the course of the network dialogue certain interactions may have been repeated or certain originally displayed objects may have moved more than once or completely erased, these interactions then become redundant to the current status of the common visual space, hence they need not be recommunicated, bringing a member back into synchronization with the other members of the meeting. Since the interaction table is in fact a storage butter of finite size, this RESTART condition may also be used by the current chairman to compress the data stored in the table, before the buffer overflows and information is lost. Under these conditions the master need only transmit a RESTART indication with a parameter saying that it is for the purposes of updating the interaction table. Since this will be done when RUNSTATE is initially in the NORMAL mode, all nodes should have a valid copy of the interaction table. Thus, the information need not be transmitted over the network. Each node is running the same application program so the update procedure may be performed at the individual nodes. If there is any confusion as to the success of the update by any of the nodes, they may simply request a RESTART(ERROR) in the same manner as before, by a request to the master. If at any time the current "chairman" is uncertain as to whether or not anyone is listening to him, he may initiate a RESTART(ERROR) for all of the nodes.

Due to the nature of the packet switched network, specifically its dynamic routing capabilities, message transfer delays cannot be assumed to have a constant timing interval. Thus a possible conflict may occur when one node has its request to be master accepted and immediately begins communicating further interactions. As a result of network delays some of the other nodes may not have received notice of the transfer of the "chair" when these new interactions arrive. Since only the master ("CHAIRMAN") is able to communicate interactions, these messages would normally be ignored by those nodes unaware of the change and synchronization would be lost. By including the current master status in all interaction messages, this situation could be detected by the receiving nodes, such that when any node is "confused" as to the current status of the "chair" a network RESTART could be generated to insure synchronization. An appropriate time delay programmed into a new "chairman" delaying him from communicating his first interaction would make the probability of this conflict and its associated recovery procedure very small.

In NORMAL mode of operation messages are generally short and of a synchronous nature, thus in the RESTART state a slightly more sophisticated flow control procedure will be required to control the X.25 interface device. This is necessary in that communication of the entire interaction table should be made as asynchronous as possible to reduce the time spent in the RESTART state (i.e., no interactions allowed).

The actual implementation of the error recovery procedures need not be totally defined at this point, re-synchronization protocols are available [7] to perform the task of ensuring overall network synchronization when an error is detected. The critical point is that the need for such a procedure is recognized by the NORMAL mode protocol.

2.3.2 LOGON Procedures

As outlined in the introductory philosophy of the image communications network, LOGON/LOGOFF procedures will generally be performed by what is called a "secretary" program. The function of this program will be to perform the tasks associated with network connection and disconnection.

In a multi-user operating environment, the secretary may be a separate "task" which is running at all times. Thus to initiate network communications a user simply has to "place the call". In a single user operating environment, the secretary must be an integral part of all of the application programs (perhaps part of a library) such that the currently running program may be interrupted to signal an incoming call.

To form a network requires at least two nodes, one of which must be declared "chairman". The assignment of the "chair" will be based on the "rank" of the nodes wishing to form a network. A very simple "ranking" system may be established based on the actual network "address" of each of the members. The "chairmanship" of the current "meeting" may then be established by arbitrarily assigning the "chair" the member with the highest "rank" (i.e., the lowest address number). Once initiated, the "chairmanship" is free to change in accordance with the procedures outlined by the chairmanship protocol.

Thus a user may call any other user and form an image communication network. It will be up to the "secretary" of the called node to inform the user of the call, such that a decision may be made as to whether or not to accept the call. If both nodes are ready to participate an "election request" is made and the "chairmanship" is established.

If a terminal calls another terminal which is currently in a "meeting", the called terminal may then invoke his priviledge to request to become master. This same message contains a message indicating that someone, whose "name" is also contained in the message, wishes to join the "meeting". Since the calling terminal has not received an election request he will be waiting for further communication. If the request by the called terminal to become master is accepted, the "chairmanship" does not transfer. Instead the current "chairman" calls the potential member. Once the call has been established, the "chairman" then sends a LOGIN INDICATION message telling the potential member how many calls to expect (the current number of members minus two). To insure that all members have indeed welcomed the new member, an acknowledgement of the completion of the call connections must be received by the master before an ELECTION may be called. Once the LOGON CONFIRMATION has been received the master then sends a LOGON INDICATION message to all of the current members (still in the NORMAL RUNSTATE), included in this message is the "name" of the new member. The particpant nodes then enter the LOGON state, place their calls and return to prepare for an ELECTION. Any attempts to interact during these proceedings will be ignored, since the "chairman" has not been "listening".

Once the new member has been accepted, he must be brought up to date by the "chairman". This may be done simply by initiating a procedure similar to RESTART(ERROR) directed at the new member. When the "chairman" receives confirmation of a successful initialization, he then proceeds to the ELECTION state, at which time all other nodes should be in the same state. Once the ELECTION is completed, the new member may wish to communicate information by following the normal procedure of requesting to become "chairman".

If the initial request to become "chairman" (sent by the member who was first called) is rejected, the member issuing the request may try again or clear the call connection with the potential member, forcing him to try again at a later time.

Since we are not dealing with interaction information, during this phase, the "secretary" may also be used to communicate general messages between the terminals. Thus when an audio link is not used in conjunction with the interaction session, messages may be sent during the LOGON phase to pass information between the terminals. This type of procedure would however, only be appropriate when a network "meeting" was initially being formed.

2.3.3 LOGOFF Procedures

In order to leave the "meeting", each member must ask to be "excused". To do this the member is required to be the current "chairman". This is guaranteed by the NORMAL RUNSTATE protocol, in that LOGOFF procedures are not initiated until the "secretary" program has been called. As this call is made during the execution of an application program and the chairmanship porotocol dictates that only the "chairman" may communicate an interaction, the member would have become master to reach this point. The master then sends a message to the other nodes indicating that he wishes to LOGOFF. The other nodes then acknowledge this message, disconnecting the appropriate communication channel and returning to "elect" a new "chairman" from the nodes remaining in the "meeting".

If a member "leaves a meeting" without requesting to be "excused", possibly due to a hardware or similar failure, the meeting will still continue in proper synchronization in his absence. Since the member who dropped out cannot request to be chairman, since he is out of the meeting, there is no corruption in the operation of the ongoing conversations. The absence of this member will be detected at the next election phase. If the absent member requests to rejoin the meeting he will be brought up to date from the interaction table. If the chairman drops out inadvertently, this will be detected upon the next request to be master which is unacknowledged in either a positive or negative sense, and will therefore be cleared up by a recovery procedure.

