International Institute for
Applied Systems Analysis
IIASA www.iiasa.ac.at

Networking Communication Operating System (NCOS)

Radziminski, A.

IIASA Professional Paper
August 1976

PP-76-7

NETWORKING COMMUNICATION OPERATING SYSTEM (NCOS)

(Established in the Nodes of Experimental Data Communication Network)

A. Radziminski

August 1976

ABSTRACT

This paper presents an overall view of the project goals and present facilities of the Experimental Data Communication Network established in Warsaw, Gdansk, and Katowice, Poland, by the Institute of Communication in Warsaw, within the framework of its research program during the period 1974 to 1976.

The experiment consists of linking different host computers, via nodal computers, using the packet switching technique. It is proposed to connect the Warsaw node of the above network to the IIASA Network, enabling several subscribers selected in Poland to link to the IIASA Network.

This paper is dedicated to the participants of the IIASA Network activity from NMOs and other national institutions, as well as to those from IIASA.

I. INTRODUCTION

This paper presents an overall view of the project goals and present facilities of the experimental data communication network. This network was designed by the research group from the Institute of Communication in Warsaw, and established in Warsaw, Gdansk and Katowice, in Poland.

It is proposed to connect the Warsaw node to the IIASA Network, which will enable several subscribers selected in Poland to link to the IIASA Network (for example, the Institute of Control and Management Sciences). The operating system in Warsaw is an existing one, and there will be some problems concerning internetwork and interfacing (see Appendix). The interface problem is shown in the paper "Experimental Connection Between the PDP 11 Installation at IIASA and the Message Switching Node at Warsaw, Based on the Singer 10 Computer System," CSN 020 Computer Science Project paper, IIASA July, 1976.

II. NETWORKING COMMUNICATION OPERATING SYSTEM (NCOS) ESTABLISHED IN WARSAW NODE OF THE EXPERIMENTAL DATA COMMUNICATION NETWORK

The goals of the project are primarily experimental. The prime objective is to demonstrate the feasibility of developing a data communication network capable of simultaneously serving a variety of organizations using a range of different terminals and computers. *Secondary objectives are to gain first hand knowledge and experience in designing and developing computer controlled data communications systems, and to experiment with different data transmission techniques.*

The primary goal will be met in designing and implementing a packet switched communication network. The basic packet switching system is functioning between the three nodes. In designing the NCOS (Networking Communication Operating System) the theoretical final grid network will be kept in mind. For each node a maximum of five neighboring nodes and a total of 20 network nodes is assumed.

A. Basic Facilities

The NCOS package is designed and used to support the following functions:

1) Inter-nodal simultaneous bi-directional communications between System Ten nodes, using addressed data packets;

2) Communication with IBM computers using BSC procedure in point-to-point leased line or dial connections;

3) Communication with ICL 1900 series computers using ICL 7020 procedures in point-to-point leased line or dial connections;

4) Communication with terminals and mini-computers in point-to-point leased line or dial connections and in multi-point leased line connections;

5) Dynamic buffering of data packets via a linked buffer pool in common memory;

6) The control module provides for overall system control, and in particular provides for the multi-path routing of data between users;

7) Dynamic storage of data on System Ten discs via the use of paged linked sequential accessing method. Using this accessing method any number of files of indeterminable length, all sharing a large disc area, can be dynamically created and accessed for read and write operations. Operated with EDIT and MAINT expended and deleted.

The user can select any or all of the above application modules to either communicate with other computer types, or to perform special processing functions.

Given below is some further relevant information for the IIASA Network project:

a) Functions of the nodal computers;

b) Data packets format (Figure 1);

c) User data flow (Figure 2);

d) Time division of data packets (Figure 3);

e) Routing policy for the NCOS system;

f) Transmission error control;

g) Testing facility;

h) Tracing facility;

The Nodal Computer performs the following functions:

1) Transmission of data to and from the various regional end users connected to that node and providing the entry point for those users to the network;

2) Transmissions of data over the network to other nodes;

3) Data buffering;

4) Control of transmission over all lines connected to that node;

5) Routing of all data packets originating at the node and passing through the node;

6) Interface to the nodal computer operator;

7) Allocation of resources at the node;

8) Handling of task priorities;

9) Informing neighboring nodes of change in operational configuration and nodal data volume;

