

PRESENTATION AND MAJOR DESIGN ASPECTS

OF THE CYCLADES COMPUTER NETWORK

by *Louis POUZIN*

Institut de Recherche d'Informatique et d'Automatique (IRIA)
Rocquencourt, France

SUMMARY

A computer network is being developed in France, under government sponsorship, to link about twenty heterogeneous computers located in universities, research and D.P. Centers. Goals are to set up a prototype network in order to foster experiment in various areas, such as : data communications, computer interaction, cooperative research, distributed data bases. The network is intended to be both, an object for research, and an operational tool.

In order to speed up the implementation, standard equipment is used, and modifications to operating systems are minimized. Rather, the design effort bears on a carefully layered architecture, allowing for a gradual insertion of specialized protocols and services tailored to specific application and user classes.

A particular objective, for which CYCLADES should be an operational tool, is to provide various departments of the French Administration with access to multiple data bases located in geographically distant areas.

Host-host protocols, as well as error and flow control mechanisms are based on a simple message exchange procedure, on top of which various options may be built for the sake of efficiency, error recovery, or convenience. Depending on available computer resources, these options can be implemented as user software, system modules, or front end processor package. For each of them, network-wide interfaces are defined, to conserve consistency in human communications.

CYCLADES uses a packet-switching sub-network, which is a transparent message carrier, completely independent of host-host conventions. While in many ways similar to ARPANET, it presents some distinctive differences in address and message handling, intended to facilitate interconnection with other networks. In particular, addresses can have variable formats, and messages are not delivered in sequence, so that they can flow out of the network through several gates toward an outside target.

Terminal concentrators are mini-hosts, and implement whatever services users or applications require, such as sequencing, error recovery, code translation, buffering, etc. Some specialized hosts may be installed to cater for specific services, such as mail, resource allocation, information retrieval, mass storage. A control center is also being installed and will be operated by the French PTT.

I - INTRODUCTION

CYCLADES is one the more recent computer network projects, which has been launched in France beginning with 1972. Its conception carries most of the characteristics found in the type of general purpose heterogeneous computer network such as experimented by ARPA, or proposed by NPL.

Our goals are to construct a prototype network in order to foster experiments in various areas, such as : data communications, computer interaction, cooperative research, distributed data bases. This action is twofold. In order to acquire valid experience, the network must also be used in a realistic environment, which requires a variety of operational services acceptable by customer standards.

In order to speed up the implementation, standard equipment is used, and modifications to operating systems are minimized. Rather the design effort bears on a carefully layered architecture, providing for an extensible structure of protocols and network services, tailored to various classes of traffic and applications.

This concern for built-in evolutionism translates itself in putting as few features as possible at levels buried in the sensitive parts of the network. With experience gradually building up, and depending on trends in international standards, more stable characteristics will eventually emerge. By putting them at some lower system level, it will be possible to obtain higher efficiency and reduce duplication, at the cost of freezing a few more parameters.

The Cyclades design attempts to be both precise and independent from the implementation at the user level, so that heterogeneous sites can have their way, and still communicate with others in a consistent manner.

II - PARTICIPANTS AND EQUIPMENT

Cyclades is sponsored by the Délégation à l'Informatique, a government agency in charge of coordinating all activities related to computing. Participating centers are only partially funded and put their own contribution on a voluntary basis. In a first stage, all network centers are research oriented organizations, universities, or engineering schools. In a second stage some D.P. centers of the French Administration will be connected to phase in real applications.

Participating centers are :

- Institut de Recherche en Informatique et Automatique (IRIA) (2 centers)
- Compagnie Internationale pour l'Informatique (CII) (2 centers)
- Météorologie Nationale (METEO)
- Institut de Recherche des Transports (IRT)
- Université de Grenoble (IMAG)
- Centre Universitaire de Calcul de Lyon (CCILS)
- Ecole des Mines de St-Etienne (MINES)
- Université de Toulouse (TOU)
- Centre d'Etudes et Recherches de Toulouse (CERT)
- Centre Electronique de l'Armement (CELAR)
- Université de Rennes (REN)
- Centre Commun d'Etudes de Télécommunications et Télévision (CCETT)
- Ecole Supérieure d'Electricité (ESE)
- Centre National d'Etudes des Télécommunications (CNET)

Computers are :

9 CII - 10070, 2 CII - IRIS 80, 2 CII - IRIS 50, 1 IBM 360/67, 1 CDC 6600, 1 PHILIPS - 1200. Communications computers are CII-MITRA 15.