2.3.4 ELECTION Procedures

The exact protocol of operation of the ELECTION state is not to be defined in this report. One possible procedure has been proposed by G. LeLann [5], although this protocol may not be used specifically, the general guidelines of this proposal should be followed.

The vital function provided by the ELECTION state, that of assigning absolute control of the "meeting", dicates that consideration be given to the "death" of any member in the "meeting", during this critical phase. As with the RESTART state, the time spent in the ELECTION state should be kept to an absolute minimum, to reduce the period when interactions are not allowed.

The basic philosphy of the ELECTION state would be as follows. If any member does not respond during the "enumeration" phase of the ELECTION (possibly timeout controlled), he is left off the "voter's list" and is no longer considered to be part of the "meeting". The most critical potential failure would be the "death" of the current member who is to be "elected" as the next "chairman". This should cause no more of a problem than to introduce a slight delay in the ELECTION phase while the "voters" cast their "ballots" again (for the next highest "ranking" member).

The ELECTION phase should not be such that a "tie vote" situation can occur, in other words when done there should only be one master (the same for everyone).

2.4 NETWORK ACCESS PROTOCOL

The use of a separate interface device for the network has reduced the network access protocol to a very basic "flow control" protocol. Simply stated, this means that messages flow across the interface (to and from the network), only when they are allowed to do so by the host node. Commands such as FLOWGO and FLOW STOP are used to control the message flow and provide the synchronous communication required by the interaction handler protocol.

The interface device is also able to provide a BROADCAST function which will enable the current master to communicate an interaction simultaneously to all of the other nodes, with only a single transmission to the interface device.

The communication is done via "packets" of information shown in Figure 11, through routines which may be designated simply as SENDMESSAGE and GETMESSAGE. SENDMESSAGE merely transfers the messages to the network, regardless of whether they are control packets or interaction messages packets. The GETMESSAGE routine however, must be able to differentiate between the two types of packets, depending upon which RUNSTATE the node is currently in.

3. IMPLEMENTATION

The initial implementation of this multi-node image communications network is to be done based on the NORMAL RUNSTATE only. Network interactions which would result in the generation of the RESTART condition will be indicated simply by a message indicating the event has occurred. Since RESTART generally indicates a FATAL network error, the current network dialogue will have to be terminated.

This simplified implementation will provide an indication of the validity of the assumption of an error-free network.

Since RESTART will not be supported initially, the LOGON/LOGOFF states will have to be restricted to a one time operation. This means that only the users initially "logged on" will participate in the interaction dialogue, and when one user "logs off", all users will "log off".

As the chairmanship protocol applies only to the NORMAL mode of operation, this simplified implementation was deemed to be a valid test of a network operating under this protocol.

Both the LOGON and LOGOFF procedures, will be performed by a single "secretary" program (SECTRY), which at this time requires the users to have some idea of the overall network configuration (i.e., number of nodes to be connected and their assigned numbers).

3.1 RUNTIME ENVIRONMENT

The runtime environment for the trial implementation of the multi-node image communication system just described consists of a five node network. This network will be formed within a "closed user's group" utilizing the DATAPAC packet switched network. The initial phase of the implementation will consist of a two node version of this network. The nodes are both located at the Communications Research Centre, in Ottawa. The remaining nodes are located at the Defence Research Establishment (Ottawa), Royal Military College (Kingston) and the University of Manitoba (Winnipeg).

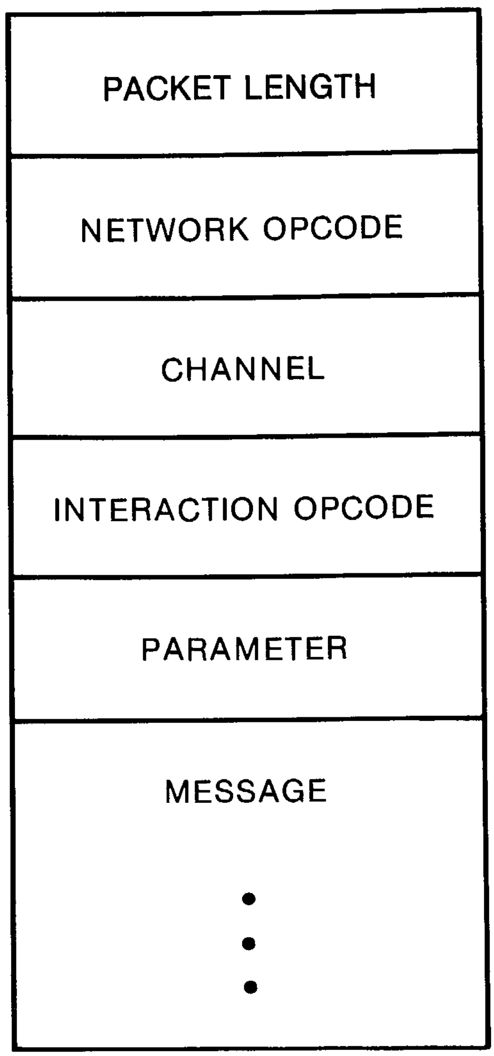


Figure 11. Sample Data Packet

The two terminals for the initial system consist of PDP-11/40 host computers, connected to Norpak RGP5000 raster scan display processors (slaves). Although the development of the network is being done with consideration to a multi-user operating environment, the initial implementation was to be done under the single user operating environment of RT-11. The ENA X.25 interface device is connected to the host via a modified DR-11C parallel interface.

As mentioned earlier, the application programs written for the image communications system, make use of the special interaction language IGPL, developed primarily for this purpose. The language allows interaction sequences to be clearly specified, and makes isolation of the driving "packets of interaction information" very easy to perform.

The information "packets" are of course, the GTIs generated by the IGPL language. With reference to Figure 3 (Page 6) these GTIs are of two general forms. The large arrows represent "forward" GTIs, while the smaller arrows represent "back" GTIs. The forward GTIs generate the drawing commands which eventually result in the images being displayed on the screen. The back GTIs represent communication of an interaction interrupt. The back GTIs themselves consist of two types, UNPOLLED and POLLED.

UNPOLLED GTIs are unsolicited GTIs which represent an interaction with one of the OBJECTs defined in the IGPL program. For example, a "lightpen" strike on the TARGET area of the sample IGPL program in Figure 12, would generate an UNPOLLED "back" GTI, which would communicate to the host the tag number of the particular OBJECT upon which the strike was made. This would generate a sequence of "forward" GTIs to update the image on the screen, according to the procedure specified by the ACTION block associated with the OBJECT TARGET.

