DESY DV 79/02
November 1979

DESYNET

Talk presented at the

SEAS AM 79

Hamburg

by

E.-L. Bohnen

DESYNET

Abstract

DESYNET is a general purpose packet-switching network, that is currently in use for an interactive graphic terminal and plotting system. The network has some similarity to X.25 networks. It consists of mashed node controllers, which take over the task of construction and removal of virtual circuits, it provides flow control of each link and port and it handles error recovery in the case of erroneous data transmissions.

Throughout the whole network all connections to external applications are done using the same physical technique as well as nodecontroller connections (DATALINE: full duplex serial datatransmission and multiplexing system, 7 Mbit/sec).

DESYNET is based on simple minis (NOVA 1200/NOVA 3) without failure prone peripherals like disks. All minis in the net, nodecontroller as well as application machines, can be bootstrapped and loaded via the dataline system from the DESY computing center. All program development is done on the IBM too. Essential parts of DESYNET programming is done in PASCAL.

1. Introduction

At DESY the physicists try to look into the internal basic structure of the nature by shooting elementary particles on each other similar to a crash test. The result of such crash tests can be demonstrated by fig. 9. They are subject to further statistical interpretation. To do all that a sufficient computing power, various online facilities for data gathering and terminals especially graphical terminals to assist in data presentation are needed [1,2].

First let me make some remarks about the online activities at DESY and the history of DESYNET. Then I would like to talk about the aims of DESYNET and the current state in more detail.

Since DESY started to work the computing power of the computing center grew in a steady way from one IBM 7044 to now two IBM 370/168 and one IBM 3033. Parallel to this development we had increasing online activities in order to make the computing power accessable throughout the whole DESY area. Two systems have been developed in the past and are currently in use successfully:

- ONLINESYSTEM (PDA)
- DESYNET

The older one, the ONLINESYSTEM[3] (fig. 1), connects up to 28 stations for PETRA experiments and the control of PETRA itself. A tree of a multiplexer and several submultiplexers is controlled by one IBM-mainframe in a line switching mode, i.e. the whole system is occupied by one user at a time during one blocktransfer. A special supervisory program in the IBM takes over the task of sequentializing different transmission requests and the loading and starting of user written programs specific to each application.

The second system called DESYNET (fig. 2) has been developed during the last years and is currently under load for graphic applications.

DESYNET arose from the "Interactive Plotting System" (IPS) which was running before DESYNET. Initially developing IPS the point of main emphasis had been lain onto problems concerning graphics. After the installation of the IPS-system, especially after introduction of additional graphic terminals and plotters we found some severe problems, which resulted from the bad organization of the data transmission system.

These negative experiences essentially have influenced the claim to have a more generalized data transport system.

2. The objectives of DESYNET

The system should allow one to connect several sources and sinks, which can exchange data in a way, which is fast, flexible and reliable. System expansion should be possible by a simple "add on" process. A strong division between application problems and communication problems should be reached as well as a clean structuring of the various communication problems proper. This is the process of isolating and deleting unnecessary dependencies throughout the system.

2.1. Reliability

To reach reliability several steps on different layers should be done simultaneously.

2.1.1. Redundant data paths

Redundant data paths allow switching data streams from broken paths to yet working ones. Another aspect of redundant data paths is the load sharing (currently not implemented in DESYNET).

2.1.2. Error recovery

In order to discover transmission errors and to eliminate them by error recovery procedures additional redundant information bits must be generated and checked.

Confusious between communicating points for example on physical level: channel to channel connection or on higher level: end to end connection must be resolved by appropriate procedures which make the problems invisible to the outside world.

All errors and special effects should be handled on that level where they arose in order to have a clean cut and a simple interface to the next higher level.

2.1.3. Congestion and Deadlocks

The system must support mechanismus for controlling the flow of data streams, in order to prevent congestion and buffer deadlock situations. This is one of the most severe problems in designing networks [4].

2.2. Userprofile

An other aspect is the management of different types of load given to the net:

- Filetransfer (massdata)
- Interactive traffic (terminals)

DESYNET should be capable of handling both types of load. Filetransfer applications like IPS produce high loads to the net, therefore there is a need for high data rates on the transmission lines (>1 Mbaud). On the other hand the performance can be increased by suitable techniques for message propagation through the net. Parallel operation of several parts in the net leads to the pipelining effect and effectively speeds up the traffic.

In the case of interactive traffic which is caused by the interactive part of terminalsessions the overall response time plays the more important role. The response time caused by the net should be negligible and less than the response time causes by the time sharing system itself.

2.3. Connection to IBM mainframe

The connection of DESYNET to all three IBM-machines should allow simultaneous activities and mixed operations of different applications, whenever possible without any change or special solutious in the IBM operating system.

2.4. Costs

The amount of equipment, especially peripherals for network controllers, should be as small as possible, because of the multiplicative effect of an extendable network: original costs and costs for maintenance would explode rapidly. The standard service and resources of the computing center can be used to develop and update the network and application programs, so there is no need to have failure prone peripherals like disks and line printers at the outside stations.

