USER-ORIENTED NETWORKS: A SERIES
PART I-A. FOREWORD AND PURPOSE OF SERIES.
PART I-B. DEFINITION, OVERALL GOALS, AND PROBLEMS.

Wm. Orchard-Hays

July 1975 WP-75-81

PART I-A. FOREWORD AND PURPOSE OF SERIES

This paper (Parts I-A, B) is the first of a series of working papers on data processing networks of a certain type. These are termed User-Oriented Networks and defined in Part I-B following. The need for some such series of discussions is brought about by current activities at IIASA aimed at creating a telecommunications network connecting Laxenburg with various centers in the NMO's and with at least a few large computing centers. The goals of such a network differ from those of conventional, existing networks and the nature of the network, if implemented as now being discussed, will introduce a number of new operational problems. The main purpose of this series is to bring to light the kinds of problems which will be encountered and to suggest designs which will minimize the difficulties and enhance overall effectiveness.

This series is concerned only peripherally with hardware, per se. Other investigators are much better equipped to evaluate particular componentry and necessary hardware interfaces and line protocols. At the same time, however, suggestions which will be made will have strong implications for capacity and compatibility among nodal units and the interconnecting telecommunication lines. In some parts of the series, rather specific recommendations will be made for computers needed as nodal switch points.

For the most part, this series is concerned with programming considerations, both for operation of a network and for its use. The programming of application software systems or application programs on large computers is not a principle area of discussion but, of course, the requirements for such work will arise from time to time and provide major considerations for network capability. The term "programming" is used herein in a broader sense which encompasses the flow of data among nodes of the network and the kind of higher-level protocols, command and control languages, symbology standards, etc., which will be required.

This writer is aware that many suggestions to be made--or their implications--may be impractical at the present time for either financial or political reasons. However, these cannot be considered permanent constraints or the network can never be realized. On the other hand, it is not intended that anything will be suggested which is technologically or logically infeasible. In general, it is hoped that the series will arouse discussion and criticism by a number of people at IIASA. It is necessary to think very carefully about the consequences of any design decision in such a long-range and far-reaching undertaking as an international network of the kind being contemplated.

Part I-B following defines the kind of network contemplated in terms of overall goals and then discusses nine general classes of problems which are immediately foreseen. Part II will take up the inter-user communication problem and suggest an overall network design which satisfies these requirements. Part III extends the discussion of Part II to user-system communication and introduces the necessary higher-level protocols to accommodate it. The overall design, terminology and style implied by Parts II and III will be the basis for the remaining discussions unless criticism is so severe as to require heavy revisions.

Following parts, after III, will deal with areas of operational convenience, system compatibility, symbology, and network effectiveness. The exact breakdown is not yet completely formulated and will depend in part on reactions to Parts I, II and III.

PART I-B. DEFINITION, OVERALL GOALS, AND PROBLEMS

TABLE OF CONTENTS

INTRODUCTION

There exist already two, or perhaps three, types of computer networks. The first arrangement is where a large computing center--or perhaps a small number of highly integrated centers--provides services to a large number of users in many cities or areas. This is done as a profit-making venture. Leased telephone lines connect concentrators in various strategic locations to the main center or centers. Users near these concentrators may then access the system via local telephone calls. The area served by a concentrator may be expanded somewhat by extended area telephone service. All these costs are included in the overhead of the company. The enterprise controls the range of equipment, the scope of basic software whose integrity it maintains and assures, and the protocols necessary to use the system. Each user must have an account with the company and is billed directly by the company. Such an arrangement may be called a commercial network. A number of heavily-used and successful ones now exist. There is a tendency toward specialization of clientele, and hence of services.

The second arrangement involves a network enterprise, as such, which provides lines to both a variety of computer centers and a variety of localities. Similar hardware and leased lines are used as in the first type. However, there are additional, standardized network protocols*--which degrade response time slightly and introduce another source of error and breakdown. However, these disadvantages are not severe. The idea is to make a large number of centers--primarily in universities and research centers--available to a large number of research organizations. Having gone through the network protocols, the user must then abide by the protocols and conventions for the center and system which he accesses. These are relayed by the network verbatim. A user pays for the network service and additionally must have an account at each center he uses, for which he pays on an as-used basis. (The center charges

---

*In fact, these also exist for commercial networks but are usually very simple since all accounting is centralized.

are usually lower than in the first arrangement.) Such an arrangement may be called a cooperative network and several are now in regular use.