The Cyclades topology is shown on Fig. 1. Transmission lines range from 4.8 kb up to 48 kb. The French PTT are providing lines and modems free of charge till end 75. Also they will run the network control center.

III - PLANNING

The Cyclades project was launched on the beginning of 1972. First host-host communications are to be tested in May 73, without packet switching, which will be introduced in June 73, on one node. Thereafter the network will come up gradually, until all hosts are connected in April 74. More centers will be introduced in 1975, along with real applications.

IV - GENERAL OBJECTIVES

I - Incremental implementation

Systems such as computer networks are still in the mainstream of research, and it would be inappropriate, if not unrealistic, to delay implementations until all issues are entirely understood, evaluated, and all possible functions completely designed. Some experimentation is necessary to gain insight, acquire know-how, and test hypotheses that appear initially in a most subjective context.

Furthermore, building a computer network is by essence a distributed effort, in order to create the motivations and common understanding so necessary for coordinating tasks and achieving network standards. Involving users in a proper way is a guarantee to have productive feedback and imaginative suggestions to cure the deficiencies of the network and extend its capabilities in a useful manner.

For all these reasons, Cyclades is being brought up stepwise and should be capable of providing some services at an early stage of implementation. Versatility, convenience, efficiency, will be phased in gradually, with the introduction of new components, and substitution of old ones.

2 - Design approach

Since Cyclades was not the first of its kind, it was more than advisable to study other networks before starting out. Most available documents were originating from ARIA⁸ and NPL¹⁰. A few ones were centered on other networks, MERIT², TYMNET¹¹, INTERNET⁹.

From this preliminary study and various live discussions with "networkers", one could draw some tentative conclusions :

- a - Data communications should be an independent sub-problem. Its main virtues are simplicity, reliability, transparency.
- b - Packet switching can work.
- c - Computer-computer protocols are still toddlers.
- d - Homogeneous computers are a lot easier.
- e - Ill defined protocols mean distributed headache.
- f - Computer communications require human communications.

Bearing these headlines in mind, the Cyclades design concentrated initially on the communications interface as seen at the basic user level. A common user interface was felt to be a keystone for building more elaborate network functions. From there on, the design proceeded inwards down to the packet switching interface, and outwards up to virtual terminals protocols, (TELNET like)³

By basic user level is meant in a broad sense a process executing a user program in a conventional operating system. Since Cyclades computers were deliberately heterogeneous, there were to be unavoidable variations in implementing the user-network interface.

Consequently, it was all the more important to produce specifications such that these local variations would not introduce ambiguities and misfits between any pair of users.

3 - Data transfer

Access to multiple data bases is a major operational objective for Cyclades, while time-sharing will take only a modest share. Even though a sophisticated system of distributed data bases may require some time to emerge, there will be a rising demand for file transfer, mainly because users tend to minimize adversity by splitting their tasks. Thus, basic protocols should provide for efficiency in using whatever bandwidth is available.

4 - Standards

Emphasis is put on using standards wherever possible, so as to protect present or future investments. A standard may be a set of recommendations promulgated by an official body. By default, it can be a widely accepted convention among network users. Specifically, communications hardware and procedures should conform to manufacturer or CCITT standards.

On the other hand, when standards do not exist, or are ill-suited, proper interfaces should insulate the domain, in order to allow for future adjustment, and defer commitment.

5 - Private user groups

In any large conglomerate of persons or associations, some groups tend to develop special ties and preferred relationship, based on common interest or necessity. Such a phenomenon should be expected as a natural ingredient of computer network sociology. Consequently, basic communications procedures should leave enough flexibility, at the user level, to allow for private conventions tailored to specific applications. On the other hand, standard network communications should be

compatible with this customization.

6 - Inter-network communications

The motivations for computer networks apply as well to networks of networks, which means that interconnecting with other networks should be a capability built in Cyclades. Presently, some networks communicate at terminal level. Although this may suit well some types of interactions, it is too restrictive for a broad class of applications. Interconnection at user, or communications network level should be anticipated.

