Center for Advanced Computation

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

CAC Document Number 161
JTSA Document Number 5508

Research in
Network Data Management and
Resource Sharing

Technology Summary

May 19, 1975

CAC Document Number 161
JTSA Document Number 5508

Research in
Network Data Management and
Resource Sharing

Technology Summary

by

Geneva G. Belford

Prepared for the
Joint Technical Support Activity
of the
Defense Communications Agency
Washington, D.C.

under contract
DCA100-75-C-0021

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

May 21, 1975

Approved for release: Peter A. Alsberg
Peter A. Alsberg, Principal Investigator

Table of Contents

Table of Contents (continued)

Introduction

The Center for Advanced Computation of the University of linois at Urbana-Champaign is preparing a three year research plan to velop network data management and resource sharing technology for plication in the World Wide Military Command and Control System (WWMCCS) tercomputer network. This work is supported by the Joint Technical pport Activity of the Defense Communications Agency.

As part of the preparation of the research plan, the state of e art in network data management and related technology was surveyed d described in a previous report [3]. This report attempts to assess th short term and long term research needs, and to indicate the chnology interdependencies which must be taken into account in conructing a coherent research plan.

Our conclusions are necessarily tentative. Some lines of search which now look promising may well turn out to be dead ends. An expected breakthrough in a critical area may have important (and now foreseen) implications for the whole future of data management. vertheless, we feel that our assessment is basically sound; few of the eds we identify are likely to be disputed.

This report is divided into two main sections. The first ction is an assessment of research needs. Three broad areas of basic search form a foundation for the development of data management chnology. These areas are 1) better formalisms and definitions, 2) ogram-proving techniques, and 3) measurement and analysis. Improved chniques for distributed data management can be developed before all the basic research problems have been solved. But progress in the basic research areas will be required before optimal distributed data management systems can be designed. The second section is a state-of-the-art summary and follows the same topic organization scheme as in the State-of-the-Art Report [3]. Some technology interdependencies are most conveniently noted in this context.

Summary of Research Needs and Interdependencies

There are three broad areas of research which are basic to real progress in data management. These areas are:

1) better formalisms and definitions, to resolve ambiguities and allow one to make rigorous statements, susceptible to proof, on optimality, correctness, etc.;

2) program-proving techniques, to be used in combination with definitions to demonstrate system correctness, security, etc.;

3) measurement and analysis, to aid the system in detecting problem areas and to form a rational basis for system decision algorithms. These three areas, together with the technologies which depend on them, are discussed briefly below.

Although the ultimate goal might be an optimal (or near optimal) data management system, there are a number of problems (e.g., compression and clustering) which may be attacked immediately and in parallel to yield useful results in the short term.

Better Formalisms and Definitions

Data structure description. One reason why progress has lagged in many areas of data management is that concepts and procedures are often vague and ill defined. Lengthy controversies go on for years as to the best way of doing things without there being any clear definition as to what is to be done, and why - or even what "best" means. In the area of data structures, serious efforts have recently (i.e., within the last five years) been made to remedy this lack. There now exists a number of formalisms for describing logical interconnections between data items (e.g. the relational form). But the very number of alternative formalisms available suggests that the ultimate answer has not yet been found.

Data definition languages. Very closely connected to the problem of formally describing data structures is that of developing a data definition language to specify such structures. Clearly the language must wait upon the formalized description. The ultimate formalism and its accompanying language must pass a severe test. They must make it easy to transfer data from one network site to another and to translate data from one structure to another. File transfer and data translation are two important, high-level facilities whose usefulness will depend to a great extent on how well the underlying definition problems are solved.