A possible third type is a data-base network. However, these are little more than special cases of the first two, and are sometimes merely an adjunct to or auxiliary service of other networks, particularly commercial. There are, however, both special problems and potentially different telecommunication arrangements for data-base networks. More will be said on this later.

IIASA has a need for and is beginning to create a network which differs from the above types. All the above types might be utilized by IIASA for various needs but no one or two would satisfy all its requirements. The concept emerging at IIASA might be termed a user-oriented network. The first approximation to such a network might closely resemble a cooperative network and, indeed, the most heavily used facility to date is essentially a commercial network (except that specific equipment at IIASA is used in place of a concentrator). However, neither arrangement is adequate for IIASA's full scope of interests. The following is a preliminary discussion of the nature of a user-oriented network such as IIASA appears to require, followed by some observations on various problems which must be overcome.

A USER-ORIENTED NETWORK

In order to describe a user-oriented network, one needs first to state its purposes. The following list of goals is less in order of importance than in order of difficulty.

1. To access the best computing facilities for required jobs.
2. To access data bases stored in various places.
3. To minimize time delays due to network or center overloading or breakdown.
4. To minimize cost per job, other things being equal.

5. To permit and foster collaboration and work-load sharing by dispersed groups in the NMO's and and other research organizations.
6. To permit and foster on-line, interactive collaboration on an international scale.
7. To monitor research efforts, avoid duplication and provide quick information on past and present research, among all participating organizations.

A cooperative network could satisfy 1. It might also easily satisfy 2. though there are some technical difficulties; more troublesome are political problems. This impediment will turn up in all succeeding goals and will be discussed further in the next section. In substance, however, goals 1. and 2. are now technically, and perhaps even financially, feasible.

Goal 3. involves two main concepts: matching of jobs to facilities by selection of "best" center currently available; and selection of best routing to that center on the network. Both of these areas have a number of ramifications. Some promising work has already been done by the Computer Science Project and Computer Services which bear on these concepts.

Goal 4. is largely an extension of 3. but involves estimations of cost effectiveness of different systems for classes of problems. Such data can be obtained only by experiment. Comparisons of this kind have been made by a number of user groups or user centers but the results are hard to locate and sometimes harder to interpret.

Goals 5. and 6. are closely related and probably constitute one of the two main capabilities in which IIASA is interested (the other being 7.). It is not sufficient merely to have comprehensive capabilities at IIASA. Continuing, coordinated effort within the NMO's is also desirable and, indeed, it is probably necessary if continuity

is to be maintained in work done at IIASA. Visiting researchers, for example, need to be able to gain some degree of familiarity with IIASA's facilities beforehand, so that valuable time is not lost when they arrive. Similarly, after returning to their home base, they need to be able to continue work with their programs, files, and procedures developed here. Furthermore, it may often be desirable to allocate subtasks to the NMO's for projects coordinated at IIASA. To be effective, this requires some degree of inter-facility communication and compatibility.

Goal 6. looks beyond merely coordinated effort by various centers to real on-line, interactive collaboration. This poses severe additional problems of both a technical and a political nature.

Goal 7. would be fairly easy to realize if all the other six had been achieved. However, it is desirable to have this capability in some degree before all the difficulties of goals 1. to 6. are overcome. Considerable research into information content, format and control can be done abstractly. As these designs are firmed up, parts of an information system can be implemented, at least locally, which will be compatible with the full capability hoped for in the more distant future.

In addition to these desired capabilities, there is clearly a need for a centralized and automated accounting system.

CLASSES OF PROBLEMS TO BE SOLVED

The problems which will be encountered in realizing the foregoing goals are not in a one-one relationship with them. Such a classification would be either strained or redundant. A more meaningful breakdown is shown below. The order here is roughly from lowest to highest level.

A) Hardware Compatibility and Telecommunication Interfaces.

B) Network Protocols and Switching Techniques.
C) System Philosophy (Style) and Data Management Conventions.
D) Data Input Formats; File Structures and Organizations.
E) Processor (Software) and Data-Base Compatibility.
F) Classification of Jobs and Tables of System Efficiency.
G) Command and Control Languages.
H) Responsive and Effective Maintenance Procedures.
I) Security and System Integrity, plus Accounting Procedures.
J) International and Inter-organizational Political Problems.