V - COMMUNICATIONS ENTITIES

1 - Network model (Fig. 2)

All host computers communicate with one another through a host software called ST (*transfer station*), and a communications network 7. There may be several ST's within a host. Except for this latter characteristic, ST's correspond to Arpanet NCP's. They are local network subsidiaries within a host town 5.

Host entities, such as processes, users, devices, etc. communicate by exchanging *letters*, which are handed over to a local ST, shipped to the appropriate addressee's ST, and finally delivered to the destination entity.

Not every host entity may enjoy the privilege of sending letters using network services. To do so, one has to be formally introduced to the network as a *subscriber*. Roughly speaking, a *subscription* is a badge that allows its bearer to obtain network services, presumably at a cost some day. It is network business, viz. ST, to manage subscriptions, but it is host business to manage their association with local entities, and enforce rules for proper sharing and privacy. In other words, as seen from host, a subscription is usually a capability attached to a local process or user under host operating system protection.

2 - Subscribers

As seen from within the network, they are permanent names known network-wide. Opening and cancellation of subscriptions are administrative procedures which require some agreement from the network Authority. At a future stage of design, subscribers might be given capabilities and resource credit. For the moment, they are just global names.

Usual subscribers are attached to a particular ST : <global subscriber name> : : = <ST name> <local name> But there can be general subscribers whose location might change in time.

Typically, a subscriber could be a software processor, a subsystem, a human user, a device, a special answering service, etc. But this association is immaterial as far as the network is concerned, provided that basic exchange protocols are adhered to.

For the sake of convenience in network operation, particularly in human communications, it is expected that associations between subscribers and host entities will be rather stable, like the pair : person name - telephone number. It should be worthwhile to print and disseminate subscriber directories reasonably up to date. If at all necessary, administrative delays or costs will be tacked on subscription changes to make them sufficiently unrequent.

Since most subscribers will use network services only occasionally, it would be wasteful to maintain subscription information at all times within ST's in high speed

memory. Therefore, a subscription can be *enabled* or *disabled*, very much like login-logout for a time-sharing user. This operation is executed dynamically on subscriber's request.

3 - Ports

Many software systems deal with data exchange in terms of flows, streams, channels, or similar concepts. One could argue on whether this is a so called natural way or if it is just bequeathed by a persistent addiction to card readers, magnetic tapes, and other sequential devices. Nevertheless, I/O-like techniques permeate most forms of inter-process communications. The concept of port has come to be commonly used to designate an abstract entity on which data flows may be anchored, and addressed.

To that effect, a subscriber can apply to its ST for port names. They are created dynamically and are local to the subscriber. They can also be exchanged between ST's, as part of specific protocols, to set up links between subscribers.

In other words, subscribers and ports make up a hierarchical name space, network-wide. The subscriber component is global and basically stable, while the port component is local and basically transient.

So far we have not felt the need for further levels. Should it appear useful, growing sub-ports would not be a technical problem.

4 - Letters -

It is a piece of information exchanged between two subscribers. There may be several varieties of letter mechanisms. Presently, 4 have been designed.

a - Regular letter

It can be sent at any time to any subscriber, as long as both subscriptions are enabled. A priority may be specified, and an acknowledgement may be requested. By acknowledgment is meant a return message sent back to the sender subscriber, after the letter has been delivered to the receiver subscriber. A letter can contain up to 240 octets of text (1920 bits).

b - Liaison

Letters are sent over a port, and delivered from a port. An initial set up is necessary to open a liaison, and exchange port names, which are only paired by order of creation. The liaison machinery includes error and flow control, and it is bidirectional. A symmetrical procedure solves all contention problems.

c - Connection

It has the same properties as a liaison. But letters are delivered to the receiver subscriber in the same order as they have been sent. Furthermore, letters can be indefinite strings of bits. The connection machinery includes error and flow control, and it is bidirectional. The same symmetrical procedure as for liaisons applies to connections.

d - Events

They are short letters (16 bits) transmitted with higher priority. They may be sent separately or over an existing liaison or connection amidst text flow, as out-of-band messages.

The previous set of mechanisms is intended to provide basic user facilities on top of which more sophisticated

services may be built. Each one is aimed at a particular class of traffic which is expected to be frequently encountered in the network.