On the other hand there must be the possibility to load any minicomputer via the net from computing center.

3. DESYNET

DESYNET is a general purpose packet-switching (store and forward) network consisting of several meshed nodes or NODECONTROLLERS. Nodecontrollers are connected via NETPORTS (fig. 3) and physical lines between them. Messages will be transmitted in blocks of a fixed maximum length (512 bytes per block). Each nodecontroller can be at the "surface" of the net, i.e. it can look to the outside world through USER PORTS which connect user applications to DESYNET. There is no physical difference between netports and userports. Both of them are capable of carrying up to 255 different and independent datastreams (LINK) distinguished by the link number. The logical difference is that a userport carries only datastreams belonging to a single user. There is some similarity between the userport and the CCITT x.25 interface which describes the rules between data circuit equipment and data terminal equipment (DCE, DTE) in public packet switching networks.

Userports can be realized inside a node controller by NETIN/NETOUT software calls (with well checked out application software) or by a pair of coax cables for a remote user connection.

Netport and remote userport connections as well are controlled by a line protocol procedure which isolates various transmission troubles and serves for flow control.

3.1 Virtual link concept

DESYNET allows to the user to build up connections between any link of his own port to any link of another userport in the net.

The basic use of DESYNET is done similar to using the telephone in 3 steps:

- dial up   (CONNECT)
- speak and/or hear via a full duplex line,
  - ° if necessary repeat parts of the message
  - ° adapt the speed, and in order to finish
- hang up the phone   (DISCONNECT)

To execute the CONNECT instruction DESYNET needs the physical destination address composed of 3 parts:

- NODE
- PORT of destination
- LINK

The linknumber can be omitted if the selected port represents a pool of neutral targets, i.e. the connection is made to a just free element waiting in a pool. If given in its logical form the first nodecontroller (nearest to CONNECT request) translates it into its physical equivalent. After receiving a connect request DESYNET tries to create a VIRTUAL LINK through the net from source to destination (fig. 3). The process of creating a virtual link - this is called a virtual call - is propagated from node to node until the target nodecontroller has been reached. The routing tables tell the link builder how to find the shortest way to the target node. Now a simplex connection exists and the real existence of the target can be checked. To obtain a full duplex line the process of linkbuilding is reflected at the target and then propagated back to the source of the connect request.

The virtual link is now ready to use and serves as a glide path for packets in both directions of traffic. It can be used as long as necessary.

DESYNET guarantees its deadlock free operation. This is reached by a so called CUT mechanism [3] during the connect phase of the link and a double buffer pool organisation during the life time of the virtual link. During link building the CUT mechanism possibly rejects the connect request.

Flow control for each virtual link is maintained by allowing up to one buffer per link and nodecontroller; thus the speed of traffic can be increased because of the pipelining effect (usefull in massdata file transfer applications).

One of the connected stations, tasks or endusers, can give up the whole link by means of the DISCONNECT command. All temporarily used resources along the way like control blocks, data buffer and task control blocks (TCB) are released and given back to the systems resource pool.

3.2 Control functions

Beside the main task of DESYNET to provide the user with data paths it offers some additional service for manual or automatic control:

- manual modifications of remote stations,
- measurements,
- aids for program development (debugging),
- and remote diagnostics.

This service can be obtained at any interactive terminal connected to DESYNET at any nodecontroller or terminal controller. There are some controlfunctions relating to global actions on the whole net and such which relate to specific objects like special links or special nodes.

- RESET cleans all activities along a virtual link, but does not release it. This feature can also be used by higher level protocols for end to end recovery procedures.

- FORMATION allows the manual setting of CUTS (restrictions for the link builder).

- Manual modification of ROUTING TABLES which is currently the only way to circumvent defective node controllers.

- Deletion of GARBAGE sometimes becomes necessary because of the problems of an automatic solution (the question is: is the link control block isolated or temporarily not in use?).

- Global modifications in all "telephone books" of the net allow the reassignment of logical to physical addresses (this can be used to make invisible to the user, on which actual mainframe his application program is running currently).

Some other powerful facilities for fault finding and measurements should be mentioned:

- remote dump,
- remote changes in core,
- generate, absorb and reflect test messages (artificial load),
- remote load.

3.3 Hardware

As mentioned before DESYNET consists of meshed nodecontrollers. A severe disadvantage of the meshed net is the rapid increasing of hardware equipment for network extensions. The costs for transmitters and receivers including cables and channel connection hardware in a fully connected net (fig. 5) can be estimated by the relation

$$\text{costs} = f\left(\frac{n^2 - n}{2}\right) + g(n) + ??$$

(where n is the number of nodes and f,g are approximate linear functions).

To reduce these costs connection of the nodecontrollers in DESYNET is based on the DATALINE hardware, which leads to a linear relationship between number of nodes and costs for a full connected net.

(certainly giving up some advantages of redundant data paths).