Within these broad classes, some overlapping problems exist. This is particularly true with what has been termed political problems. For example, Security, Accounting an Maintenance Procedures all involve political considerations. However, there are other overlaps. Switching techniques are dependent both on available equipment in different countries and on the intended mode of use. For example, packet-switching might be found most effective for large data transfers, such as RJE or data-base access, but might be unworkable for real-time, interactive operation.

Permeating all considerations, of course, is the question of financing. It is clear that such an undertaking as that envisioned will be expensive. No attempt will be made here to assess fiscal feasiblity. In some cases, the effect of the availability of substantial funds will be noted.

Each of the above classes is discussed in varying degrees of detail in the sequel.

Hardware Compatibility and Telecommunication Interfaces

Before any kind of computer network can exist, electronic machines must be able to send signals to each other. When one considers a network linking different computer systems,

then one must, of course, arrange interfaces between their differing electrical characteristics and logical conventions. Since distance implies communication links--essentially telephone lines--interfacing between the computers and the telephone system is also necessary. A variety of proven equipment exists to perform these functions. In the USA with its unsurpassed Bell Telephone system, one need scarcely worry about anything more except cost. In Europe, however, with its numerous international boundaries and national telephone systems, there are additional problems. Clearly, equipment is in place for connecting virtually any two telephones and also for computer networks across boundaries. Nevertheless, one cannot place a call from, say, Vienna to Bratislava as easily or quickly as from New York to San Francisco, roughly a hundred times the distance. Furthermore, the additional interfaces which must exist--though unknown to the user--are another potential source of error and breakdown, as well as delay. The extent to which this situation will be an impediment to an international network needs thorough investigation.

Given enough money, then it is clearly possible to electronically link nearly any two pieces of equipment. Compatibility, per se, should be achievable with essentially standard equipment. However, speed of access, reliability, speed of transmission in volume, and freedom from interference are also essential characteristics.

Network Protocols and Switching Techniques

Given an electronically and operationally feasible set of interconnections, there are still decisions to be made regarding the mode of transmission. Some of these considerations depend on the equipment in place at nodes of the network and perhaps at intermediate points. It seems clear that the network envisioned will require dedicated lines, available for transmission continuously during the hours in which the network is operational. Not

all parts of the network need be on-line during all hours, perhaps, but is is impractical to wait until a demand is made to dial up the appropriate center. There must be pre-scheduled periods of operation.

For greater clarity, let us make the following definitions:

Switch point	(S)	A junction of incoming and outgoing lines at which a routing option must be taken.
Node	(N)	A switch point where at least one pair of lines (in, out) connects to terminal equipment (computer, user, or both).
Local Computer	(L)	A node which has some computing capacity available to local users.
Computer Center	(C)	A node at which there is large-scale computing and storage capacity.
Master Control	(M)	The IIASA node.

It is possible that a network contains no switch points which are not also nodes. On the other hand, some nodes may have only one pair of connections besides those to terminal equipment. Both situations are illustrated by the hypothetical setup in Figure 1. (The dotted lines show connections of minor interest to IIASA.) It is, in fact, unlikely that the network as such would have switch points which are not nodes. The telephone lines probably have their own switch points but these are below our level of recognition here.

Figure 1, which though hypothetical is not far from possibility, discloses several problems. Consider first Moscow. The center there, let us say, is primarily for use on national problems and is connected to an internal network. Even theough they might be anxious to participate in the IIASA network, they might not be willing to install additional terminal equipment, different from what they