Regular letters are intended for conversational traffic between slow terminals and server processes. They can also be used as control messages between several processes cooperating within a distributed activity.

Liaisons are intended for bulk traffic such as file transfer or data base processing, where letters contain self-identifying items of information, and are well suited for parallel processing.

Connections are intended for I/O streams, typically remote sequential devices or files, as well as conventional inter-process communications.

Events are intended for control information when it is desirable to send it asynchronously with data flow. A typical case is attention or diagnostics messages to be used by a control environment rather than the normal receiver process.

VI - FUNCTIONAL COMPONENTS

1 - Component hierarchy (Fig. 3)

Starting with the communications network, one finds :

a - A host communications interface, implementing a line transmission procedure. Initially, it will be one of the bi-synchronous family, depending on the host at hand. In the future, an ISO standard procedure of the HDLC type¹² will be installed, when I-O adapters will be available on the market. One may notice that a host can have more than one physical link with the communications network, to provide for more reliability in case of node or line failure.

b - A transfer station (ST), implementing the subscriber name space, ports, and a basic letter handling at an intermediate system interface. There may be several ST's, for testing new versions, implementing special services, and communicating with foreign host protocols, e.g. Arpanet, or Cost-11¹³. Monitoring and diagnostic aids are also introduced at this level.

c - A set of user oriented letter handling functions implementing error and flow control, queue management, and liaison/connection management when applicable. This approach results from the recognition that there is no ideal way of handling data exchange. It depends on user environment. Rather than piling layer upon layer of functions, with the associated overhead and duplication, it appeared more efficient to leave room for expansion not only upwards as usual, but also sideways. Again, it becomes a casual matter to try out new options, and to develop private network access methods, without losing the benefit of standard interfaces.

2 - Transfer station structure

Our objective was not limited to specify a set of rules for exchanging messages between hosts. Rather, it was ideally to write the specifications of a complete ST, including various letter handling, so that every network user would have a common interface, regardless of the host type.

It is clear that this is a trivial problem in homogeneous networks. One possible approach for a heterogeneous network would be to use a portable programming language. But operating system peculiarities introduce a variety of discrepancies and inefficiencies.

Consequently, the design could not be so ideal ; it could only attempt to define functions without referring to specific host facilities. This objective resulted in the following scheme.

An ST is thought of as an abstract machine, (Fig. 4), driven by commands, and exchanging information with the external world through a communications area. Some internal states may be observed through a glass window. Communications mechanisms are implementation dependent, but they should not bear any relationship with the ST internal logic. On the other hand, they can be implemented using well known engineering techniques.

An ST is then further broken down into individual components given maximum autonomy (Fig. 5). So as to allow for implementation freedom, individual components can be thought of as asynchronous machines cooperating via state variables, or queues. These constituent machines are listed below.

- Command : checks arguments and signals appropriate machine ; 1 mach
- Subscription: enables/disables subscriptions ; 1 mach/subscriber
- Regular letter : send/receive regular letters ; 1 mach/subscriber
- Port : handles ports ; 1 mach/subscriber
- Liaison : handles liaisons ; 1 mach/subscriber/liaison
- Connection : handles connections ; 1 mach/subscriber/connection
- Communications : send/receive packets ; 1 mach
- Debug : special modes ; 1 mach
- Operator : manual/automatic control ; 1 mach

Readers may not have failed to notice the structured programming approach used as a design methodology⁴. Of course, implementation may deviate somehow in making machines less autonomous, such as sub-routines. But the design structure should be kept highly visible, or else unanticipated interferences may well creep in.

The logic of each machine is specified in natural language algorithms, using a loose form of pseudo-Algol. It was not felt at this point that a genuine programming language would have helped human communications.

3 - Subscriber interface

Although communications between a user process and an ST machine are implementation dependent, it was considered important to specify ST commands in a non-ambiguous way, resembling a sub-routine or macro-call. Therefore, all commands have been given some sort of formal representation, using mnemonics and argument names, as they should be passed over to the ST. Message formats and states are also specified.

E.g : OPEN, LI, LOC-SUB, DIS-SUB, LI-X, MIN, MAX

meaning : open a liaison from local subscriber to distant subscriber, liaison number, minimum and maximum characteristics proposed in terms of buffer allocation, letter length, bandwidth.