In this case this process results in the eventual return of a POLLED GTI in response to the host's request for the current MARKER LOCATION. The result is a smaller yellow area being drawn on the larger blue TARGET area at the current MARKET coordinates.

Thus with a program such as this, in the network, several users could be trying to draw a yellow box on the TARGET area at the same time. The network protocol would ensure that only one box appears at any one time, at the coordinates specified by the current master ("chairman").

Although this is a very simplistic example the same basic procedure would follow for more complex programs involving many more OBJECT blocks.

3.2 NETWORK SOFTWARE SPECIFICATION

The design of the network interface software has been governed mainly by the requirements of the IGPL applications program structure. The end result of this design is a software module which will interface to the interaction handler module in exactly the same manner as any other network protocol modules. Figure 13 illustrates how the various software modules for the different network protocols interface to the "Interaction Handler". As a result of this compatibility constraint, modifications to the IGPL host were not to be considered and the network module would have to accommodate the IGPL language structure in order to provide the required END to END protocol. This definition of the host interface requirements was to be the first phase of the overall software design procedure.

An attempt has been made to provide finite state descriptions for each of the sub-modules required in the implementation. In essence, each of the five main routines may be considered states of the overall graphics program, being entered on some condition and returning with some conditions set. However, any attempt to depict these states as they may appear within the IGPL language itself, would add little to the clarity of the description. "The problem is basically that large state diagrams, although technically an exact representation of an interaction dialogue, are difficult to read and therefore their specification error-prone" [4]. Thus the five interface modules required are described only in terms of their entry and exit conditions.

3.2.1 Host Interface Specifications

The interaction interface to the host application program is accomplished through three predefined routines within the "Interaction Handler". Two additional calls (directly from the applications program) deal with the network initialization and termination procedures, as specified for the "secretary" of the network. Following is a specification of these five routines.

3.2.1.1 Network

This routine is called directly by the IGPL applications program and is used to perform the hardware and software initialization of the network interface. This routine is called within the entry block of an IGPL program, before any interactions have taken place. In stand-alone operation this call is simply ignored. On generation of a network RESTART, this is the point to which the program will return to re-initialize the common visual space. In addition to initializing the network state variables, this routine must also initiate the LOGON procedures required to link the nodes wishing to participate in the current interaction dialogue.

```
I G P L  V2B

         *
         *
         *          SAMPLE.IGL
         *
         *          A SAMPLE IGPL PROGRAM
         *
         *


         REAL X,Y

         EXTERNAL NETWRK,NETLOG

         PAGE 319,239

         ENTRY
                    CALL NETWRK                **CALL TO INITIALIZE NETWORK
                    DISPLAY TARGET,STPBLK      **DISPLAY OBJECTS
                    MARKER;AT(120,220);FREE    **MARKER ON, FREE TO MOVE

         OBJECT TARGET
                    AREA 160,160;COLOR(BLUE);AT(40,40)
         ACTION
                    MARKER LOCATION(X,Y)       **GET THE MARKER LOCATION
                    INSERT TARGET
                    AREA 10,10;COLOR(YELLOW);AT(X,Y)
                    END INSERT

         OBJECT STPBLK
                    AREA 35,25;COLOR(RED);AT(260,210)
                    TEXT  'STOP';COLOR(WHITE);AT(265,215)
         ACTION
                    CALL NETLOG                ** CLOSE NETWORK COMMUNICATIONS
                    STOP

         END

END PASS 1
END PASS 2
```