3.3.1 Dataline-concept

The dataline system (fig. 6 and 7) initially developed for graphic applications (IPS) is a very simple and modular concept for:

- high speed serial data transmission (7Mbaud) and
- multiplexing different datastreams.

The dataline hardware handles tiny non-divisable data elements, 32 bits in length, on a handshake basis. The format is as follows:

bit	6... Ø	destination address	(TO)
bit	14... 8	source address	(FROM)
bit	31...16	user data	(DATA)
bit	15	odd parity	(P)
bit	7	mode	(M)

A single basic hardware module (dataline module, fig. 6) has been developed, which contains one transmitter and one receiver to realize full duplex point to point connections.

Handshake is achieved by sending the acknowledge signal in the reverse direction using the same coax-cable. Double buffering of dataline words (32 bit elements) allows fast pipelined operation on the hardware level.

Besides the parallell to serial, serial to parallel converter, linedriver and buffer memory, the dataline module carries some additional synchronizing and address recognition hardware for multiplexing and routing purposes in an DL STAR.

The dataline module can be used in two configurations:

- terminal or enduser configuration or
- dataline multiplexor STAR-configuration.

The DL-STAR consists of an open end bus construction. For each outgoing bidirectional beam one DL-module has to be put onto the bus.

Outgoing beams can be connected to endusers or to other STARS again. This allows coupling of several clusters with a high degree of freedom in realizing individual data paths without any central supervisory control.

Each of the up to 128 endusers at the surface of the dataline network can transmit data elements to each other even to himself (this is very helpful for testing local star equipment).

The clustering can also be utilized to isolate different stations and to distribute the load on local stations.

3.3.2 DL-Protocol

The dataline offers a basic multipoint transport mechanism without any global rules. In order to construct symmetrical channels for reliable exchange of data packets between any station at the surface of the DL-net (fig. 7) we need a protocol[1] which solves several problems (fig. 8).

- RELIABILITY is reached by different types of redundancy for:

  ° data (parity, longitudinal parity),
  ° time (time limits) and the
  ° protocol itself.

  The protocol procedure tries to eliminate all detected errors by appropriate recovery mechanisms in order to make them invisible to higher software levels.

- The problem of FLOW CONTROL can be found in many places in designing networks. The basic idea of flow control is to push off problems (normally resource problems, for example no buffer or no channel currently available) into stream up direction in order to prevent punishment of the rest of the world. Some types of punishment are congestion and deadlock.

  On the other hand flow control allows asynchronous read and write commands initiated from higher software levels without knowledge of real hardware speeds.

- ADDRESS - conflicts are special problems of the dataline concept. They occur whenever any two stations try to write a packet to the same destination. By means of a FIFO technique simultaneous blocktransfer requests are serialized.

- Last but not least we need a device independant software interface which easily allows the partial replacement of DL-connections by other transmission techniques and protocolls whenever necessary (for example HDLC).

1) internally called "STECKDOSE"

3.3.3 IBM - channel connection

The connections of DESYNET to the IBM mainframes play an important role because currently and probably in future most applications need number crunching and mass storage facilities. So most connections are established to the mainframes.

Each of the three IBMs has its own access to a separate node controller (fig. 4) of DESYNET via programmable control units and selector channels. By aids of programms any standard or non standard IBM-device can be substituted or simulated.

One IBM-device or a pool of several devices of same type connects to one USERPORT of DESYNET. For the pool solution a dynamic assignment of unit numbers to linknumbers is arranged during runtime.

Currently accessible are pools of several 3277 - terminals and pools of 2701 parallel data adaptors. The IPS graphic terminals and plotters located at different places of DESY have access to these facilities using DESYNET as a transport medium |5|.

4. Final remarks

Let me conclude with some additional remarks.

- DESYNET is based on popular 16-bit minis (NOVA 1200/NOVA 3) without disks and line printers.
- All programs for DESYNET have been developed and checked out employing equipment of the computing center.
- Bootstrapping and loading of these programs into the minis is done from the IBM via the dataline (in 3 automatic steps).
- 3 modular programs have been written besides several application interface programs:

  MULTITASKING SUPERVISOR (Ass.),
  - memory mangement,
  - TASK-TASK communication,
  - generalized queue and resource handling,

  I/O SUPERVISOR (Pascal + Ass.),

  NETWORK CONTROL SYSTEM (Pascal + Ass.).