In an actual implementation, OPEN LIAISON could be a system primitive, DIS-SUB and LI-X could be arguments in registers, MIN and MAX packed into a liaison control block, and LOC-SUB supplied by the operating system.

4 - Communications network interface

The ST makes packets out of letters, and vice versa.

Letters are stitched with control information, and if at all possible several letters are blocked within a single packet towards the same destination. Infinite letters are fragmented. Thereafter, the packet is passed to a line handler to be delivered to the communication network. The packet format is :

Header : 72 bits (9 octets), Text : 2040 bits (255 octets) max. As usual, additional bits are inserted when output to transmission lines takes place : synchro, CRC, etc.

VII - PACKET SWITCHING

Packet switching technology is just emerging, and building a computer network is an adequate opportunity to experiment and acquire know-how in this domain. So far well defined problems have been solved quite satisfactorily, e.g. fault detection, remote loading, packet ordering, etc. On the other hand, there are ill defined problems, such as congestion, routing, topology, which are only partially understood, and most likely interdependent. Our concern in a first stage is not to make breakthrough in packet switching technology, but to build a reliable communications tool for Cyclades, while preserving the possibility of a major redesign when more experience becomes available.

Consequently we have been very strict in insulating logically, and even physically, functions related to computer network on one hand, and those germane to packet switching on another hand. E.g. terminal concentration will be done by mini-hosts containing a stripped down ST, implementing unsophisticated connections.

Some specific features of our packet switching network, called CIGALE, are presented in the following.

1 - Addressing

The basic purpose of a packet switching network is to deliver messages to an addressee located outside of the network, and not to reach its own components. Therefore, there is a need for a global name space network-wide, to designate source and destination of messages. In Arpanet such a name space maps onto network components, viz. node and line number. There are two consequences : a - addressees can only be reached through a unique gateway, b - topology changes may require address changes.

Let us assume that an addressee is not a single computer, but a distributed computer, i.e. a network, then it will likely be required to link them by multiple paths, for reliability, traffic smoothing, response time, etc. An addressee becomes an outside target, which may be reached through several possible gateways. Thus, we need an independent name space.

In Cigale, names are ST's, which can be reached from several nodes. Furthermore, there may be several ST's on one line. In a large network, it would be a severe constraint if every node should know all possible addressees. Therefore, we use a hierarchical name space : region - ST.

Each node has only to know region names, and ST names within its own region. Any ST belongs to only one region. But let us note that this does not prevent from reaching an ST directly from a node in a different region, as long as this ST name is also listed in the node name space. But this practice should be restricted to isolated ST's, as it tends to make address look up more costly.

International communications will have to deal with a jumble of address formats, and the only practical way out will be to introduce variable formats as a way of switching. This will bring another hierarchical structure, for which every network should be prepared. In anticipation of that, an address type component is provided in Cigale.

The general address format would be :

type (3 bits), region (5 bits), ST (8 bits)
But Cyclades does not need such a large name space, and some bits may be set aside for future use, leaving region (4 bits), ST (4 bits).

2 - Internal ST (STI)

Some special functions are useful within a packet switching network, e.g. collecting bad messages, echoing, traffic generation, ... Rather than having specially formatted packets along with the decoding software, it is much easier to reserve a subset of the ST name space to address those special components. Since the general addressing mechanism applies, STI's may be either located within nodes, or be within real hosts considered as extensions of the network, and supported by the network Authority. Also some services may be experimented within a real host, and once approved, integrated within the network functions.

In commercial networks some services should be offered to attract customers and add more convenience : e.g. data conversion, mailboxes, broadcasting, file editing, etc... Using STI's is a handy way to hook those services without disturbing network operation.

In Cigale, some address variations allow a few STI's to be located : - at only one node, - at every node, - at some nodes. Through that facility, STI's may be distributed according to traffic requirements, and even moved about the network during operation.

3 - Message reassembly

Cigale does not fragment messages. It only takes in packets.

4 - Message ordering

Cigale delivers packets as soon as they arrive at a destination gateway. There is no ordering. On the other hand ordering does not appear compatible with multi-paths to a host.