Protocol definition. Another area in need of new definition techniques is that of network protocols. Network-specific problems, such as random delays and information loss, militate against simple formalisms (e.g. state diagrams) which are adequate for describing single-site processes. But until some formalism is developed, protocols will remain ambiguous and subject to incorrect interpretation by implementers. Furthermore, real progress in developing resilient protocols cannot be made by identifying difficulties one by one and suggesting ways to fix them on an *ad hoc* basis. Fixing one difficulty may well lead to another. A formally specified protocol would be subject to systematic study, perhaps by automatic, computerized techniques, and resiliency (or lack thereof) could be demonstrated unequivocally.

Program-Proving Techniques

Recently there has been considerable work on the problem of rigorously proving that computer programs do what they are supposed to do. A prerequisite is, of course, that a rigorous formal definition of what the program is supposed to do is available for comparison- and that this definition is correctly stated. Thus again we see the necessity of developing methods for formal definition of the various aspects of network data management.

At present, formal proofs are limited to small programs, and even those must be carefully designed so that the proof techniques are applicable. Nevertheless serious efforts are currently being made to design provably correct operating systems and security kernels. There is little doubt that proof techniques should be (and will be) developed sufficiently to allow the systems designer to verify positively that the system is secure, will operate "correctly", etc. The network protocol designer will similarly be able to verify correctness, as well as resiliency. Rigorous proof techniques will be a welcome replacement for the exhaustive study and testing now needed to provide even a shaky conviction of correctness.

Measurement and Analysis

A better understanding of measurement and analysis techniques is basic to many aspects of distributed data management. In a network, measurement is necessary on a continuing basis to detect malfunctions and monitor traffic flow (in order to control flow and thus avoid traffic jams). On a sporadic basis, measurement has been needed in the evaluation of routing algorithms and communication protocols. Furthermore, if true workload sharing is ever to be a reality, ways must be devised to measure (and predict) site workloads accurately.

As for data management *per se*, efficient, dynamic measurement methods must form the core of any automatic structuring/restructuring system. (See discussion of clustering, below.) In essence, the data management system must be able to monitor its own performance - measure response times, identify difficulties in answering high priority queries rapidly, etc.

Once measurements have identified problems, however, the system must take some sort of action. Decision algorithms are needed. And, in order to develop these algorithms, extensive analysis must be carried out on all phases of data management. Some aspects of data management in particular need of study and analysis, and their interconnections, are included in the discussion below.

Optimal Design of the Overall Data Management System

The technology exists for designing reasonably effective, single-site data mangement systems. As indicated in the state-of-the-art summary, however, further research is desirable for the improvement of single-site techniques and is essential for the development of techniques for handling very large and/or distributed data bases.

Compression techniques, fairly well understood, are necessary for reducing large data bases to manageable size. A modest effort expended in this area could yield a better understanding of the best compression strategies to use in various situations. The compression strategy will depend to some extent on how the data is organized, but prior research into data organization *per se* will not be necessary.

Similarly, a good bit of independent progress may be made in certain other areas. File allocation, for example, can be studied without knowledge of the file content or file organization fine structure. Analysis of simple models based on block sizes, retrieval patterns, and various constraints can (and indeed has) yielded considerable insight. What is most needed now is a set of easily applied guidelines - a very feasible, short-term goal. Clustering is another technique which can b developed further in the context of a simple model, as can integrity.

The real problems arise when one tries to put together the various insights and guidelines that one has developed into an overall data management system which is, one hopes, near optimal and which contains facilities for automatic, dynamic structuring. It is clear that one must have begun with the thorough study of the various individual areas - those noted above as well as, for example, effective means of transforming one data structure into another and guidelines for optimal index construction. Once decision algorithms and guidelines have been worked out for the various subtasks, it should be possible to build them into an overall system. The system itself will then be responsible for, say, the decisions on structuring and indexing. And by appropriate monitoring and automatic application of decision algorithms system performance may be expected to improve.

In short, a systematic attack on a number of areas of data management may be begun immediately and proceed in parallel. Each positive result obtained can be applied fairly rapidly to current data base management. The ultimate goal of consolidating the results into an optimal, automatic, data management system will not be achieved in the near future and indeed is likely to pose many difficulties as yet unforeseen.