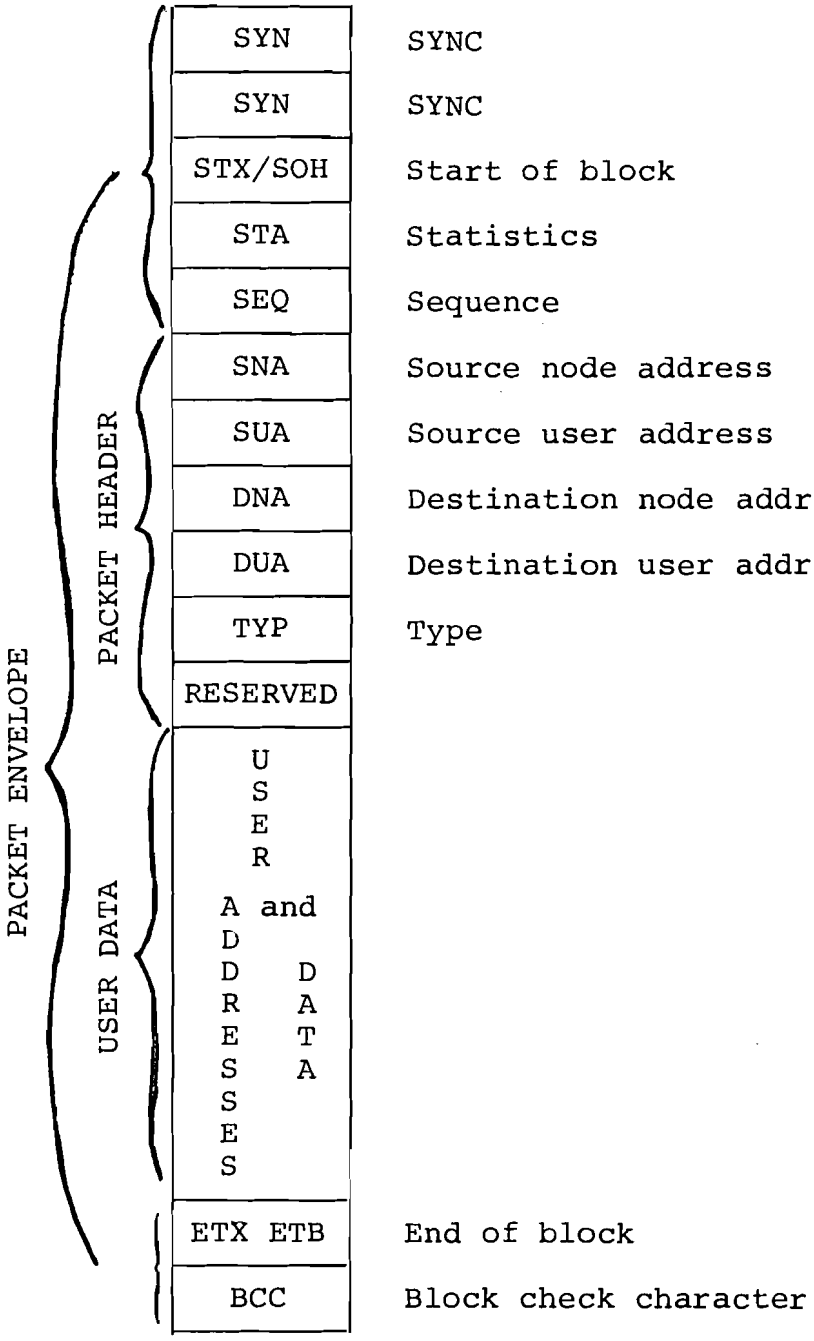
10) Transmission by line testing of segments connected to the node;

11) Display of system and network status for operator information;

12) Creation of performance statistics for the node;

13) Creation of accounting information for the regional users connected to the node.

Figure 1. Packet Format



SYN — The Sync character required for SCA to obtain character sync with arriving buffers.

STA — Statistics data for this packet. Bits 1-4 are a consecutive count of the number of nodes the packet passes through. When lines or nodes are not working packets will be re-routed automatically. If the nodal count in STA reaches 8 then it is assumed that there is no working route to the destination and the packet is discarded.

Bit 5 is turned on if a parity error occurred any time during transmission.

Bit 6 is turned on if a time-out occurred any time during transmission. In either case a retransmission would have occurred (the count of nodes passed is not increased during retransmissions). This information is used for statistical purposes in determining end-to-end line quality.

STX or SOH — Start of text or start of header character. Starts blocks check character accumulation and used in conjunction with ETX/ETB to serve as ACKØ, ACK1, and wait-a-bit control signals:

STX, ETB } ACKØ, SOH, ETB } ACK1

STX/SOH, ETX } WAIT

SEQ — Packet sequence. The user message is broken into shorter length packets for transmission over the packet switched networks. Since packets can take different routes due to congestion or line failure a packet sequence number is needed in order to reassemble the packets in the correct order. Bits 1-4 give the sequence number from 0 to 9. Sequence numbers alternate between a string of positive numbers (0 to 9) and a string of negative numbers (- 9 to - 0).

RESERVED — Reserved for test purposes.

ETX or ETB — End of transmission block and last character block check accumulation.

TYP — Type of packets

Type 0 : Control diagnostic

1 : Request

2 : Positive acknowledgement

3 : Negative acknowledgement including cause:
- number busy, - access barred, - invalid call,
- out of order

4 : Positive acknowledgement of 6 packets

$\overline{4}$: Negative acknowledgement of 6 packets

5 : End of transmission

6 : Data

7 : Test

User Data Packets Priorities

User data packets are assigned priorities according to the user application and priority rating as given in the user accounting file. Arranged with the highest priority first, examples of the available user data packet priorities are:

1) Conversational/inquiry high priority;

2) Conversational/inquiry normal priority;

3) Remote job entry;

4) Bulk file transmission high priority;

5) Bulk file transmission normal priority;