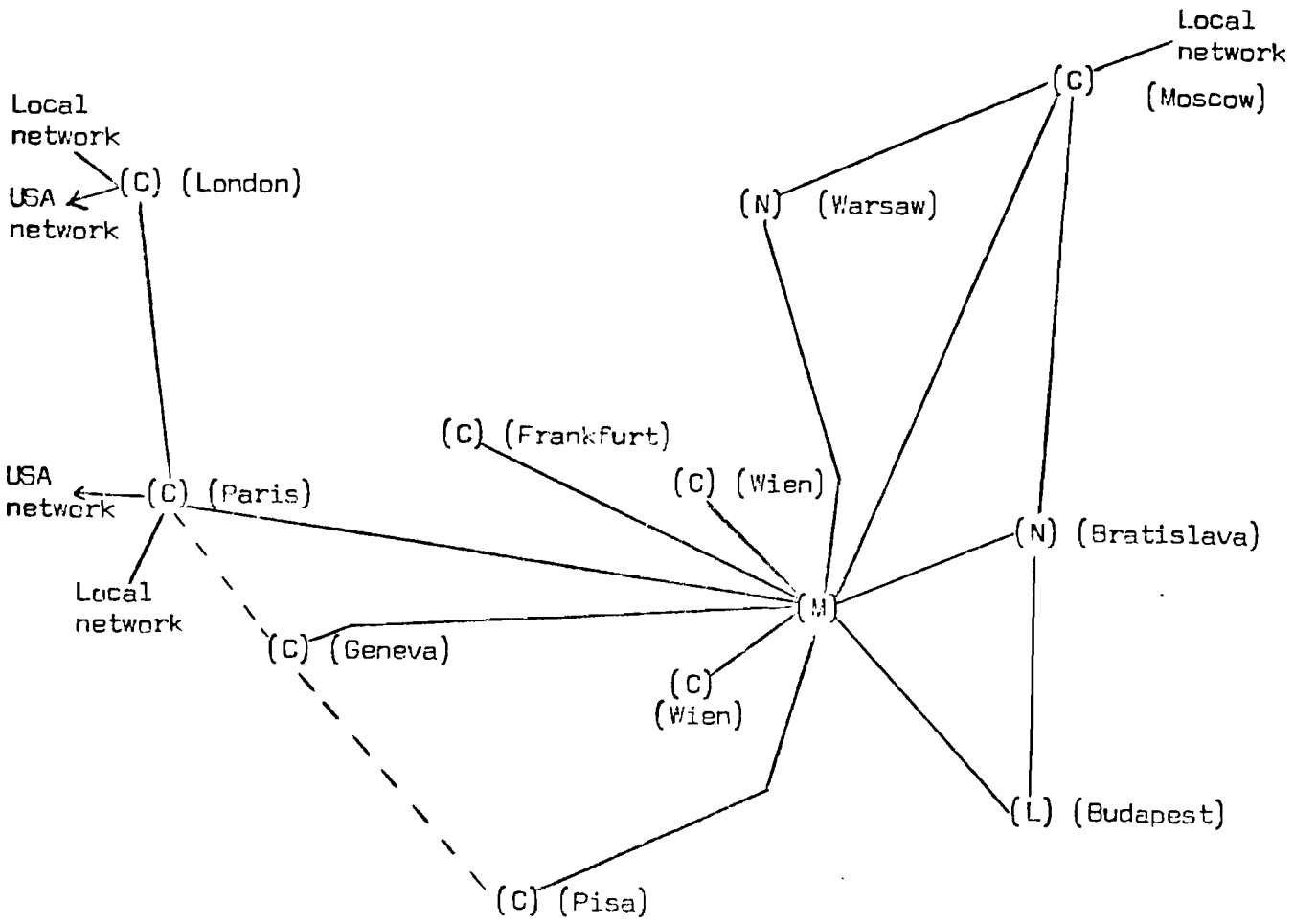

Figure 1. A Hypothetical Network

already have in use, to facilitate transmission modes desired by IIASA. Now consider Paris. This center, let us say, is part of another network and is only loosely connected to the IIASA network. At the same time, it is the only entre to London and the networks in Great Britain. Clearly, it will take a set of well-thought-out and carefully negotiated agreements to permit the kind of flexibility desired. Both policy and financial matters will become important, on top

of the not insignificant technical problems. Since the technical planning will involve a substantial effort, and since the feasibility of a technical approach will depend on the agreements which are possible, it is clear that a very astute planning schedule is required.

System Style and Data Management Conventions

Assuming that workable transmission lines are available and network arrangements are complete in the sense discussed above, one is now faced with logging-in to a variety of computers and transmitting meaningful character strings to initiate some computing task. Each computer system has an established set of conventions (higher-level protocols) which must be honored in order to get useful work done. When working consistently with one system, one either becomes familiar with the pertinent style and conventions or arranges for the necessary assistance. The latter may take the form of human aides or, now more likely, standardized packets which can be simply invoked (for example, EXEC files or catalogued procedures). If one must switch between systems (either hardware or software), he is likely to make mistakes which grow combinatorially with the number of systems used. In any event, most users probably do not understand the reasons behind the conventions and protocols but simply accept them as necessary and use them by rote. It seems clear that network standards must be defined if it is to be usable by many researchers. Now, however, the network must provide translations between its standards and those of the systems.