Finally, the development of a distributed data management system is predicated on the solution of the various networking problems. Protocols must be effective and resilient. Directories must be convenient to use, so that resources may be readily located. The difficulties with updating database copies must be overcome. And the whole environment must be able to guarantee the user the degree of security that he requires.

Summary of the State of the Art

Overview

For the convenience of the reader, we begin this section with a table commenting on the status of the various technology areas and their value to WWMCCS. The remainder of the section is devoted to more detailed summaries of each area, including notes on technology interdependencies.

Table 1

Technology Area		Comments	Projected Value to WWMCCS of Further Work
DATA MANAGEMENT	Data Structures	Individual structures are well understood. Designers of very large data bases need guidelines for choosing among structures. Little work has been done on automatic structuring and restructuring.	Very Valuable
	Hashing	Basic technology is understood. Work is proceeding on optimum choice of algorithm and on choice of keys to be transformed.	Useful
	Clustering and Partitioning	Work is just beginning.	Very Valuable
	Compression	Basic schemes are understood. Work is needed to automate the optimal choice of compression schemes.	Very Valuable
	Data languages	Design of structured-English query languages is straightforward.	Marginal
	Integrity	Automation of consistency checking is a difficult problem. Little progress has been made.	Valuable
	File Allocation	Optimal algorithms are too expensive to use. Heuristic guidelines are needed.	Very Valuable

Table 1 (continued)

Technology Area	Comments	Projected Value to WWMCCS of Further Work
Communications and and Networks	The technology is well advanced and well funded by several agencies.	Marginal
Resource Allocation and Control	In need of work are the problems of 1) maintaining multiple data bases 2) naming and locating resources 3) protocol definition and resiliency.	Critical
Measurement and Analysis	What to measure is not well understood. Techniques of measurement need more work. Modeling of distributed systems needs much work.	Very Valuable
Network Access Systems and Front-ends	There have been several unsuccessful attempts at developing an access system. More research is needed.	Valuable
Security	The authentication problem appears solvable but is not yet solved. Data security is just around the corner. But the confinement problem may not be solvable.	Critical
User Support	Most problems are straightforward applications of known technology. Intelligent terminals are a research problem.	Very Valuable
Installation and Project Support	Network-wide resource accounting and billing need work.	Valuable
Network Support	Communications support is understood. Performance evaluation and software verification are not.	Valuable

Data Management

Various aspects of computer data management have been under study for fifteen years or more. Much of the technology has, however, evolved gradually and by ad hoc approaches from earlier hand or punched card techniques for handling business data - inventories, payrolls, etc. Genuine technical improvements have tended to be discovered and rediscovered by independent workers who were too busy trying to get a system operating under tight time constraints to read the literature. As Teichroew described the situation in 1971 [11], "rather than having the result of n man-years of effort, we have the result of, say, one man-year of effort, expended n times".

This limited viewpoint has had another unfortunate aspect. Not only has work been duplicated, but there has been a lack of development of truly innovative techniques making effective use of the power of modern computers. Most data management system design has been done in the context of putting together something that will work as quickly as possible and not with the object of designing a system that will make optimum use of the latest technology.

A symptom of this lack of real progress in data management is the swamping of the literature by what are usually termed "concepts" papers - dreary productions which sometimes coin new terminology for old ideas and point out problems, but never go into possible problem solutions in any depth. It must be admitted that sometimes new terminology or a fresh viewpoint can really point the way to innovations. In most cases, however, one fashionable jargon replaces another and no progress is made.

With the recent development of computer networks, new prospects for data management have opened up. Data may be stored at various network sites and retrieved from anywhere in the network. All sites need no longer have their own copies of needed data. File backup copies may be stored at various sites for rapid recovery and continual availability in case the master copy (or the site holding it) disappears. This new challenge to the art of data management has, for the most part, spawned nothing but a new set of concepts papers, pointing out problems but only hinting at solutions. Much needs to be done. But a thorough understanding of current data base technology is required for real future progress.