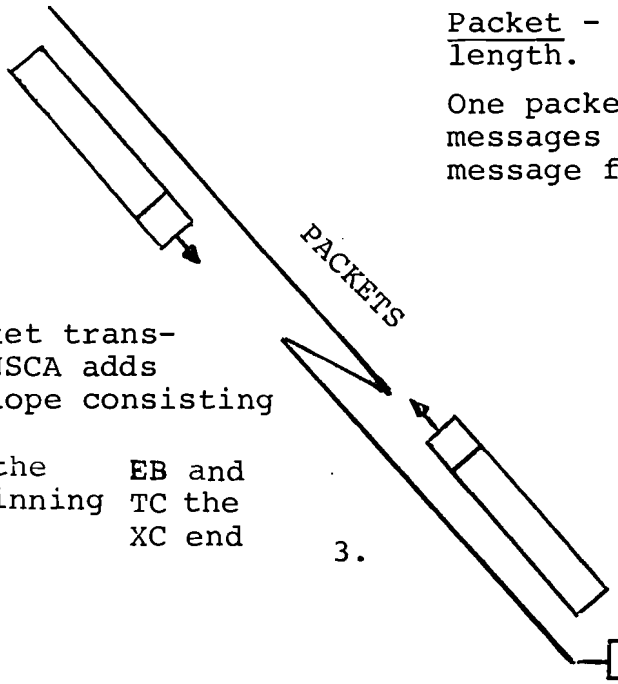
6) Bulk file transmission night priority.

Data packets can vary in length

Message - a logical block form or to an end user.

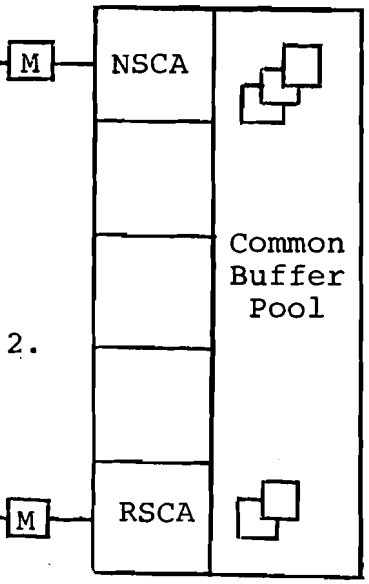
Packet - a block of fixed max. length.

One packet per message for short messages and several packets per message for long messages.



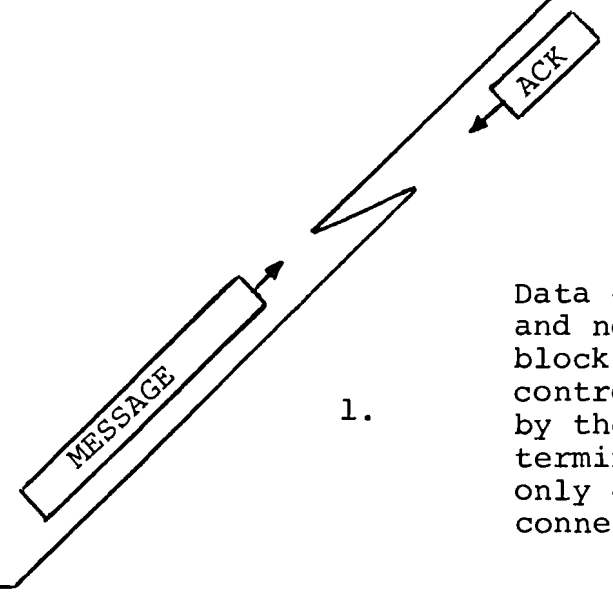
This packet transmission NSCA adds the envelope consisting of the SSSS at the YYTT beginning NNXA and EB and TC the XC end

3.



The interface RSCE breaks the message block into packets of up to 200 user characters each. If a user message is less than 200 characters it goes into one short packet. This RSCA adds the packet header characters and passes the buffers containing the packets to the packet transmitting SCA.

2.



RSCA - Regional SCA

NSCA - Network SCA

SCA - Synchronous Communication Adaptor

1.

Data communications between user and node uses whatever message block size and transmission control procedure that is required by the end user computer or terminal. The end user supplies only data except for an initial connect message.