To appreciate the seriousness of this problem, it may be instructive to briefly examine the job control language (JCL) of OS/360, the IBM 360 operating system. Although many users consider this the worst example of a control language extant, it is widely used and has a very extensive repertoire of capabilities. Aside from the (ugh) syntax, many features of the language and its use are dictated by

system design philosphy. One key tenet is that all storage space and data sets of any kind to be used on a job shall be identified and described in the job input stream (essentially a deck of punched cards, no matter how camouflaged in the literature). There are advantages and disadvantages to this style of data management and we need not enter into a controversy about it. It exists in a widely used system and there are different approaches in other widely used systems.

The JCL statements contain various kinds of information and some of it is in excruciating detail. Both the terminology and the syntax represent extreme examples of "computerese", usually abhorrent to more theoretically inclined people (who have their own strange jargons). This is particularly true of the data definition statements (DD-cards) which define or identify data sets and their device and storage requirements. Since even experts in JCL find it impossible to create perfect input decks for complex jobs consistently--and, in any event, it is tedious--a device known as catalogued procedures (procs for short) is used. These amount to JCL macros with substitutable arguments having, in some cases, default values. Such procs must be created with great care for every class of job and then installed in the local system library. The latter process usually requires special administrative arrangements. Even such an apparently simple and standard function as calling the system assembler requires numerous JCL statements, including DD-cards for scratch files of no interest to the user. Hence procs are nearly always provided for this purpose. (In some interactive systems, a prompting program for the same purpose is made available as an extra-charge "program product.")

Every data set in use has up to three names associated with it, sometimes more internally. (The fact that these are sometimes made to appear identical does not change the logical differences.) First, there is the name of the

data set as it has been or will be stored. These names consist of concatenated symbols which amount to a path into a tree-structured file system. Associated with such a name is a description of the general organization into which it falls, its storage requirements, status, and so forth. These names and descriptions can also be catalogued by the system, so that once they are defined and exist, the data set name provides a sufficient description. However, status and disposition arguments must still be specified for each use. These are actually associated with the second name.

The second name is the DD-name which is limited to 8 characters. It is the referent to the data set for use in the current job. These are often standardized for particular applications but alternates may have to be permitted.

The nature of the third name will vary by application but usually it is the general file name incorporated in the application software. For example, an MP system may require an input file, a model file, and a work matrix, among others. Internally, they might be called INPUT, MODELS, MATRIX, respectively. Since MATRIX is a scratch file, its DD-name will probably be MATRIX also and its data set will be described in detail in a proc, since it is not retained. The MODELS file might also have a default DD-name of MODELS but provision must be made for the user to issue commands which switch internal file MODELS to some other DD-name. Similarly, there may be a default data set description but the user may wish to associate DD-names with different data sets belonging to his project. Since the INPUT file(s) is (are) unique to the job, even more flexibility may be required for this internal name.

Now suppose a user has been running on a different system with a different style of data management. One day

it is decided to switch his job to an OS/360 system, for one of several easily-imagined reasons. Several difficulties can arise. We will indicate just a few.

1. If the user has files (data sets) at his usual center (A) which are required for today's job at center (B), the job cannot be switched.
2. If he has submitted a job deck in the style of (A), it makes no sense to the system at (B).
3. If users submit jobs in network formats, then one of three provisions must exist:
   a) The master node must have translation routines to all system styles to be used for each class of job, and each user must funnel his requests through the master node;
   b) Each node must have translation routines as above;
   c) Each system must have translation routines for the network formats.

   The last would appear the most practical except that agreement by all participating centers to do this is probably impossible to obtain.
4. Even if the difficulties in 3. are resolved, then one of the following provisions must exist:
   a) Prerequisite procs or their counterparts must have already been installed at all centers which are genuine alternates;
   b) The submitting node (or master node) must transmit complete JCL decks or their counterparts (very complicated to create and large to transmit).

The above difficulties apply to batch jobs. Interactive systems, by nature, would be stopped by 1. above unless a common data storage were maintained by the network. This appears technically infeasible and/or prohibitively expensive.