In this brief summary of existing data management technology, we will be particularly concerned with its relevance to and state of development for application to a network environment. Some areas of the technology are reasonably well developed, at least to the point where they may provisionally be used in a network setting without extensive reworking. Other areas, previously applied in only very limited settings, are clearly in need of work before they may be effectively applied in a network.

Data structures. The basic schemes by which data is organized and accessed in a computer have been extensively studied. The simplest storage scheme is that of a table or linear list of data items and associated attributes. Searching a table of any size is time-consuming; usually directories, indexes, etc., are provided to expedite the process.

The indexes - or sometimes the data itself - may be organized hierarchically into trees or other complex structures for more efficient searching. Such structures and search patterns through them are well understood. (See [8].)

The person wishing to organize a data base will find no lack of structures among which to choose. The problem is that there are no really effective guidelines for making the choice. There are, indeed, arguments for the merits of this or that structure, discussions of optimal structures in one sense or another (and usually on a small scale), but there is little to guide the designer of very large data bases.

The proponents of the relational form (see [6], for example) suggest solving the problem by keeping things simple and organizing the data into a set of normalized tables. This solution has its attractive aspects and may well be a good approach to the basic organization of the data and to simplifying the human interface. For efficient retrieval, however, directories and indexing schemes must still be designed.

Automatic data structuring and restructuring, based on query data, are areas which have just recently begun to be studied. Given that the comparative advantages and disadvantages of various structures are not well understood, it is not surprising that work so far has been in very limited contexts (i.e., small sets of data, limited choice of alternatives, etc.). Nevertheless, as data bases grow larger it is becoming imperative that the power of the computer be invoked to aid the data base designer and to provide for rapid readjustment to changing user needs in situations where retrieval speed is critical. Serious research into simple, effective algorithms for deciding when and how to restructure large data bases for more efficient retrieval is needed immediately.

Hashing. Hashing is a data indexing scheme which replaces a key vs. address list by a computed key-to-address transformation. In general, use of such a scheme saves both retrieval time and storage space, and most data management systems make some use of it.

The classic problem of hashing (essentially that of efficiently obtaining distinct locations for all data) is largely solved. Recently, however, research has begun into the important problems of which (of several available) hashing transformations should be chosen in various situations and of which key (or keys) should be hashed for optimum retrieval efficiency. One sees that this is a part of the larger problem identified in the previous section - the development of guidelines for overall data organization and retrieval system design. Research into optimum (or at least good) hashing implementations will necessarily play a role in any general attack on the problems of organizing large data bases.

Clustering and partitioning. These two terms describe the basic process of grouping records (or data items) into blocks for efficient retrieval. Document retrieval systems have made some use of the idea of forming clusters of documents with similar contents. Extension of the idea to more general data bases (e.g., clustering of records with similar attribute values) is a possibility which has begun to be investigated [10], but which requires further study.

A promising approach which deserves thorough investigation is clustering based on actual retrieval or query patterns [4,7]. That is, items are grouped together which are observed to be usually retrieved together. We have looked into this approach and have invented a scheme (dynamic query clustering) for efficiently and automatically identifying the query patterns [2]. This line of research should lead to effective methods for data clustering and for automatically structuring a data base.

Compression. Compacting data to conserve storage is an old technique. It has, however, only begun to be systematized in the last five years or so. Optimal compression codes based on statistical analysis of the data are well understood. Recently, a system has even been developed which automatically carries out such a statistical analysis and assigns the optimal codes [9]. Implementation of such an automatic compression system could effect enormous savings in such areas as file transmission over a network and storage of backups.