References

[1] J.M. Gerard: CERNET, The CERN Packet-switching Network

[2] W.L. Bauerfeld: HMINET, Protokolle, Übertragungsprozeduren und Interfaces

[3] D. Mönkemeyer: EXP-Manual, Interner Bericht DESY R1-76/04

[4] W. Wimmer: Ein Verfahren zur Verhinderung von Verklemmungen in Vermittlernetzen, DESY DV-78/05

[5] P.-K. Schilling: Computer Graphics at DESY, DESY DV-79/03

OBJECTIVES OF DESYNET DESIGN

- STRUCTURED DESIGN
- RELIABILITY
  redundant data paths
  error recovery
  congestion and deadlocks
  (flow control)
- USER PROFILE
  file transfer (mass data)
  interactive load (terminals)
- IBM MAIN FRAME CONNECTIONS
- COSTS, MAINTENANCE

CONTROL FUNCTIONS

– CONNECT
– DISCONNECT
– RESET
+ FORMATION
+ ROUTING TABLE CHANGE
+ GARBAGE
+ GLOBAL NAMES
+ REMOTE DUMP
+ REMOTE CHANGE IN CORE
+ TEST MESSAGES:
  (generate, absorb, reflect)
+ REMOTE LOAD

– normal operations on a link
+ for experts only or monitor programs

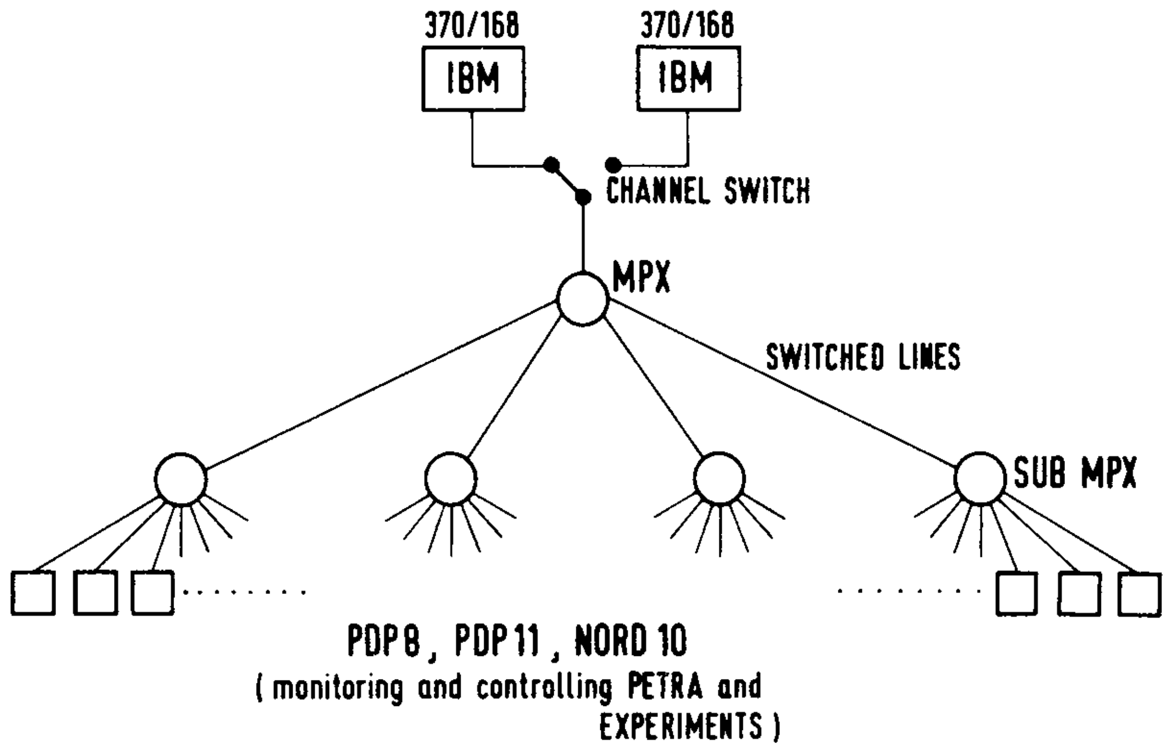


Fig.1: DESY ONLINE SYSTEM

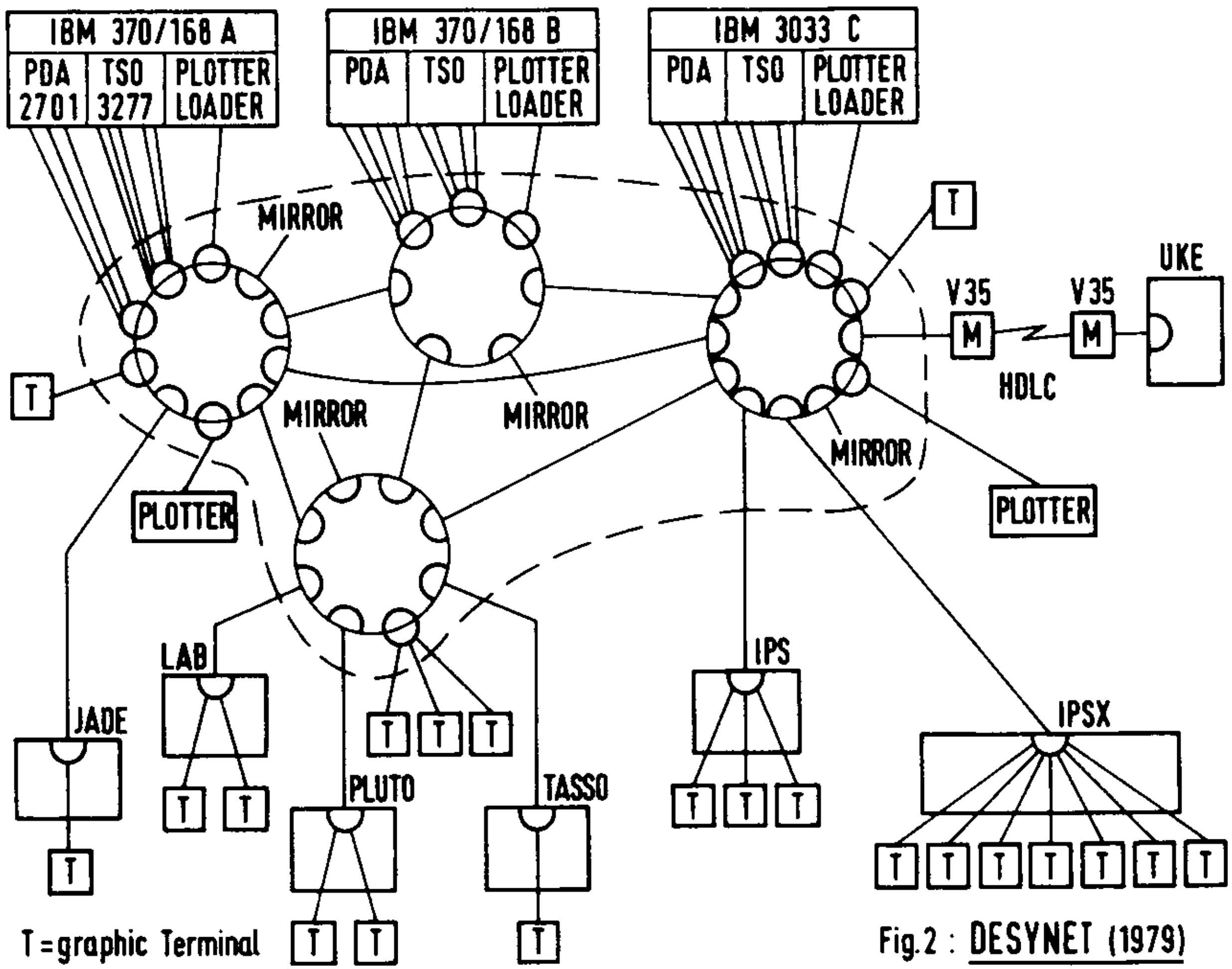


Fig. 2 : DESYNET (1979)