Some, but not all, of the above difficulties would be alleviated if every center had the same computing system. However, this is antithetical to the main concept.

Consequently, it appears that, for the foreseeable future, decisions as to which system to use for which job must be made beforehand using human judgement. Possibly some self-contained and relatively simple tasks, such as compilation and execution of a FORTRAN program, can be switched on a last-moment basis.

Data Input Formats; File Structures and Organizations

The problems in this class are similar to those in the preceding class--but one level higher--and need not be discussed at such length. However, they can be severe and must be reckoned with.

One of the great deficiencies in the world of computing is that input formats for entirely similar purposes have not been standardized across systems. There are, in fact, some de facto standards and strong similarities among systems. However, deviations creep in which are not universally honored. It often happens that the least comprehensive version becomes the most nearly universal standard. (There are several reasons for this which would be tedious to recite.) Consequently, the most advanced systems are most likely to deviate from any basic standard and also among themselves.

The deviations are of two kinds: syntactical and logical. Syntactical variances can usually be resolved by a relatively simple (though often messy) conversion program. However, one still has the problem of getting such converters installed in the right places. The logical differences, on the other hand, usually lead to impasses. Another example from math programming will illustrate this. One very useful feature in the model structures of the SESAME system (and certain others) is what are called indirect coefficients. These are essentially symbolic referents, used for coefficients and

elements, which are evaluated at some time or times during execution of the algorithmic procedures. (They are somewhat like substitutable arguments but more dynamic in use.) However, the most widely used systems have no such provision. Even though SESAME has a standard procedure for converting its model files to formats acceptable by widely used systems, the indirect coefficients (and certain other model component configurations) have no logical equivalent. Conversely, MPSX has certain additions to the "standard" MPS/360 formats which SESAME cannot interpret.

Even when agreed-upon standards are used, different systems do not always give identical results. Anyone who has transported programs written in basic FORTRAN from system to system is aware of this difficulty.

Besides external formats, internal file organization, which is superimposed on the basic data management schemes, can be a further source of difficulty. One system for example, may use tree-structured files and permit random revision or replacement of components. Another similar system may use sequential files in which any change invalidates any old information recorded beyond the point of change. Users adapt to these different philosophies and construct their command and control sequences accordingly. But it is quite a different matter to attempt to intercept the user's commands and convert them to a safe and meaningful sequence for a different system.

Processor (Software) and Data-Base Compatibility

The difficulties in this class are mainly extensions of those in the preceding two classes and have already been hinted at. However, they deserve some additional comment. Just as there is a lack of standardization in data management style, and data format content and syntax, so there is also a lack of agreement on what a particular application system should properly provide. An example from assembly language macros illustrates this. (More pertinent examples for this class exist but would require much more preliminary explanation.)

A macro is a prespecified sequence of prototype instructions which permit substitution of character strings (macro arguments) and conditional or optional inclusion of parts of the sequence. Macro processors have developed into very powerful but highly intricate programming tools. A macro is given a name and it is invoked by using this name as a command (verb) followed by the values to be substituted. The question arises as to whether macros can be nested, that is, whether a macro can include, in its sequence, the invocation of another macro. Modern macro processors permit this as it greatly extends the power of macro usage.

Now a macro prototype must also be defined before the macro can be invoked. A further question arises as to whether macro definitions can also be nested. Both specifying and processing nested macro definitions can become extremely complicated. It is really worth the cost? This is a matter of opinion: UNIVAC permits it, IBM does not. Personally, I would probably not use nested definitions even when available. (I have found the IBM processor quite adequate.) But suppose a programmer used to UNIVAC software finds nested definitions useful. How can these be unscrambled for conversion to an IBM program? (Such a conversion is unlikely, but the concept illustrated is valid.)

This kind of incompatibility also extends to data-base systems. Much of the information content of a system of files is not in the specific data recorded, but in its organization and in the scope of ability to record and extract data by various classifications. Hence, if it is claimed that one data-base system contains the same data as another one, this may be only a half-truth. One must compare the organization of files and the power and flexibility of the processing routines as well.

There are further difficulties connected with data bases which span several of our classes of problems. They may as well be mentioned here.