Although most compression schemes assume that decoding is a necessary part of searching the data base, it has been recently pointed out [1] that considerable savings could be made by compressing queries instead and then searching the compressed data base. If the ordering of certain field values is important, then those fields must be compressed by an encoding technique which maintains the desired order. Completely automatic compression systems are not feasible because they do not maintain ordering and other critical attributes of data values. Some human intervention is required.

In summary, although compression schemes are well developed, it is not usually clear which scheme (or combination of schemes) should be used in a given situation. Work needs to be done on devising compression strategies which optimize overall system efficiency.

Data languages. We consider under this heading all languages which in some sense manipulate data. This covers everything from the low-level languages for specification of physical data structure to the high-level languages - which may be very close to natural English - in which the user states his queries. The construction of query languages does not seem to pose any serious problems at the present time. No breakthroughs are needed to design a reasonably convenient language for any particular application. Of course, there are trade-offs to be considered. For example, users might like query languages to be as close as possible to natural English. But the less rigid the language is, the more software is required to interpret the queries. It is not difficult to arrive at a compromise in the form of an easily learned, reasonably powerful, yet carefully structured language.

The investigations into low-level data definition languages and their application to transferring data between sites are likely to be important for distributed data base management. There is some controversy as to whether an appropriate data definition language can really solve the problem. However, it is clear that a common language for describing data format, structure, etc., is a requisite for data exchange. As in the case of query languages, however, the general design of such languages does not seem to pose any serious problems.

Integrity. Integrity refers to the problem of maintaining the accuracy and completeness of the information in a data base. At the present time, the principal technique for identifying loss of integrity is *consistency checking*. This usually means devising a set of *ad hoc* rules, specific to the particular data base, to check the data. This area is in serious need of work to determine whether some more systematic approach is possible.

Once it has been determined that data has been lost or is in error, the usual scheme is to appeal to a *backup* copy for the correct data. Maintenance of backups, particularly in a network environment (or anywhere that concurrent access is possible), poses serious problems. For example, users at different sites may be simultaneously making changes in various copies. It is not easy to maintain consistency under such conditions. Our recent investigation [2] into these problems has led to a proposed solution, but much more work needs to be done to verify its validity and investigate its interactions with other aspects of network data management.

File allocation. Most work on file allocation has been for single-processor systems, where the problem is to allocate the files needed by a program (or programs, in a multiprogramming environment) among the various memory devices available. In a network, the problem broadens to include the possibility of distributing files (including backups) among the various network sites. Algorithms for optimal (or near optimal) allocation have been published, but these are too complicated for regular use on large data bases. It would be worthwhile to develop a set of simple guidelines for allocation in a constrained (i.e., by security and proprietary considerations) network environment.

A striking feature of work to date is that costs are always minimized under time constraints. It would be interesting to turn the problem around and minimize average retrieval time under a cost constraint. Trying to get the best possible (in the sense of efficient retrieval) design for your money might be the practical approach in many situations.

Network and Systems Environment

It is impossible to study a network data management system in a vacuum. It is essential to understand the technology determining the design of other computer systems which interact with data management. These systems include security systems, operating systems, and control systems specific to networking. In addition, it is important to have some understanding of network communications technology. In this section we briefly review the state of the art in these areas.

Communications and networks. Various communication media are under study or in use in computer networks. Of particular current research interest are broadcast techniques, which have the potential to provide inexpensive, rapid, and portable computer communication. Network characteristics (medium, topology, communication line capacities, etc.) enter into distributed data management problems by causing extra problems (e.g. delays) as well as by providing extra facilities. Research into certain network software features, such as routing procedures and traffic (flow) control is badly needed to decrease delays and increase throughput. Enough work has already been done to allow working systems to be designed from empirical considerations and rough guidelines. But further investigation and analysis could lead to methods which provide a considerable improvement in network performance.

Resource allocation and control. The allocation and control problems of most interest to us are those critical to distributed computing in a network. One of these problems - the maintenance of multiple database copies - has already been discussed in the Integrity paragraph. As was pointed out there, this problem has only begun to be seriously studied. At the present level of understanding, the best solutions to be expected are specific ones that work in a useful percentage of the cases encountered, or even just in some particularly important case.