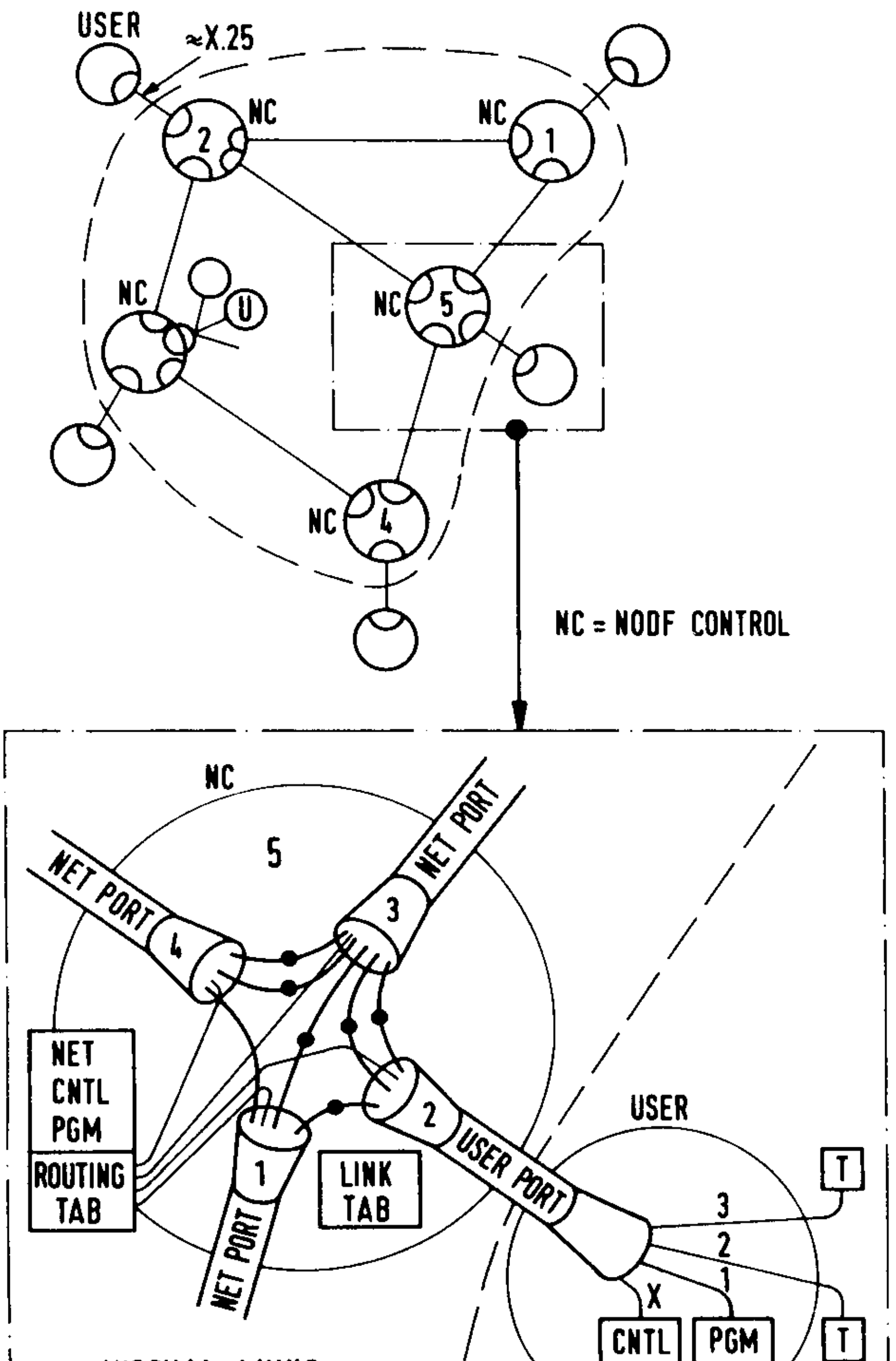


Fig.3: VIRTUAL LINKS

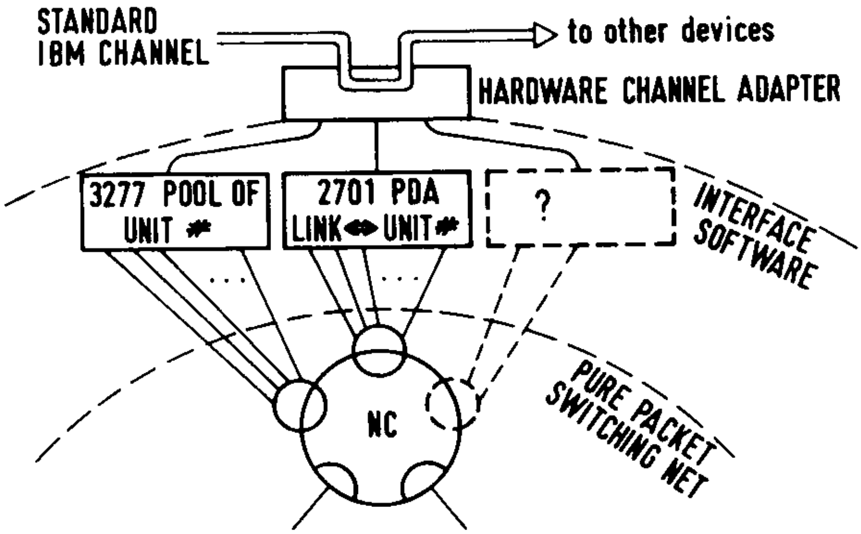
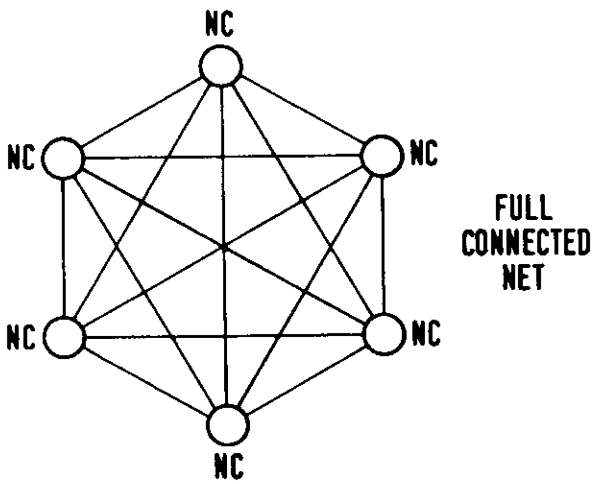