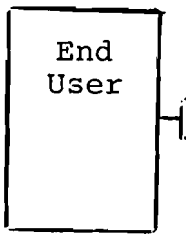


Figure 2. Flow of Data Storing Packet Switching

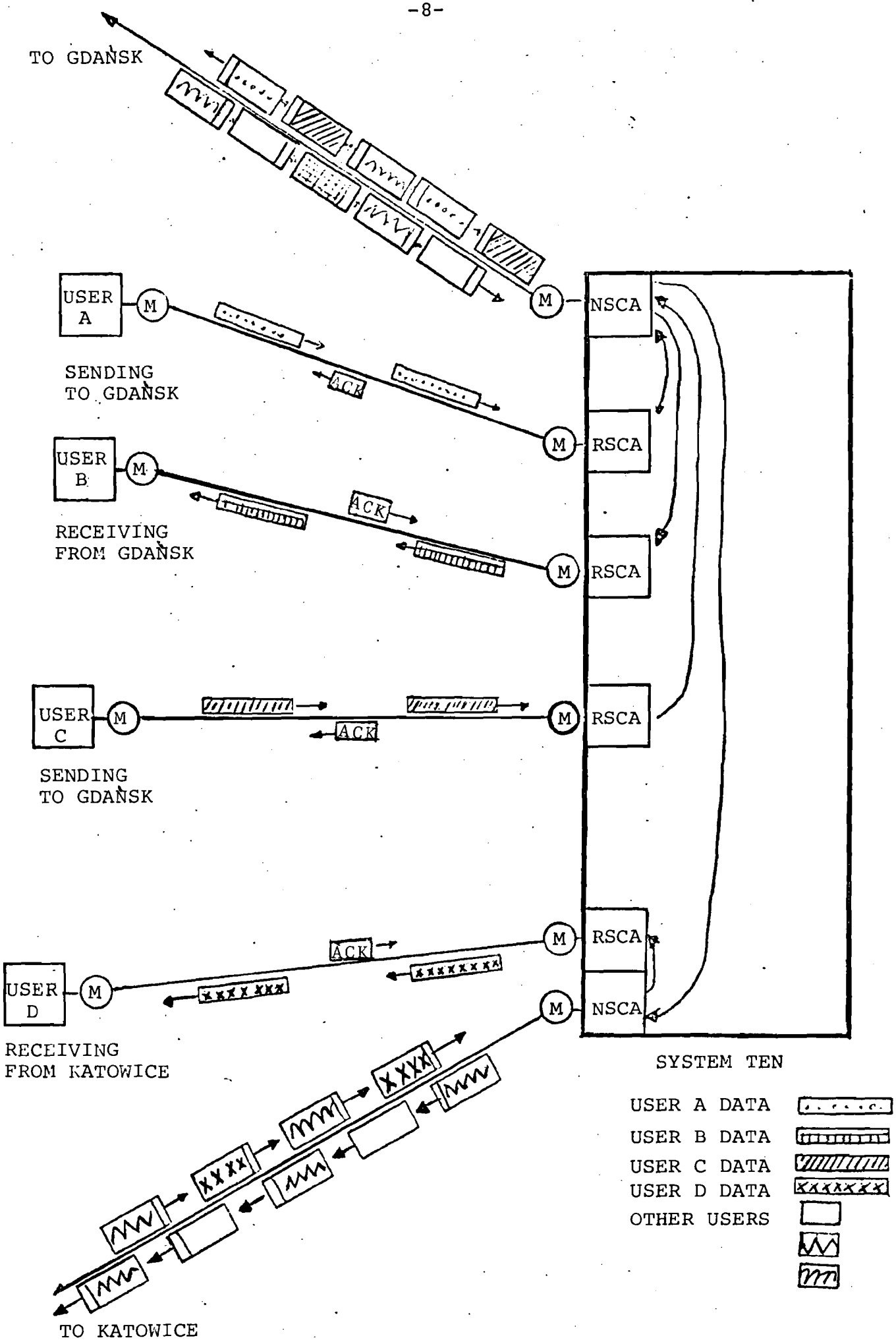


Figure 3. Time Division Multiplexing of Data Packets

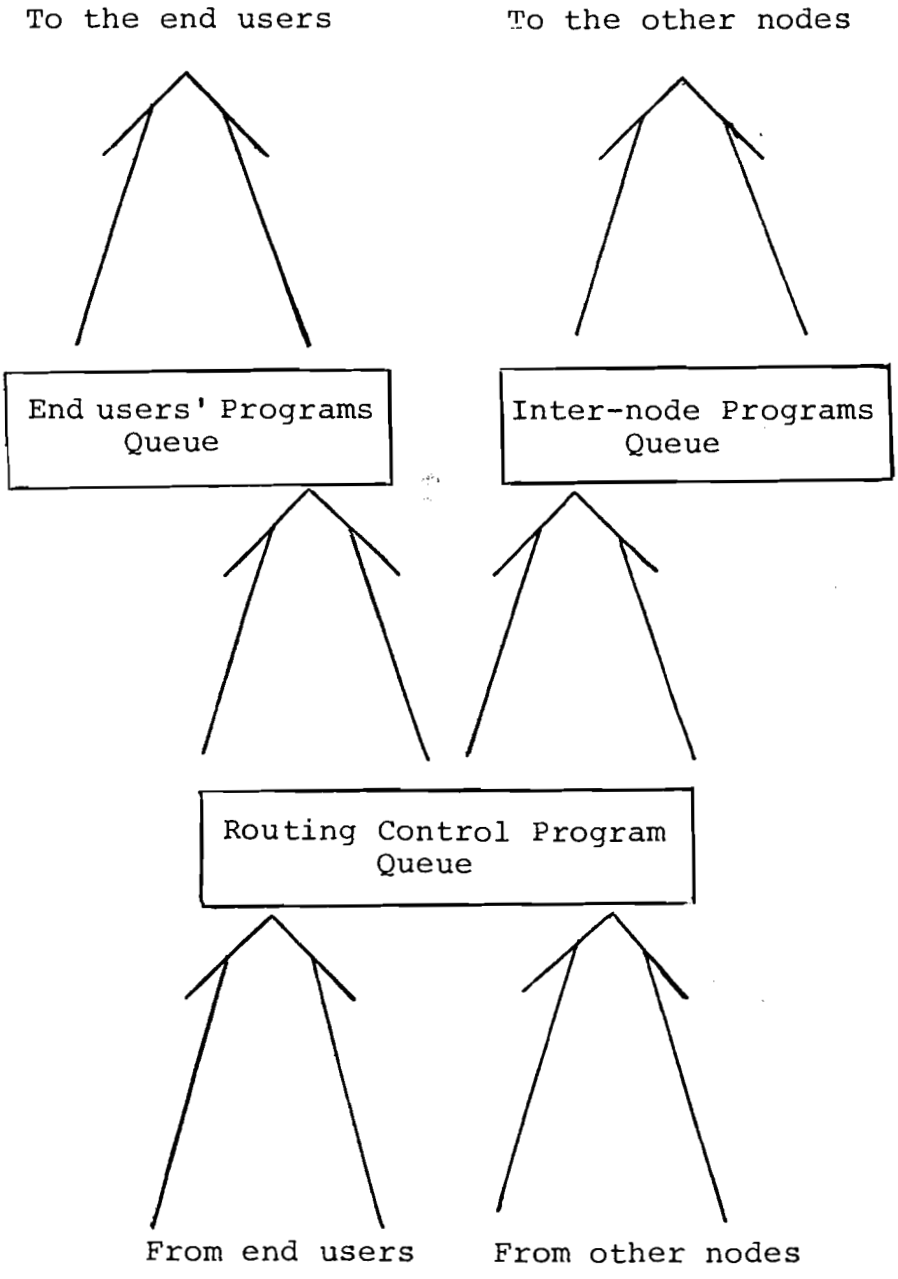


Figure 4. The Overall Idea of Packet Protocol Flow

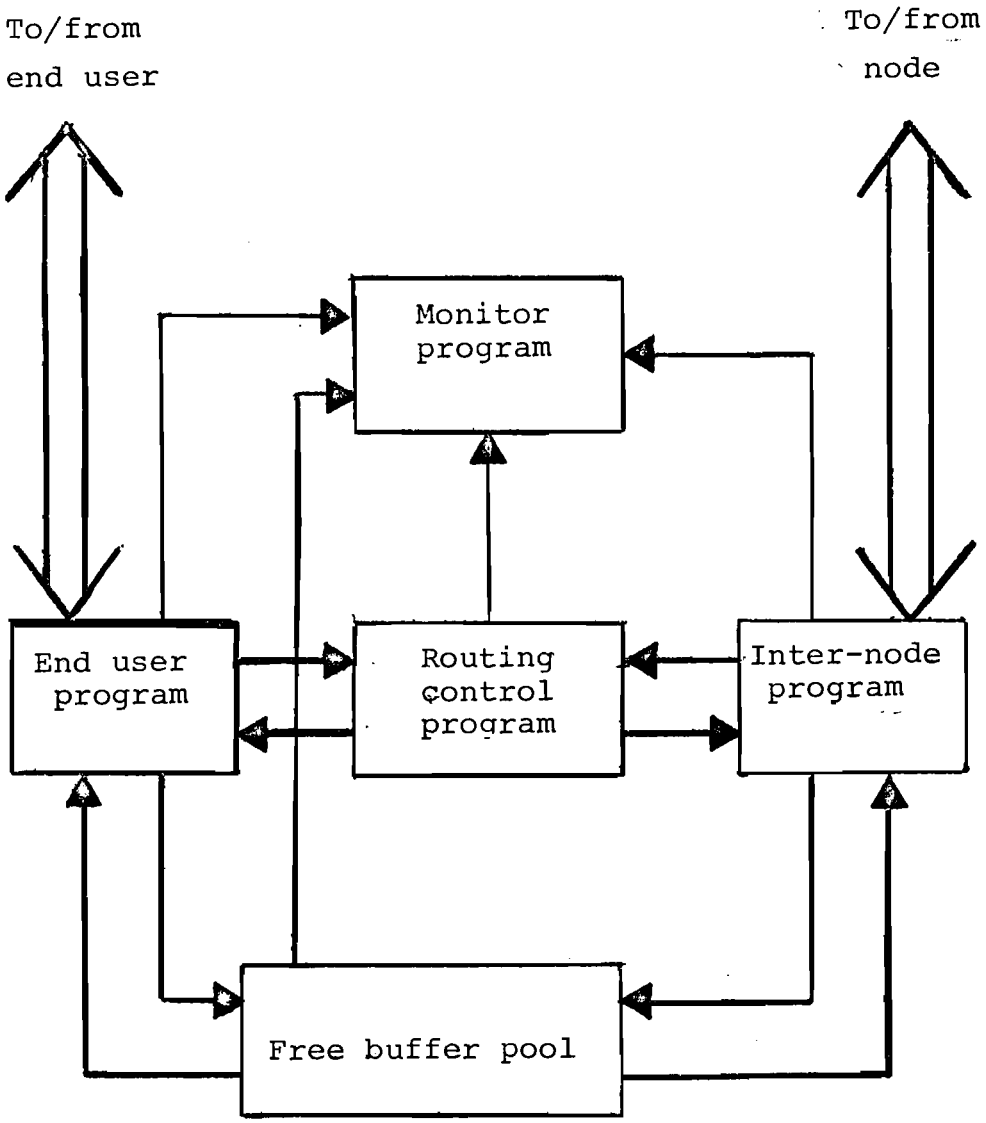


Figure 5. Functional Scheme of Node

B. Routing Policy for the NCOS System

1) Initially a fixed routing policy is used. As the network becomes operational and experience is gained the fixed routing policy is replaced by an adaptive policy capable of modifying route priorities as a function of the system equipment failures, traffic volumes, and time of day.

2) Routes of transfer (strictly the first route segments) between given and all remaining nodes of the network ordered to the fixed priorities created by the schedule of the routing.

3) Priorities of the routes are established according to the rule of minimal number and intermediate nodes. The chosen route may be rejected and another will have to be investigated when the number of buffers exceeds the defined value or line quality.

   The factor defined by failures and time-outs has one of the following values:

   - the line is correct
   - the line is unreliable
   - the line is out of order.

4) The routing table flags provide the possibility of indicating three conditions:

   - the operator has disabled the route;
   - the system has automatically temporarily disabled the route because of transmission difficulties;
   - the route is temporarily overloaded.

5) Each node is able to transmit to its neighbor any changes in the status of its lines by the means of the status characters in each data packet envelope.