Other key problems are those of naming and locating resources. Processes, files, etc., must be assigned unique names in the network. Various methods have been suggested for accomplishing this, but the problem deserves a hard look in the overall data management system context. In order to locate resources, directories (or catalogs) are required. Directories are in a sense just heavily used files. All of the usual problems - where to put the file (including copies), whether it should be distributed, how it should be backed up, etc. - become particularly crucial if resources are to be located rapidly.

The possibility of concurrent user access to a data base requires the development of other sets of controls. Locks and other blocking mechanisms are used to keep users from interferring with one another or making conflicting updates which leave the data in an inconsistent state. But such mechanisms lead to another problem - that of deadlock, or the permanent blocking of a process. Deadlock is a fairly well understood subject in computer systems. Techniques have been developed for deadlock prevention, detection, and recovery. But work on extending these techniques to distributed environments is only in the preliminary stages [5].

Protocols provide the basic languages and procedures by which computers in a network communicate with one another. The ARPA network has furnished a testing ground for the development of a set of protocols that work reasonably well in a benign environment. The construction of basic facilities for host-host communication, Telnet (remote terminal access), file transfer, and remote job entry is a fairly straightforward application of current technology. For a production environment, attention must be paid to security and to protocol resiliency - i.e., the protocol must assume a hostile environment where communications, hosts, and software may fail and include procedures to recover from these and other errors. Looking only at the ARPANET file transfer protocol, we have compiled a lengthy list of trouble spots and suggested design changes [2]. On the basis of this study we conclude that the design of resilient protocols is clearly feasible. However, resiliency should be taken into account from the start; making existing protocols resilient is costly and inefficient. Other potentially useful network facilities - for example, workload sharing -require considerable study before an effective basic protocol may be designed. Finally, there is much current interest in defining a clear, unambiguous method for specifying protocols. It does no good to design good protocols if their descriptions are such that incompatible interpretations may be made by the implementers at the various network sites.

Measurement and analysis. In order that computer system design be done systematically, it is important that measurement and analysis techniques be utilized. As has been indicated often in this report, the system designer has many alternatives to choose among. Measurement of various aspects of system performance can be an invaluable aid to identifying poor decisions and suggesting better ones. An accumulation of measurements of the behavior of existing systems under various conditions can help choose system alternatives for particular applications. The problem of measurement is not generally one of feasibility, but of deciding what should be measured and how the measurements should be interpreted. For example, what aspects of network performance does a distributed data management system need to monitor?

What aspects of its own performance (e.g. retrieval time) should it monitor? And finally, what implications do these measurements have for other aspects of system performance? Should some action be taken? These difficult questions on the role of measurement in data management systems have yet to be studied in any depth.

A good bit of work has been done on system analysis. Most research has involved the development of system models. These are then used for prediction of system behavior, identification of potential trouble spots, study of design alternatives (without having to build the system), etc. Although a fair amount of modeling technology exists, most models are still somewhat primitive and have been built to study only a small subset of the interlocking problems involved in, say, distributed computing.

Measurement and analysis are problems which need to be studied together. Good system measurements are required to provide valid parameters for model analysis. On the other hand, analysis can show which measurements are essential for the understanding and improvement of real system behavior.

Network access systems and front-ends. The last few years have seen a growing interest in two related developments: the front-end and the network access system. The front-end is essentially a mini-computer connected both to the network and to a larger general purpose system (or host). The network access system is again a mini-computer, but one providing network access to terminals at sites without local hosts. Up to the present time there has been only limited experience with both of these devices. Many design problems need to be studied in depth. For example, it is unclear what facilities a network access system should be capable of providing for its users. There is also controversy over whether certain network protocols should be implemented in the front-end or in the host. This controversy is inextricably bound up with the whole problem of developing a better understanding of high-level protocols; i.e., of how they should be designed and implemented. And, of course, effective network access is a necessary prerequisite to successful distributed computing.