5 - Flow control

Cigale does not apply flow control to any specific flow. On the other hand, it will attempt to resist congestion, but the techniques to be used are not yet clear. An approach would be to allocate input traffic according to available buffer space, using exponential smoothing. Simulation studies are planned.

VIII - INTER-NETWORK COMMUNICATIONS

Inter-network communications have still to demonstrate their practical feasibility, if one excepts the present situation where a network mimics a terminal to the other. It seems that key-points include simplicity and open-endedness.

The more sophisticated a network, the less likely it is going to interface properly with another. In particular, any function except sending packets is probably just specific enough not to work in conjunction with a neighbor. The result is an intersection of properties rather than a union.

In this respect Cigale does not present any excess properties. All functions are self-contained, none extends across network boundary. As long as packets are within the maximum size, with proper format, they will be delivered to a known ST. Some mismatch may result from error messages sent back to the source in case of wrong progress. A possible solution would be to use an STI as middle man, in charge of the necessary format conversions.

Trans-network communications bring up another problem. Assuming that interface problems are solved, intermediate networks, supposed to carry messages along, do not possess the final destination in their name space. Thus a new function arises : international routing. But this is a general problem unrelated to specific network characteristics.

Cyclades hosts are also well suited to inter-network exchange, since : - their basic letter protocol is simple, - more ST's or protocols can be added. In the worst case a special ST must be built to interface with a foreign host. However there may be some devious timing problems that can probably be solved on an ad hoc basis. But this would provide only a straight-forward type of exchange. The whole set of procedures and practices that make up a computer network environment is a much larger task.

IX - CONCLUSIONS

Cyclades is one of the largest computer projects in France. Major universities and research centers are actively working on its development. It is expected that techniques and insight acquired during the project will benefit research and industry, specifically at a moment when several communications networks are being planned by large corporations, and the French Administration. Its extensible structure at several levels makes it well suited to all sorts of experiments on a national and international scene.

X - ACKNOWLEDGMENTS

The Cyclades design is largely a teamwork, and it is quite difficult to trace back the genesis of ideas. Our work stemmed mainly from earlier research accomplished at NPL and within the ARPA community. Among individuals who contributed major parts of the design, are M. ELIE (CII) and J.L. GRANGE (Cyclades). A particular acknowledgment is due to H. ZIMMERMANN (Cyclades) for an outstanding contribution on host protocols (ST). Stimulating discussions with D. WALDEN (BBN) brought substantial improvement and simplification.

- 1 - CARR C.S., CROCKER S.D., CERF V.G. - Host-Host communication protocol in the Arpa network. SJCC (1970), 589-597
- 2 - COCANOWER A.B. - Functional characteristics of CCOS.MERIT computer network, (Jun. 71), 39 p.
- 3 - CROCKER S. et al. - Function oriented protocols for the ARPA computer network. SJCC (1972), 271-279.
- 4 - DIJKSTRA E.W. - Notes on structured programming. (Aug. 69), 84 p.
- 5 - ELIE M., ZIMMERMANN H., et al. - Spécifications fonctionnelles des stations de transport du réseau Cyclades. SCH 502, (Nov. 72), 105 p.
- 6 - GIRARDI S. - SOC project, an experimental computer network. Intern. Comp. Symp. Venice, (Apr. 72), 210-220.
- 7 - GRANGÉ J.L., POUZIN L. - CIGALE, la machine de commutation de paquets du réseau CYCLADES. Congrès AFCET 73, 24 p.
- 8 - ROBERTS L.G., WESSLER B.D. - Computer network development to achieve resource sharing. SJCC (1970), 543-549.
- 9 - RUTLEDGE R.M., et al. - An interactive network of time-sharing computers. 24th ACM Nat. Conf. (Aug. 69), 431-441.
- 10 - SCANTLEBURY R.A. - A model for the local area of a data communication network - Objectives and hardware organization. ACM Symp. on problems in the optimization of data communications systems (1969), 179-201.
- 11 - TYMES L.R. - Tymnet, a terminal oriented communication network. SJCC (1971), 211-216.
- 12 - ISO/TC 97/SC 6. Doc. 731 - HDLC procedures. Proposed draft international standard on frame structure (Feb. 73), 4 p.
- 13 - BARBER D.L.A. - The European computer network project. ICCS, Washington D.C., (Oct. 72), 192-200.