Figure 12. Sample IGPL Program

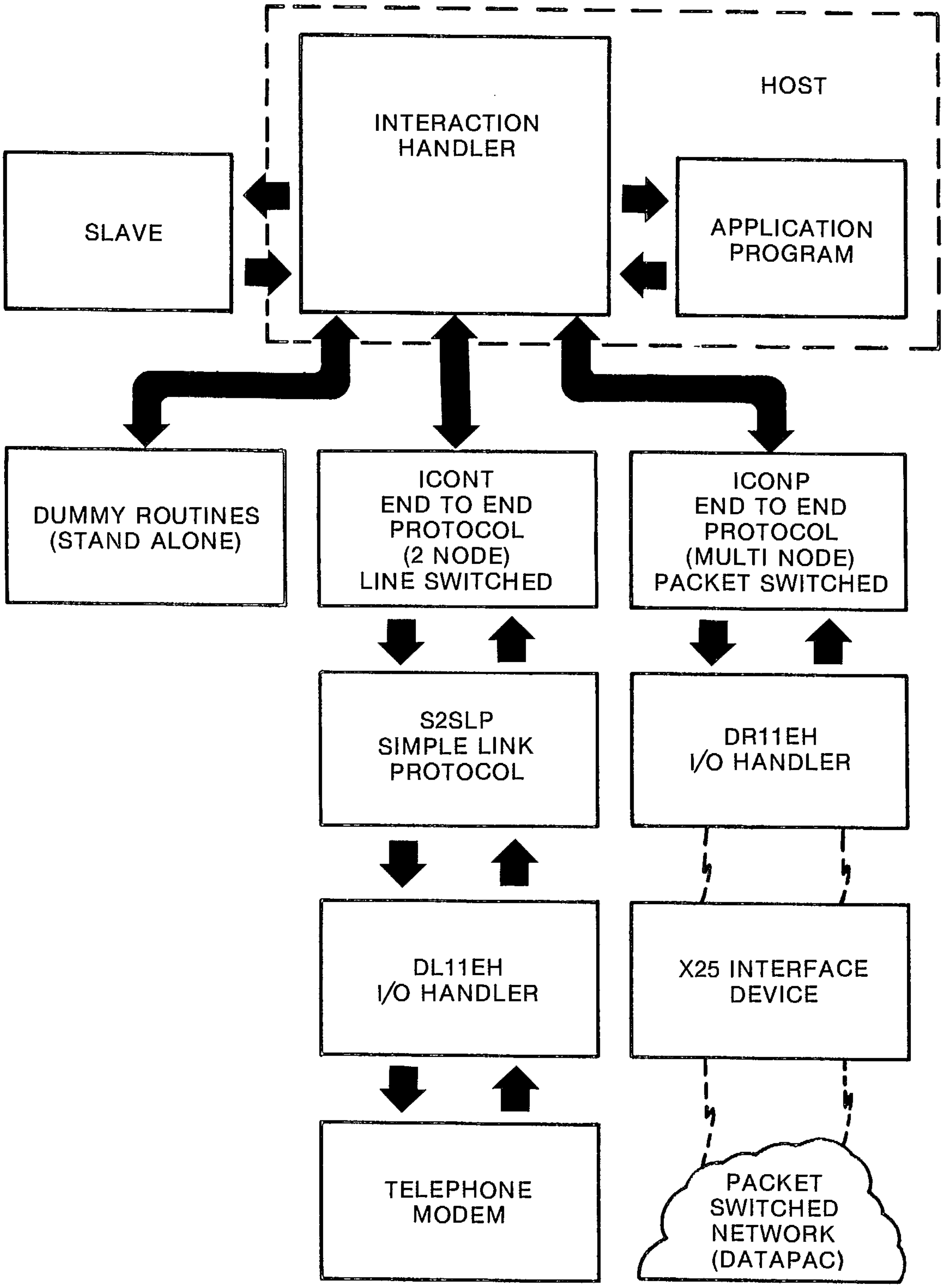


Figure 13. Software Modules for Interactive Image Communication System

The LOGON process will be done through the system console, rather than through the IGPL program itself. Part of the initialization phase includes the creation of the interaction table, used to record the graphical messages (GTIs) sent over the network. In addition to recording both POLLED and UNPOLLED GTIs, the current master ("chairman") at the time of the interaction, will also be recorded. The table will not however, record any of the network communication protocol commands.

After initialization and a successful ELECTION of a "chairman", control returns to the IGPL program, which may then commence execution.

This LOGON procedure will also establish a call DIRECTORY, of the form of Figure 14, which will be used by each node to determine the "identity" of the nodes currently in the network.

BYTE	FUNCTION
1	NUMBER OF CALLS CONNECTED (CALLST)
2	NUMBER OF HOME STATION (CALLID)
3	NUMBER OF NODE ON CHANNEL 1
4	NUMBER OF NODE ON CHANNEL 2
5	NUMBER OF NODE ON CHANNEL 3
6	NUMBER OF NODE ON CHANNEL 4

Figure 14. Functional Representation of the Call DIRECTORY

Since, in this initial implementation, this is a "one-time" action the "secretary" will perform both the LOGON and ELECTION phases of initialization. The states associated with network initialization are diagrammed in Figure 15.

3.2.1.2 NETLOG

This routine is again called directly by the IGPL program, as a result of a request to LOGOFF from the network. As this call is made within an ACTION block, it can only be made as a result of an interaction dialogue which results in the requesting node becoming the current master. Thus the basic protocol of acceptance must be followed to reach this state.

The function of this routine is to close out the interaction table, in its current state, initiate a LOGOFF procedure and then disconnect the requesting node from the network, in a manner acceptable to the protocol established by the communications network (DATAPAC).

Once completed, control would normally shift to the ELECTION state, to establish a new "chairman". However, in this implementation control will return to the IGPL program at which point the program will halt.

The general LOGOFF dialogue has been discussed in the section dealing with the chairmanship protocol, and the states associated with this procedure are illustrated in Figure 16.

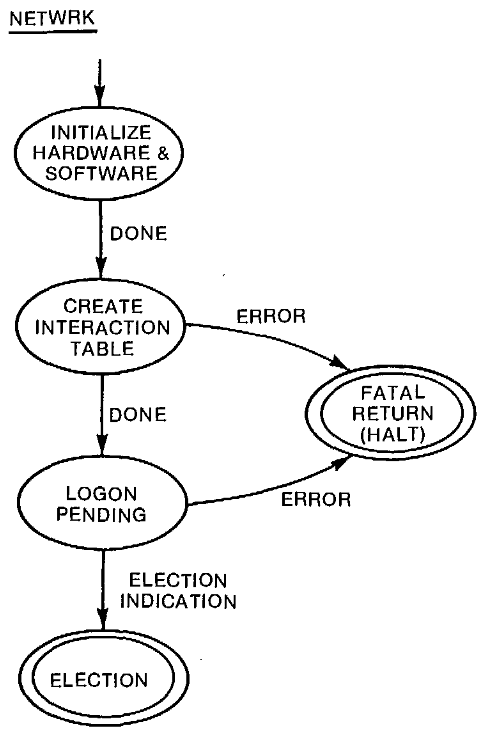


Figure 15. State Diagram of Network Initialization Procedures

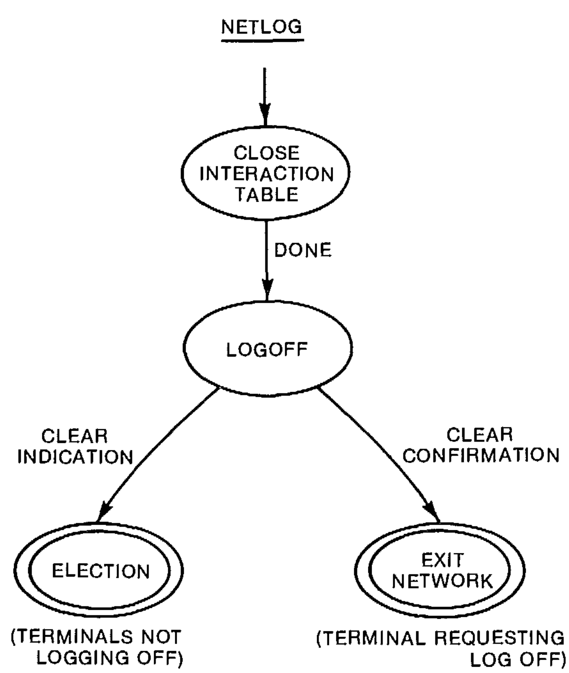


Figure 16. State Diagram of LOGOFF Procedures

3.2.1.3 S2TSTF

This call is imbedded within the IGPL language and is transparent to the application programmer. The call is made at critical points in the execution of the IGPL program, to allow for temporary suspension of the program while an interaction "interrupt" from the graphics slave is processed. Thus, the requirement of this routine is simply to poll the slave status flags, (both self and network), returning to the applications program with a yes/no condition set. The critical points at which this routine is called are generally before such action as ERASE and INSERT and the result of the call to S2TSTF is crucial to the sychronization of the running program. Depending on the state of the program at the time this polling is performed the slave interrupt may or may not be processed at this point in the program. If it is to be processed it is done through the next routine S2CHKB, otherwise it is left in a pending state until processing becomes appropriate. This process is diagrammed in Figure 17.