Security. Security is a pervasive consideration in the development of a resource sharing system. The overall security of a system requires the secure functioning of the support personnel, the hardware, the system software, and the application software.

In a network, the _authentication problem_ (verification of a remote user's identity) must be solved. No present authentication method is reliable, but there is some hope that current research into physiologically-based, continuous authentication may prove fruitful in the reasonably near future.

Data security deals with the problem of assuring that no user can gain access to any data items without explicit permission. This problem has been intensively studied for about ten years and is well understood. It is probable that some certifiable, data-secure systems will become operational in the near future.

Work on the _confinement_ problem (the problem of _guaranteeing_ that one process will not transmit information to another without specific authorization) is not progressing so well. A solution to the confinement problem may not be possible. However, it may be that the rate of leakage from confinement breaches can be made quite low. If so, it will not be a serious problem as long as cheaper means of breaking security (e.g., compromising personnel) are available.

Network Application Support

Long distances between sites tend to cause problems which either are not present or are less severe at a single site. In this section we briefly survey the state of development of facilities to support network use.

User support. When a user is geographically separated from the computer, the need for good documentation and consulting becomes acute. There seem to be no technological barriers to providing such aids to the user. For example, many sites on the ARPA network provide on-line documentation systems of their services. The difficulty is to keep such a system correct and up to date. Some work has also been done on the design of a network-wide help system. Such a system may eventually become involved with problems of resource naming and location (see previous section).

Good, usable human-to-human communication through the network is also needed. The "mail" facility on the ARPA network has been highly successful in this regard. Security problems, such as mail sender verification (authentication) and control of access to mailboxes, have arisen. Teleconferencing has proved itself to be a highly useful facility and deserves further development. Again, there are no technical difficulties.

The ARPA network community has also found a network information center to be a useful (and perhaps essential) facility. The precise role such a center should play is currently a matter for experimentation and discussion. But any decisions will probably be based more on management considerations than on technology.

A network user is often in need of management facilities to maintain files (including accounting information), to set up conferences, to prepare documents, etc. Such facilities are obviously feasible and indeed are provided by several ARPA network sites.

Finally, the ease and convenience of using the network may be immensely improved by providing the user with an <u>intelligent terminal</u>. (See [2].) The cost of providing each terminal with a built-in minicomputer and floppy disk may seem high at present, but will rapidly decrease over the next few years. Meanwhile, the facilities which such a device can provide may be designed and their cost effectiveness studied. These facilities may include such features as failure detection and recovery aids (including local storage of files), automatic encryption and compression, automatic translation of a simple user-oriented language into languages used at various sites, and sophisticated graphics and network help systems.

<u>Installation and project support</u>. The management of a network installation or of projects using several network hosts requires novel management tools. The areas of accounting and billing are of prime interest; an application of particular importance is a network bank. There are a number of technical problems involved in maintaining a distributed accounting data base and assuring its security. These problems have been noted earlier in this report.

<u>Network support</u>. Two basic types of functions are required for the support of a network as a whole. These are the maintenance of the communications subnet and network performance evaluation. In the ARPA network, maintenance is successfully handled by a Network Control Center; there seem to be no technology gaps in this area. Performance evaluation does need further study, however. In particular, techniques need to be developed for the evaluation of protocols. This is closely tied into the whole problem of measurement: what should be measured and what do the measurements mean?

References

1. Alsberg, P.A., "Space and Time Savings through Large Data Base Compression and Dynamic Restructuring", to be published in Proc. of the IEEE, Aug. 1975.

2. Alsberg, P.A., Belford, G., et al., "Research in Network Data Management and Resource Sharing: Preliminary Research Study Report", JTSA Doc. No. 5509, CAC Doc. No. 162, Center for Advanced Computation, Univ. of Ill. at Urbana-Champaign, May 19, 1975.