Fig. 4: IBM-DESYNET INTERFACE



$$\text{costs} = f\left(\frac{n^2-n}{2}\right) + g(n) + ?$$

where n is number of nodes

Fig. 5: FULL CONNECTED NET

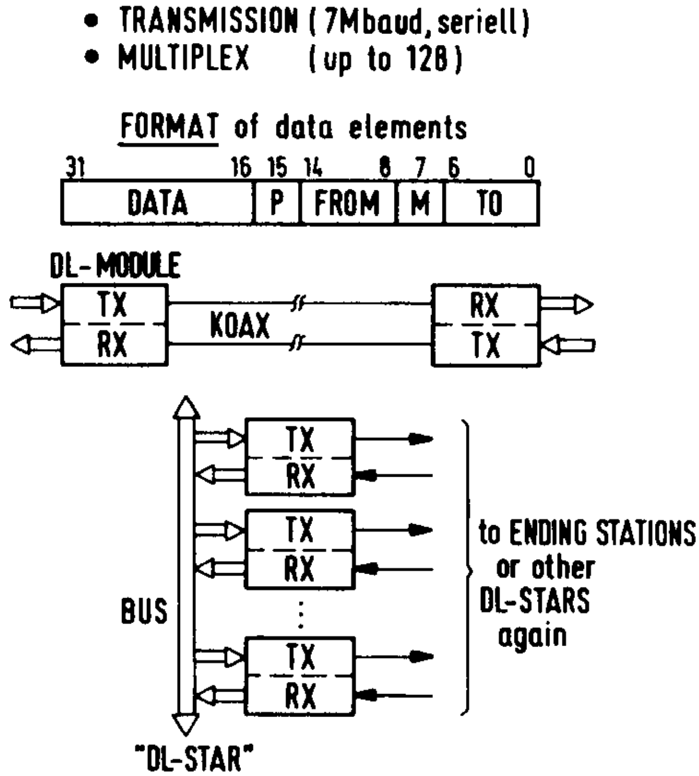


Fig. 6 : DATALINE

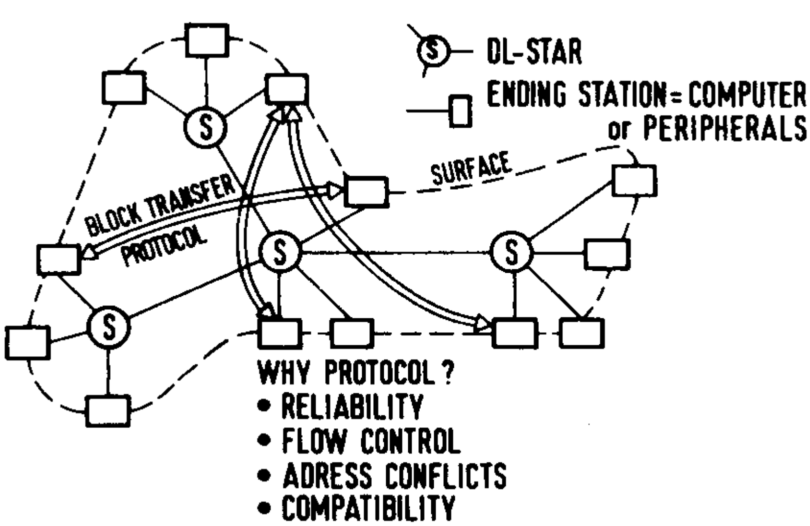


Fig. 7: DATALINE NET

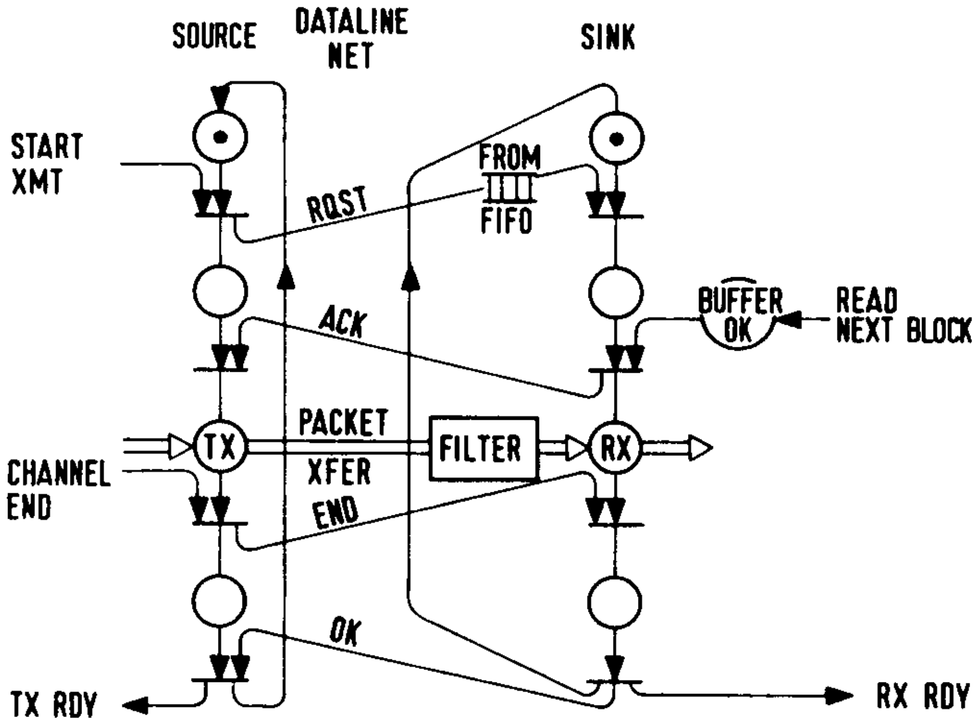


Fig. 8: Simplified DATALINE protocol
(without recovery)