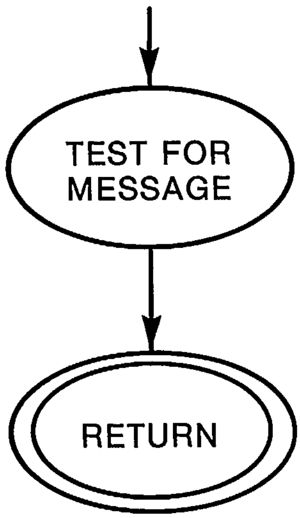


Figure 17. States Associated with S2TSTF Synchronization Routine

3.2.1.4 S2CHKB

This routine is called when there is a need to check whether or not the back GTI buffer should be updated. This may be as a result of the conditions after a call to S2TSTF or upon being awakened from a WAIT state. Only unpolled GTI's are dealt with in this routine, consequently only the tag number associated with the particular object which initiated the interaction interrupt is communicated. Since all of the terminals connected to the network are running the same application program this is the only information which need be communicated (the number will identify the same object for all users). The basic function of this routine is to acquire this back GTI information and return to the applications program with an EVENT flag set, indicating that there is a new interaction to process. If there is new information from the network the goal is to acquire this message. Resolution of the master/participant status will be taken care of by the SEND and GET routines which provide the END to END protocol required by the IGPL program.

There are three basic return states from the SEND and GET routines. Upon a successful return the interaction table is updated, the new information copied to the return GTI buffer and a return to the IGPL "Interaction Handler" is executed, with the EVENT flag set. If the sending terminal was not the current master, a return from the SEND state is made indicating a request to become master is pending. The program then transfers to the GET state to await the outcome of the pending request. A return indicating a SEND failure can safely be ignored by returning to the IGPL program without the EVENT flag set. It will then be up to the user to try again to complete the interaction which was attempted. If the terminal calling this routine is the current master, a return from the GET routine may take place with an indication that a request to be master has been received and is eligible for acceptance. The program then transfers to the SEND state with the request pending. The only other allowable return from the GET state would be an error in GET with a request to be master pending. This may again be ignored by returning to the IGPL "Interactions" are not currently enabled when the program enters the S2CHKB state the entire routine is ignored and an immediate return is executed (no EVENT). The states associated with this update procedure are shown in Figure 18. (SEND and GET are implemented as SENGTI and GETGTI).

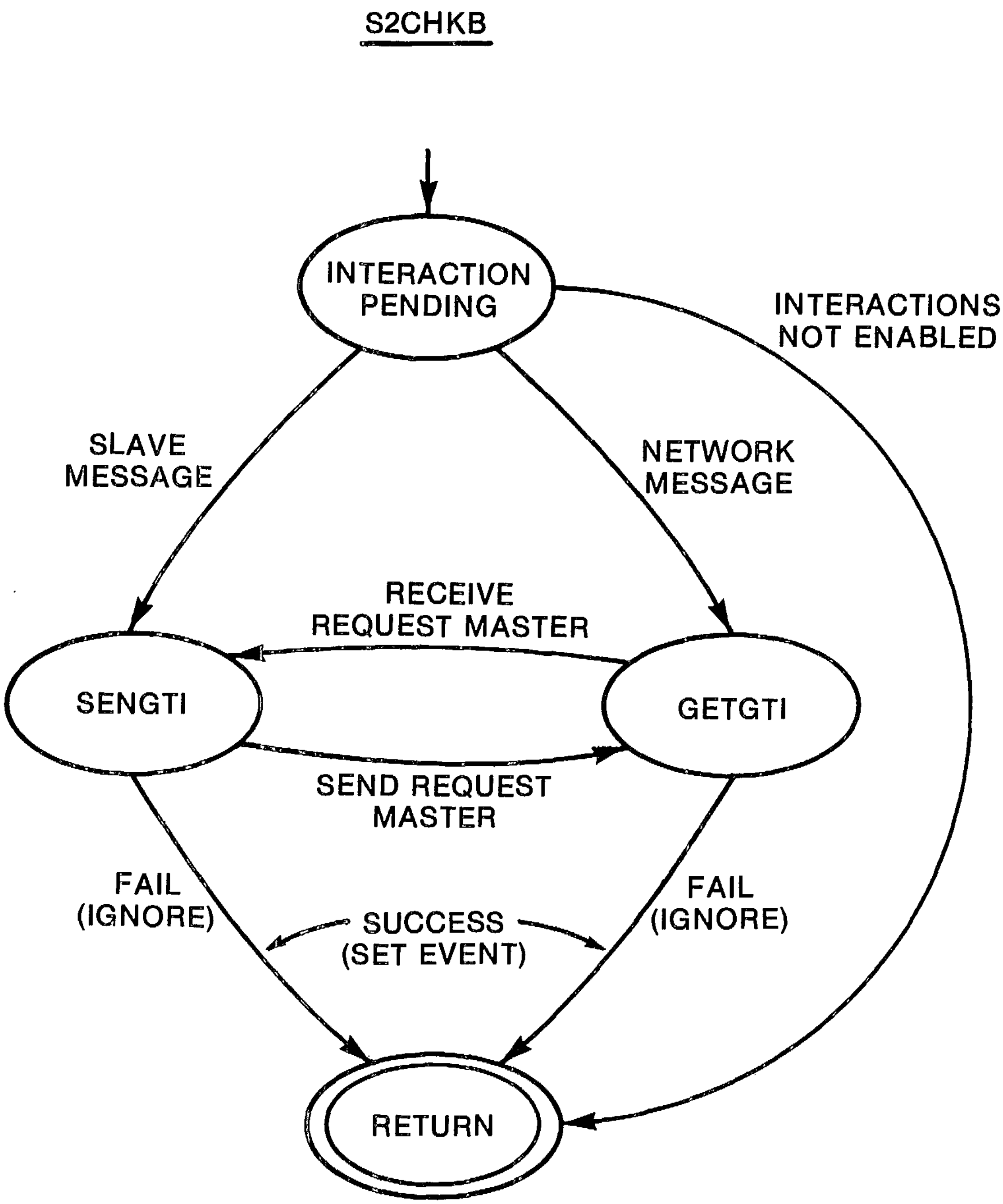


Figure 18. State Diagram Associated with GTI Buffer Update Procedures

3.2.1.5 S2PGTI

This routine is called by the "Interaction Handler" during processing of a forward GTI which requires input from one of the slave interaction devices, such as the co-ordinates of the current MARKER LOCATION. The slave will have "polled" the appropriate device for input and then wait for this information to be requested by the host. Because all of the slaves in the network will be responding to the same forward GTI they will all have information to be processed. The function of this routine is to acquire this information and return to the "Interaction Handler" to continue processing. Since this state occurs during the execution of an interaction which has already been partially processed, failure to acquire this information will be fatal to the overall program execution. Thus NO EVENT flag is required since the only return allowable is with the information already expected by the "Interaction Handler". Although each terminal will have it's own information to process, only current master's information is valid. Consequently only the master will attempt to SEND, all other terminals will go GET the network information. Failure in either case must initiate a recovery process to maintain system synchronization. It is necessary to first acquire one's own polled GTI first in order to maintain synchronization between the host and the slave. This information may safely be ignored by all but the master, however, failure to acquire a valid response must also initiate recovery procedures, since host/slave synchronization may be lost. These procedures are illustrated in Figure 19.