Data bases tend to be specialized and large ones are unlikely to be duplicated. The creation and maintenance of a large system of files is in itself a sizeable and continuing project. Suppose a valuable data base D has been created at center A using system X. A user at center B, using system Y, wishes to extract information from it. The following kinds of problems can arise:

1. System Y cannot access D directly but must submit procedures to X which cause the required data to be made available. If the user wishes to browse through D to find something, he will probably have to abandon Y and access X remotely;
2. When the correct instructions to X are known, X must output information from D in a form interpretable by Y. This data must be "put somewhere" until Y is ready to accept it;
3. Center A may not permit unlimited access to D. Any restrictions must be understood and honored by B (and the network) or else no data may be obtained. Even an attempt to access restricted data, though inadvertent, may lead to disagreement;
4. System X may be in the process of updating D when a request is received. Presumably X will have appropriate interlocks but this may disrupt scheduling at B;
5. If B can submit new information for D, all the above problems exist in reverse;
6. If B needs information from several data bases, these problems are compounded.

Classification of Jobs and Tables of System Efficiency

The difficulties in this class are mainly conceptual and expensive to resolve. Although we have already seen that it may be impossible to fully automate on-line allocation of jobs to resources, it is clear that an overall classification of jobs with recommended allocations is necessary. The

accounting system must also be planned and implemented. A great deal of study and experimentation will be required, followed by the preparation of excellent manuals and guidelines, some in computer retrievable form.

The difficulty of classifying jobs and evaluating system efficiency is increased by the following facts:

1. Apparently similar tasks may vary widely, in reality, according to size, complexity and mode of operation;
2. Apparently similar capabilities among systems may, in fact, not be truly comparable;
3. The number of cases that can be run for any one class under reasonably constant conditions is insufficient to give a good statistical sample;
4. The types and mix of jobs are continually changing.

Command and Control Languages

The prior discussions have already indicated incompatibilities and conflicts in existing command and control languages, at multiple levels. Some degree of standardization for the network appears necessary. In addition to the real conceptual, technical and pedagogical difficulties (which are by no means insurmountable), this area is rife with strong differences of opinion and/or vested interests. (Possibly, multiple natural languages may add to this. For example, an American computer in Germany or Austria has German markings on the lights and buttons, French in France, etc.)

Looked at from the positive point-of-view, however, this may be a golden opportunity to begin a universal process of standardization. It will be much better to start at a high command and control level and gradually force compatibility downward, than to attempt to patch together multiple styles from the bottom up (except as necessary in the early stages). Coordination and cooperation with existing or planned European networks should be an important part of this effort.

Responsive and Effective Maintenance Procedures

The political problems come clearly to the forefront in this class, both national and organizational. First, it is not certain that adequate maintenance staff and procedures really exist at all places which would be nodes. This needs investigation but let us assume for now that they do. This still leaves questions of commitment, responsiveness, cooperation and authority. Consider again the hypothetical network of Figure 1. Suppose IIASA is working on an important project with London but there are intermittent failures in the connections at Paris. Suppose further that maintenance personnel in Paris are installing new equipment for another network or system and are not much inclined to leave that to search for some bothersome failure in the equipment used by IIASA. (They can easily plead that they have strict instructions to install the new equipment as quickly as possible.) Where and how does IIASA bring pressure to bear to get the necessary repairs made?

Since everything rests on the performance of telephone lines, situations can arise in which IIASA is further removed from a point of trouble. Who does one call if there is a breakdown of network circuits across the CSSR-Poland boundary or across the English Channel?

When a system falls below some level of performance, it is to all intents and purposes unusable. One cannot sit for long at a keyboard if each line takes ten or fifteen seconds to respond. (I have tried.) Somehow the importance of good maintenance must be translated into responsive commitment by a variety of organizations.

Security and System Integrity, plus Accounting Procedures

One of the most important functions of any computerized system is record-keeping. (Indeed, apart from calculations, this is the only function.) Clearly, an accounting system is needed to keep track of usage and charges. This is true even

though bills from individual centers may be prepared, submitted and paid separately, although ideally one would like this to be integrated into network accounting. However, accounting, per se, is only one sort of record-keeping that is required.

A second important aspect of this function is the security and integrity of files. This, in turn, has two sides: safety from interference or confusion within the network, and safeguards against unauthorized access to files both within the network and by others outside. We will discuss the former first.