6) On receiving a change of line status characters from a neighboring node, a given node updates its routing tables.

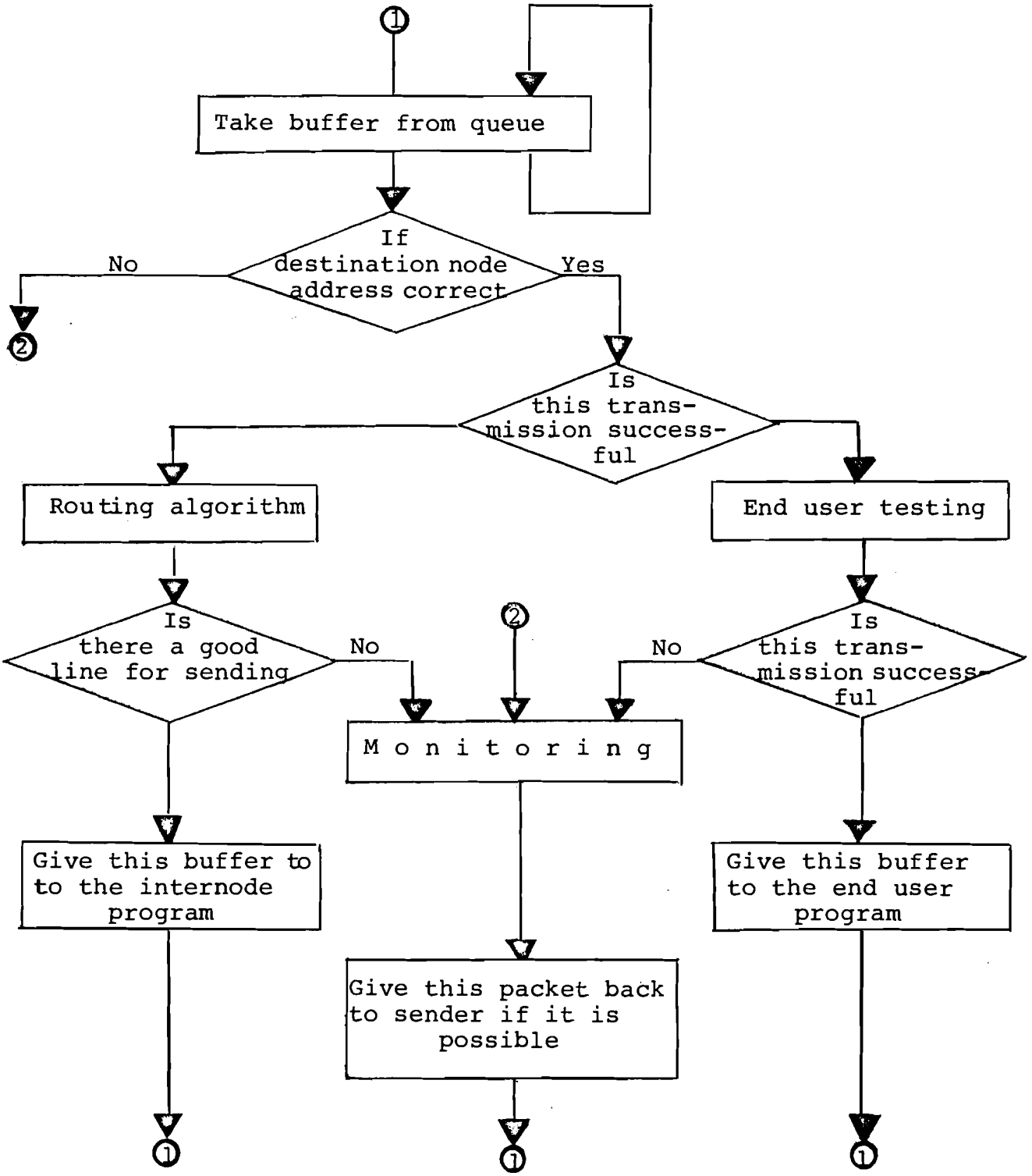


Figure 6. Overall Flow of the Routing Control Program

7) Packet routing is done over the highest priority line that is not overloaded provided that the following node on this route also has an operational route that is not overloaded for passing the data packet on to a third node.

8) Data packets are never routed back over the same line that they arrived on.

9) If a data packet has passed through more than nine nodes it is discarded. This is because no route in the system passes over more than three intermediate nodes and if a data packet has passed through nine nodes then the various routes to the destination are probably not working. There is no need to signal the loss of the data packet since by means of the end-to-end error control procedures the destination node will automatically note that its sequence number is missing and will ask the source node to re-send the packet.

10) If there is no available operational route over which to forward the data packet it is discarded.

11) No bulk data storage is provided at intermediate nodes for data packets in transit. Each node provides core memory buffer pools capable of storing several data packets per line. When the buffer pool is full further input packets are refused.

12) If any bulk storage of data is required as a service to users, it is only provided at the user entry or exit nodes of data starting or ending at those nodes.

13) Data packets of different priority queued up for each line are sent according to priority regardless of the sequence in which they arrived at the node. Within a given priority group data packets are sent in the same sequence in which they arrived.

c. Transmission Error Control

Transmission error control in a grid network is not as straightforward as it is in more traditional networks because of the multiple nodes data packets can pass through. In grid networks two levels of transmission error control must be incorporated: an internode control and an end to end control.