3.2.2 Chairmanship Implementation

The chairmanship protocol is essentially to be accomplished through the routines SENGTI and GETGTI.

SENGTI is required only to send the GTI to the network (one or two packets) and return to the calling routine (S2CHKB or S2PGTI), or send a request to be master, along with the interaction or control message. This decision will be based on the status of CHAIRSTATE.

GETGTI must be able to recognize the interaction data or control messages and depending on the status of RUNSTATE execute a return or a "jump" to the appropriate non-NORMAL RUNSTATE. In addition this routine must also be able to recognize the network message for an incoming call such that the appropriate LOGON procedures may be initiated.

These functions are outlined in the following state diagrams, (Figures 20 and 21).

3.2.3 X.25 Interface Protocol

In the NORMAL RUNSTATE the X.25 interface protocol is quite simple due to the synchronous nature of the flow of interaction messages. The SENDMESSAGE routine has only to transfer the contents of the send buffer to the network, wait until the send is acknowledged by the interface device (i.e., a FLOWGO), indicate that the channel is to remain open (send a FLOWGO) and return to the calling routine with the appropriate condition set (SUCCESS,FAILURE).

The GETMESSAGE routine is only slightly more complicated. Because some of the messages may be two packets in length this must be recognized by the interface protocol in order that the complete GTI is obtained. Thus there must be a pending state where only a continuation of a previous message is recognized. Besides FLOWGO the only other opcode which is recognized as valid in the NORMAL RUNSTATE is an INCOMING CALL.

The FLOWGO will acknowledge all received messages, including the one not recognized in the NORMAL state. These messages will simply not be copied to the network receive buffer. (SENDMESSAGE and GETMESSAGE are implemented as SMESS and RMESS respectively and appear in Figures 22 and 23).

3.3 SOFTWARE STRUCTURE

The following is a pictorial representation of the structure of the software developed for the initial implementation of the image communication system described in this report. An attempt has been made (Figure 24) to relate the various procedures in the implementation to the levels of protocol originally outlined.