In both batch systems and interactive systems, user files--whether procedures, programs, or data--are stored in hierarchical files. The primary key into a user's files is a user identification (userid). In interactive systems, both a userid and more-or-less secret password are required and, in both batch and interactive systems, an additional coded account number may be required. Catalogued procedures in batch systems (procs) are usually available only to jobs running under the userid, except for system-supplied procs. Installation of procs requires administrative arrangement so they are fairly safe from inadvertent or unauthorized change.

The userid's may be in hierarchical arrangements themselves. Files whose integrity must be maintained usually require special keys (batch) or special userid's (interactive) to effect any change. However, such files commonly have a read-only status with respect to other users in the same main branch of the hierarchy (same company, same major department, same project, etc.). Even this may be blocked to all but the owning userid or some special class of userid's. These mechanisms have been in operation for several years and there is little chance of inadvertent change to or loss of files except by the owning userid. Additionally, complete system dumps are taken periodically

(once or twice a week) to provide backup in case of system failure. This does create another possible source of exposure of sensitive data but such backup files are usually carefully guarded by center management.

Consequently, within one center complex, proven procedures have long been in use to ensure the integrity of files and to largely forestall any confusion in the manipulation of large amounts of data of many kinds belonging to many users. However, a user-oriented network introduces a need for some additional safeguards. For example, the network probably has a list of valid userid's itself. Care must be taken that these are unique and that errors do not occur in associating jobs with users with systems, etc. This is only an upward extension of conventions already well thought out and in use, but the necessary planning and implementation must be done.

As to unauthorized access to files, the situation is less clear. Administrative discipline and the inherent difficulty of circumventing the provisions existing for integrity of the system will prevent access by the idly curious. Additional safeguards and encodings designed by the user can add further protection. Nevertheless, it is an open question whether a skilled and intrepid programmer, with access to a network and intent on accessing a specific piece of data, can be thwarted. It is easy enough to provide traces which show last access to a file (this is standard in many systems) so that the access will not go undetected after the fact, provided someone looks. But absolute prevention of unauthorized access is difficult to guarantee.

It is easy to make a bogeyman out of security and to blow the problem up out of all proportion to reality. First of all, most information is only meaningful and of interest to those legitimately involved with it. Second, the sheer complexity of modern systems make it unlikely that unauthorized persons know what information exists, where it is, or how to interpret it if they got hold of it.

The main thing is that participants in the network understand the provisions that exist, and agree on the necessary administrative and disciplinary procedures. After all, large corporations keep records of all sorts in elaborate computerized systems, some shared with other companies, and seem satisfied with their security and integrity. Governmental agencies computerize confidential and even secret data. Even though this may be on in-house systems, the number of employees is often large so that the risks are about the same as in commercial environments. To those who are still concerned, the only answer is the following: if you have data so sensitive that absolutely no risk of exposure can be allowed, keep it in a vault or on your own private system.

International and Inter-organizational Political Problems

It seems necessary only to mention this class. IIASA already has more experience in this area than this writer can possibly add to. However, a few words may be in order on the kind of effect that can occur to a project or working system when high-level decisions, or a lack thereof, are made without a full understanding of the technical implications.

Suppose a decision is made to undertake a cooperative project and all participating organizations are in agreement. The project must now be planned and the work-load assigned to various groups. It is almost certain that this allocation will be based in part on special expertise previously gained by the various members. First of all, there may be both duplications and deficiencies in existing skills. If a duplication exists between two groups with some amount of professional competition, it can be a delicate matter as to how to assign tasks. If a deficiency exists, it may be because no one has found that area of effort interesting, so another delicate situation arises in assigning that task.

Now consider a case where a group has a unique capability but their work has been done on a highly incompatible system

in which they have a vested interest. It may not be in the best interests of the project to use their existing programs and procedures. Another delicate situation arises.

Now the resolutions of the above hypothetical--but realistic--situations are very likely to be made on purely political grounds, regardless of the operational consequences for integrating a useful system across the network. If the highest-activity procedures are assigned to a group working on the least accessible or least compatible system, then the effectiveness of the whole project will be badly degraded.

Furthermore, considerable foresight must be exercised in anticipating changes in attitudes which may occur during the project, for any of a number of reasons. For example, a decision by a participating center to change systems or organizational structure may have a disasterous effect on ongoing work.

Of course, all these difficulties are merely extensions of those which have arisen in many corporate managements when new systems and structues have been proposed. However, the lack of centralized authority in an inter-organizational and international framework makes the resolution of such problems more difficult.