3. Belford, G.G., Bunch, S.R., and Day, J.D., with Alsberg, P.A. et al., "A State-of-the-Art Report on Network Data Management and Related Technology", CAC Doc. No. 150, Center for Advanced Computation, Univ. of Illinois at Urbana-Champaign, April 1, 1975.

4. Casey, R.G., "Design of Tree Structures for Efficient Querying", CACM 16, 1973, pp. 549-556.

5. Chu, W.W., and Ohlmacher, G., "Avoiding Deadlock in Distributed Data Bases", Proc. ACM Nat'l. Symp. 1, 1974, pp. 156-160.

6. Codd, E.F., "Normalized Data Base Structure: A Brief Tutorial", Proc. 1971 ACM-SIGFIDET Workshop, pp. 1-18.

7. Gorenstein, S. and Galati, G., "Data Base Reorganization for a Storage Hierarchy", IBM Research Report RC 5063, October 1974.

8. Knuth, D.E., The Art of Computer Programming. Vol. 3/Sorting and Searching, Addison-Wesley, 1973.

9. McCarthy, J.P., "Automatic File Compression", Intern. Comp. Symp. 1973, A. Gunther, B. Levrat, and H. Lipps, eds., American Elsevier, 1974, pp. 511-516.

10. Rettenmayer, J.W., "File Ordering and Retrieval Cost", Inform. Stor. Retr. 8, 1972, pp. 79-93.

11. Teichroew, D., "An Approach to Research in File Organization", Proc. Symp. on Inform. Stor. Retr., J. Minker and S. Rosenfeld, eds., ACM, 1971, pp. 147-154.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
CAC Document Number 161
JTSA Document Number 5508

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*
Research in Network Data Management and Resource Sharing - Technology Summary

5. TYPE OF REPORT & PERIOD COVERED
Research Report

6. PERFORMING ORG. REPORT NUMBER
CAC #161

7. AUTHOR(s)
G. G. Belford

8. CONTRACT OR GRANT NUMBER(s)
DCA100-75-C-0021

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Joint Technical Support Activity
11440 Isaac Newton Square, North
Reston, Virginia 22090

12. REPORT DATE
May 19, 1975

13. NUMBER OF PAGES
27

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*

15. SECURITY CLASS. *(of this report)*
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*
Copies may be obtained from the
National Technical Information Service
Springfield, Virginia 22151

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*
No restriction on distribution

18. SUPPLEMENTARY NOTES
None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*
Data management
Distributed data management

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The state of the art of distributed data management in a network environment is summarized. Areas in need of further research and technology interdependencies are noted.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

BIBLIOGRAPHIC DATA SHEET

1. Report No.
UIUC-CAC-DN-75-161

2.

3. Recipient's Accession No.

4. Title and Subtitle
Research in Network Data Management and
Resource Sharing - Technology Summary

5. Report Date
May 19, 1975

6.

7. Author(s)
G. G. Belford

8. Performing Organization R
No. CAC #161

9. Performing Organization Name and Address
Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

10. Project/Task/Work Unit

11. Contract/Grant No.
DCA100-75-C-0021

12. Sponsoring Organization Name and Address
Joint Technical Support Activity
11440 Isaac Newton Square, North
Reston, Virginia 22090

13. Type of Report & Period Covered
Research Report

14.

15. Supplementary Notes

16. Abstracts

The state of the art of distributed data management in a network environment is summarized. Areas in need of further research and technology interdependencies are noted.

17. Key Words and Document Analysis. 17a. Descriptors

Data management
Distributed data management

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement
No restriction on distribution
Available from the National Technical Information Service, Springfield, Virginia 22151

19. Security Class (This Report)
UNCLASSIFIED

20. Security Class (This Page)
UNCLASSIFIED

21. No. of P
27

22. Price