The internode control in the NCOS system works on a bi-directional principle in which data packets are sent alternatively in both directions over the single four-wire line between nodes. Each data packet also serves as the acknowledgement for the data packet just received in the opposite direction, thus eliminating the overhead of separate ACKS and NAK while retaining the security of a confirmation of good reception after each data packet is sent. If a data packet is not received correctly the next data packet sent would contain a flag bit asking for a repeat of the last packet. If the packet were received correctly the flag bit would be off giving the go-ahead to send the next packet. In checking for correct reception of data packets the transmission procedure used by NCOS is such that every transmitted character is controlled by both vertical and horizontal parity checks (except for sync and STX characters). Between neighbouring nodes data packets are retained in core memory and only erased when they have been successfully sent to, and acknowledged by the next node. In addition each data packet sent includes an internode sequence count to assure that no complete data packet exchanges are lost. In the NCOS system if there is no data to send in one direction between neighbouring nodes then only empty envelopes containing the ACK/NAK and sequence bits are sent in that direction in reply to each data packet received. In addition under NCOS, data packets are allowed to vary in size. Thus over the single line between neighbouring nodes the data volume can dynamically and instantaneously vary from all in one direction to all in the other direction, and including all ratios in between. In all cases complete and reliable error control is assured between neighbouring nodes since all data is controlled and repeated until it is received error free.

End-to-end error control works on the same principles as internode error control but is more complicated due to the fact that end-to-end control usually encompasses several nodes and requires longer retention of data packets and strict sequencing of data packets. The need for end-to-end error control arises from the fact that even with the neighbouring node control described above it is still possible for data packets to be lost en route. Such would be the case, for example, if a nodal computer suffers a power failure while it has several packets in memory and being transmitted.

End-to-end error control can be carried to various levels of sophistication. The simplest way is to send one data packet from a given user and then wait for the packet to travel through the grid to the destination node and for the acknowledgement from the destination node to arrive back at the source node before sending the next packet from the same user. This method also avoids the problem of data packets getting out of sequence and of duplicate packets in the network. The major disadvantage is that it greatly reduces throughput for the user since there is a significant delay between successive data packets due to the time it takes a packet to travel through the network (it is delayed at least one packet time at each intermediate node) and for the acknowledgement to return through the grid.

A better method is to send a limited number of packets, about five being a typical number, before waiting for the destination node to acknowledge. This is a much better method as it decreases the number of end-to-end acknowledgement messages by a (internode control still acknowledges each packet) factor of five and the matter of packets getting out of sequence is minimal since only five packets need be sorted at a time at the receiving node. In addition the matter of duplicate packets is also greatly simplified. This method is best if the majority of communications is of a conversational or inquiry/response type in which most complete messages can be sent within the five packet group. For sending data files this method is

not optimal since after every five data packets a considerable delay occurs while the last of the five packets travels through the grid and the acknowledgement giving a go-ahead to send the next group of five packets arrives back at the source node.

A more efficient--but also more complicated--method is to develop a pipeline effect in which like the method above the receiving node would only send one end-to-end acknowledgement for every five data packets received but in which the sending node does not wait to receive the acknowledgement but instead carries on with sending the next five data packets. In this method there would therefore be a continuous stream of data packets going in one direction and a stream of less frequent end to end acknowledgements coming back at the same time in the opposite direction. Each acknowledgement would indicate the latest packet sequence number received correctly at the destination node and request the repetition of any missing or erroneous packets having a sequence number less than the latest received. On receiving this acknowledgement the sending node would insert the packet that had to be repeated into the stream of packets being sent and release the other packets covered by the acknowledgement. Obviously the series of sequence numbers used would have to be long enough to avoid any confusion. A sequence of module 20 would normally be sufficient if the receiving node sends an acknowledgement after every five packets and if the sending node stops sending if 15 packets have been sent without receiving an acknowledgement. With this method there is of course the problem at the receiving end of putting the packets in the correct sequence and the problem of duplicate packets but these problems can be handled without too many more complications than are encountered in the previous method described above.

This last method of a continuous pipeline stream of packets is the method used in NCOS.