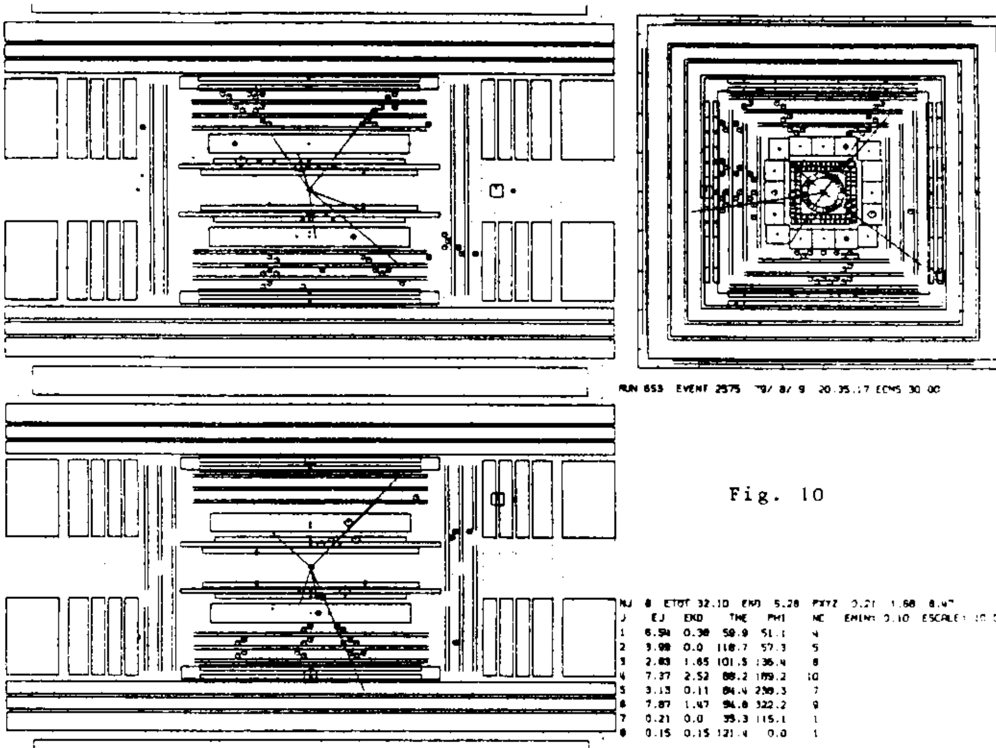


Fig. 10

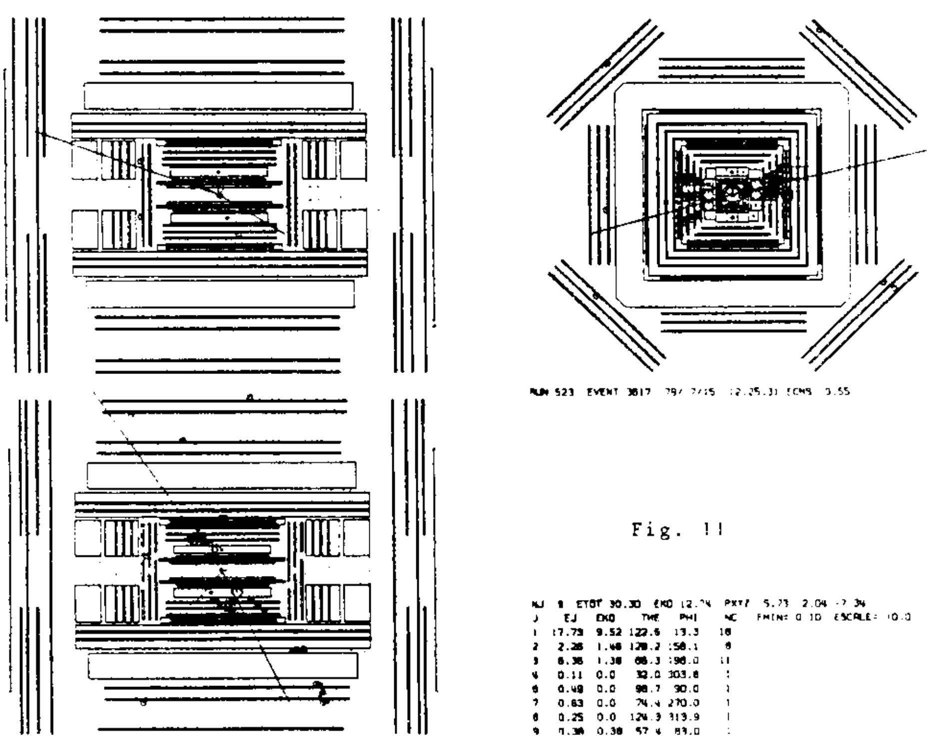


Fig. 11

Fig.9 Typical data transmitted via DESYNET