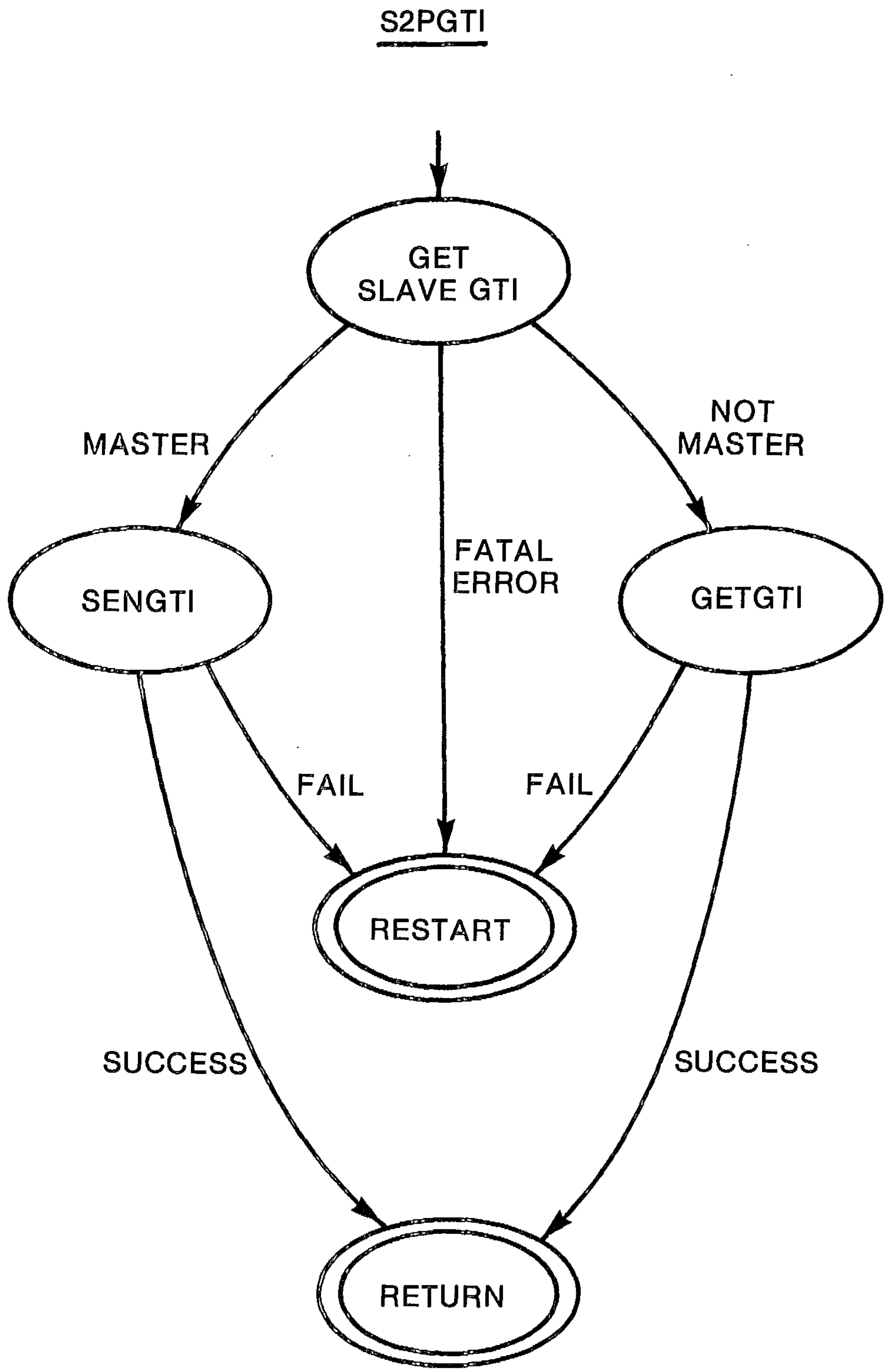


Figure 19. State Diagram of Polled GTI Data Acquisition

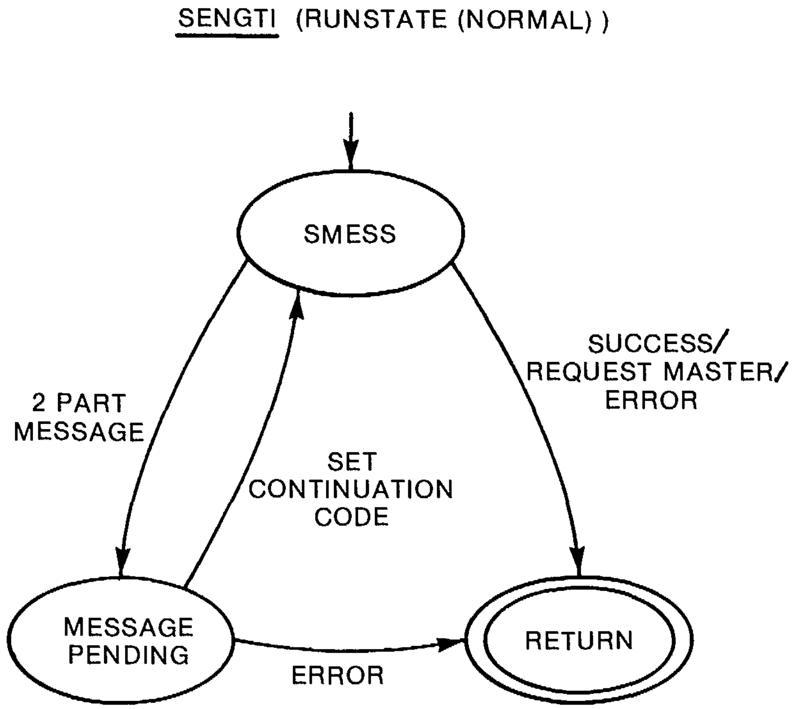


Figure 20. State Diagram of Procedure Associated with Transmission of a GTI

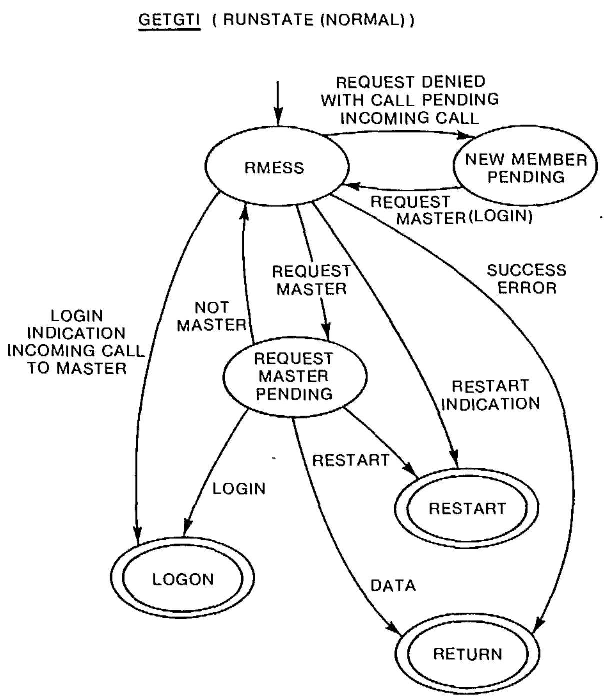


Figure 21. State Diagram Associated with Receipt of a GTI

SMESS (RUNSTATE (NORMAL))

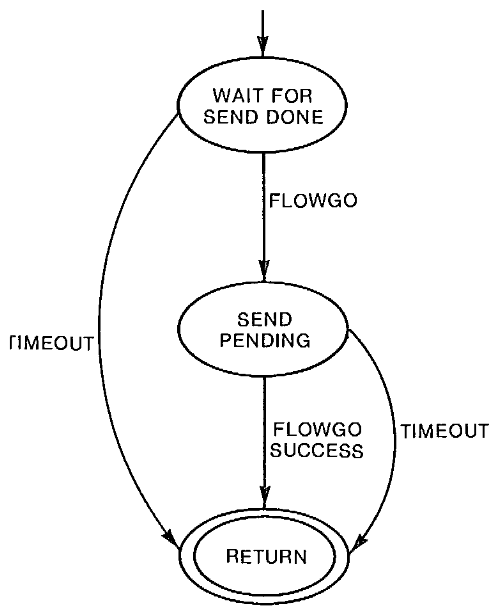


Figure 22. State Diagram Associated with Transmission of a MESSAGE

RMESS (RUNSTATE (NORMAL))