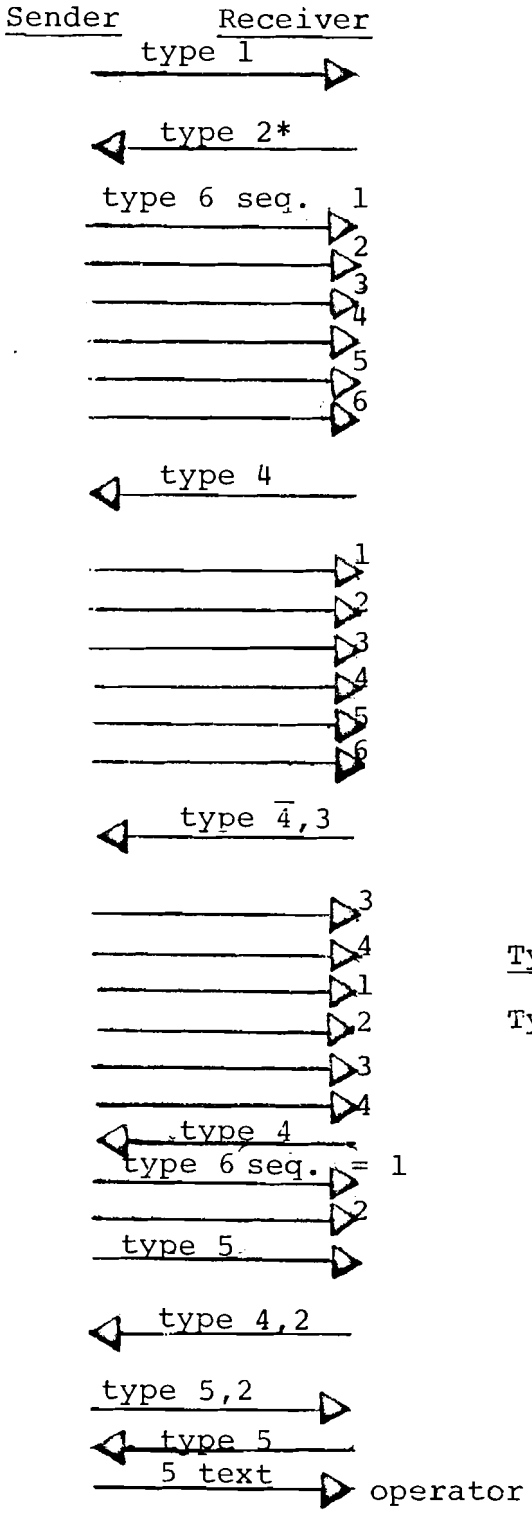
D. Testing Facilities

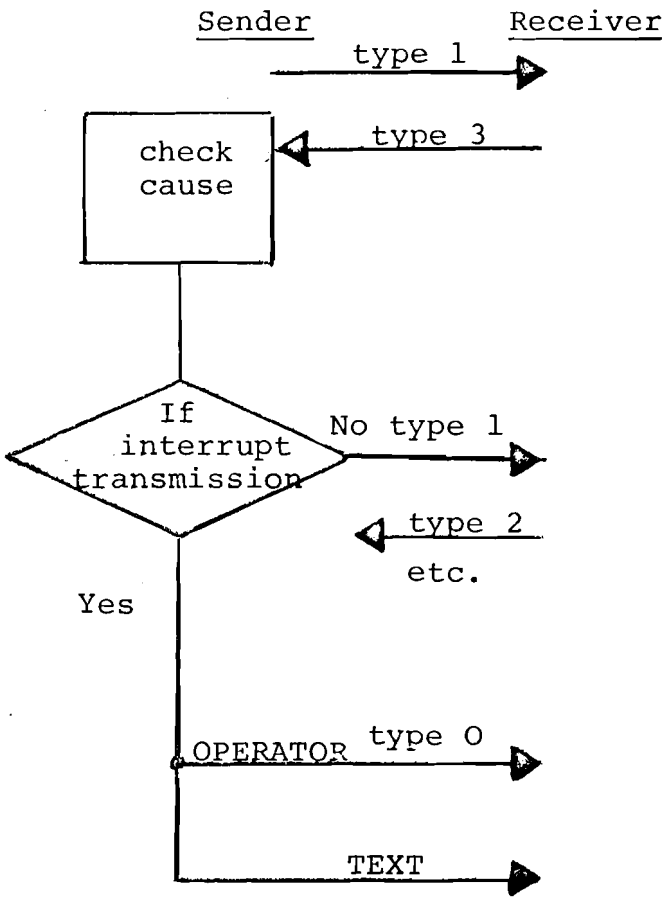
The following information is tested:

- is there an existing subscriber with given address;
- is it in an active state;
- may the sender communicate with desired subscriber;
- is the subscriber in connection with other subscribers;
- is the line to subscriber in order.

E. Tracing Facility

- current configuration of operating system (including number and type of subscriber);
- current state of all system programs;
- tracing of programs with exact addresses of executed instructions registration;
- current state of all queues to each node and subscriber, and state of the switching programs queues and free buffers;
- diagnostic and testing of communication lines and the programs servicing them.





Type of Packets:

Type 0 : control-diagnostic packet

1 : request packet

2 : positive acknowledgement

3 : negative acknowledgement packet, including cause: number busy, access barred, invalid call, out of order.

4 : positive acknowledgement of 6 packets

$\overline{4}$: negative acknowledgement of 6 packets

5 : end of transmission

6 : data

7 : test packet

Figure 7. End-user to End-user Call Job Structure

APPENDIX

Interfacing Problems on Linking the Warsaw Message Switching Node to the IIASA PDP Installation Running Data Link Control Procedures Described in CSN 004 IIASA Computer Project Paper

	IIASA Proposal	Warsaw Facility
1) Code	8 bits for disposing	7 bits for disposing 8-th bit for parity check only
2) Protection	cyclic redundancy check	BCC check character
3) Control characters	SYN, DLE, ETB, PAD	Possible to recognize
4) Addressing mode	Address byte	Acceptable
5) Command subset	Is using 8 bits	Unacceptable

Bibliography

[1] Radziminski, A. "Technical and Organizational Provisions for the Construction of an Experimental Data Communication Network Gdańsk - Warsaw - Katowice," June, 1973, Institute of Communication, Warsaw.

[2] Radziminski, A. "Switching Nodes in the Data Communication Network," June, 1975, Institute of Communication, Warsaw.

[3] Stagner, M. "NCOS Demo System," July, 1975, Institute of Communication, Warsaw.

[4] Kujawa, R., Lenarczyk, E. and Radziminski, A. "Connection Between the PDP 11 at IIASA and the Message Switching Node at Warsaw, Based on the Singer 10 Computer System." Computer Science Project paper, CSN 020, July, 1976.