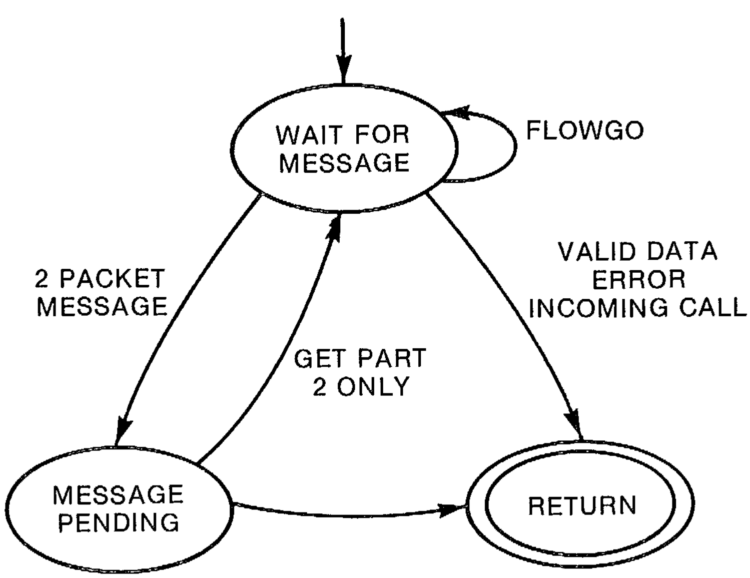


Figure 23. State Diagram Associated with Receipt of a MESSAGE

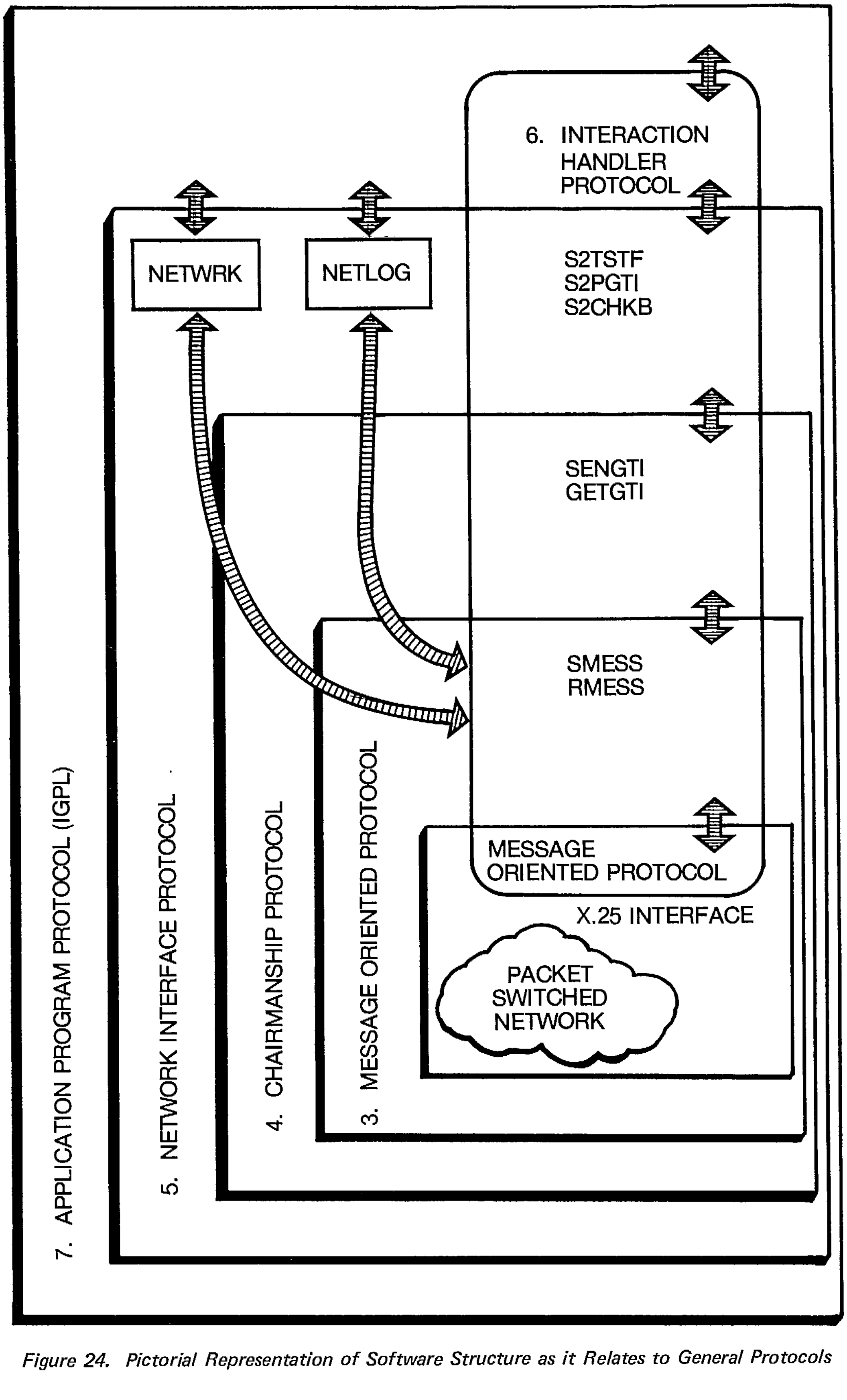


Figure 24. Pictorial Representation of Software Structure as it Relates to General Protocols

4. SUMMARY

The foregoing report has attempted to provide general design specifications of an interactive image communication system. Additional requirements of the initial implementation of such a system by the Department of Communications have also been defined to illustrate some of the considerations not a part of the generalized protocol of operation (chairmanship protocol).

Testing of the newly designed X.25 interface device and of the network as a whole is currently underway at the Communications Research Centre in Ottawa. Based on the results of these tests the specific protocols of operation for the non NORMAL state of operation are currently being defined and it is hoped that a fully operational multi-node Interactive Image Communication System using Packet Switched Networks will be demonstrated in the near future.

5. REFERENCES

1. Computer Communications Group, *Datapac, Standard Network Access Protocol Specifications*, 1976.

2. Sawchuk, W., H.G. Bown, C.D. O'Brien and G.W. Thorgeirson, *An Interactive Image Communication System Using Narrowband Lines*, Computers and Graphics, **Vol. 3**, No. 4, 1978, pp. 129–134.

3. Bown, H.G., C.D. O'Brien, R.E. Warburton and G.W. Thorgeirson, *System Independence for Interactive Computer Graphics Applications Programs*, Proc 4th Man-Computer Communications Conference, Ottawa, 1975.

4. O'Brien, C.D., *Image – A Language for Interactive Manipulation of a Graphic Environment*, M. Engng Thesis, Carleton University, Ottawa, 1975.

5. LeLann, G., *Distributed Systems – Towards a Formal Approach*, Proc. IFIP, Toronto, 1977, pp. 155–160.

6. *IGPL – Interactive Graphics Programming Language: User's Reference Manual*, Norpak Ltd., Pakenham, Ontario, September 1977.

7. Finn, S.G., *Resynch Network Protocols*, Proc. ICC, 1977, pp. 30.1–254 – 30.1–258.

8. Schwartz, M., *Computer-Communication Network Design and Analysis*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1977.

CRC DOCUMENT CONTROL DATA

1. ORIGINATOR: Department of Communications/Communications Research Centre

2. DOCUMENT NO: CRC Technical Note 701

3. DOCUMENT DATE: December 1980

4. DOCUMENT TITLE: An Interactive Image Communications System Using Packet Switched Networks

5. AUTHOR(s): W.T. Lalonde and C.D. O'Brien

6. KEYWORDS:
(1) Image
(2) Packet
(3) Networks

7. SUBJECT CATEGORY (FIELD & GROUP: COSATI)

17 Navigation, Communications, Detection, and Countermeasures

17 02 Communications

8. ABSTRACT:

The role of computer-to-computer communication has become essential in today's society. Establishment of two-way image communication, between remote locations, has until recently suffered from the high bandwidth requirement associated with visual data.

Over the past several years, scientists at the Communications Research Centre have been developing a bandwidth reduction coding scheme for the transmission of image data over various communication media. A two node network for interactive image communication, operating over 300 baud switched telephone lines, has been demonstrated at CRC in Ottawa. The system is based on the concept of a common visual space, wherein an identical image is presented to both users. The image may be modified by either participant, the results of which will be seen by both.

This report outlines a proposed protocol of operation for a multi-node image communication system, using a packet switched network (DATAPAC) as the medium of inter-node communication. The objective of this protocol is to maintain synchronization between all of the participating nodes, such that the concept of a common visual space is extended to multiple users.

9. CITATION:

Government of Canada Gouvernement du Canada