ORNL/TM-11797

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

Survey of Data Compression Techniques

Reid Gryder
Kerry Hake

MANAGED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

ORNL/TM-11797

SURVEY OF DATA COMPRESSION TECHNIQUES

Reid Gryder
Kerry Hake

September 1991

Prepared for the
The Assistant Secretary of the U.S. Army
for Research and Development, Acquisition
Office of the Product Manager,
Acquisition Information Management
Fort Belvoir, Virginia
under
Interagency Agreement DOE No. 1405-1351-A1

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
managed by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400

TABLE OF CONTENTS

ACKNOWLEDGEMENTS vii

1. BACKGROUND 1

2. THE OBJECTIVE 3

3. PRIOR WORK 5

4. INFORMATION THEORY AND DATA COMPRESSION 7
   4.1 ERROR DETECTION AND CORRECTION 7
   4.2 CRYPTOGRAPHY 7
   4.3 DATA COMPRESSION 7
   4.4 CHARACTERISTICS OF DATA COMPRESSION METHODS 8
      4.4.1 Data Compression Fidelity 8
      4.4.2 Data Compression Symbols 9
      4.4.3 Data Compression Symbol Tables 9
      4.4.4 Data Compression Efficiency 9
      4.4.5 Data Compression for Special Purposes 10

5. USES OF DATA COMPRESSION 11
   5.1 TECHNOLOGIES BENEFITING FROM DATA COMPRESSION 11
      5.1.1 Data Compression for Communication 11
      5.1.2 Data Compression for Storage 11
   5.2 EXAMPLES OF DATA COMPRESSION USE 12
      5.2.1 Early Uses of Data Compression 12
      5.2.2 Current Examples of Data Compression 13
      5.2.3 Future Prospects of Data Compression 14

6. AIM SYSTEM CONSIDERATIONS 15
   6.1 IMPLEMENTATION OF DATA COMPRESSION 15
      6.1.1 Hardware Data Compression 15
      6.1.2 Software Data Compression 16
   6.2 COMMUNICATION MEDIA AND DATA COMPRESSION METHODS 17
      6.2.1 Data Compression on Dedicated Leased Lines 17
      6.2.2 Data Compression on DSNET1 17
   6.3 OTHER OPTIONS 18
      6.3.1 Reduction of Frequency of Updates 18
      6.3.2 Distribution of Partial Updates 19
      6.3.3 Distributed Data Base Design 19

7. CONCLUSIONS 21

8. RECOMMENDATIONS 23

REFERENCES 25

APPENDIX A 29

APPENDIX B 33

ACRONYMS

AIMS	Acquisition Information Management System
PM-AIM	Project Manager, Acquisition Information Management
AIMNET	Acquisition Information Management Network
AMPMOD	Army Material Plan Modernization
DDN	Defense Data Network
DSNET1	Defense Secure Network, secret level
PMIS	Program Manager Information System
PMSS	Program Manager Support System
MMES	Martin Marietta Energy Systems, Inc.
TRANSFER	An AMPMOD file
LPSA	Logistical Program Support Agency
SIMA	Letter Kenny Army Depot
MSC	Major Subordinate Command
DASD	Direct Access Storage Device
MSS	Mass Storage System
JPEG	Joint Photographic Experts Group (JPEG)
MPEG	Motion Picture Experts Group

ACKNOWLEDGEMENTS

The authors are grateful to the Data Systems Research and Development organization of Martin Marietta Energy Systems for their support during the preparation of this document.

1. BACKGROUND

The Assistant Secretary of the Army for Research, Development, and Acquisition has assigned to the Project Manager, Acquisition Information Management (PM-AIM), program management responsibilities for the Army Acquisition Management system development, including the development of the Acquisition Information Management Network (AIMNET), a secure classified network.

PM-AIM also has oversight of the Army Material Plan Modernization (AMPMOD) secure telecommunications network.

The AIM project plans require assessment of the current capabilities of AIMNET, the AMPMOD Network, the development of plans for transition of these to the Defense Data Network (DDN) Defense Secure Network 1 (DSNET1, where the 1 refers to Secret level), the deployment of an AIM Program Manager Information System (PMIS), a Project Manager–Program Executive Officer AIMNET workstation that will run the Program Manager Support System (PMSS) software, and the development of plans for and analysis of telecommunications and hardware-software system enhancements, including tests of alternative enhancements.

PM-AIM has requested DOE to provide assistance in researching the feasibility of utilizing data compression techniques to improve transfer efficiency over both the dedicated and DDN networks. DOE has tasked Martin Marietta Energy Systems, Inc. (MMES), a prime contractor, to provide the required assistance to PM-AIM. This paper reports the status of Task 2B of the statement of work.

2. THE OBJECTIVE

PM-AIM must provide to customers in a timely fashion information about Army acquisitions. This paper discusses ways that PM-AIM can reduce the volume of data that must be transmitted between sites. Although this paper primarily discusses techniques of data compression, it also briefly discusses other options for meeting the PM-AIM requirements. The options available to PM-AIM, in addition to hardware and software data compression, include less-frequent updates, distribution of partial updates, distributed data base design, and intelligent network design. Any option that enhances the performance of the PM-AIM network is worthy of consideration.

The recommendations of this paper apply to the PM-AIM project in three phases: the current phase, the target phase, and the objective phase. Each recommendation will be identified as (1) appropriate for the current phase, (2) considered for implementation during the target phase, or (3) a feature that should be part of the objective phase of PM-AIM's design. The current phase includes only those measures that can be taken with the installed leased lines. The target phase includes those measures that can be taken in transferring the traffic from the leased lines to the DSNET environment with minimal changes in the current design. The objective phase includes all the things that should be done as a matter of course. The objective phase for PM-AIM appears to be a distributed data base with data for each site stored locally and all sites having access to all data.

3. PRIOR WORK

In *Telecommunication Network Study of the AMPMOD System*, published as ORNL/TM-11249 in September 1989, L. Jung and K. A. Hake addressed the AMPMOD needs for increased data throughput on the current dedicated leased communications lines. In addition to suggesting that additional throughput could be obtained by increasing bandwidth (increasing the line speed, or adding more lines), the authors suggest that the volume of data to be transmitted could be reduced by:

1. sending only pertinent data in partial updates,
2. reducing data transmission volume by hardware, or
3. reducing data transmission volume by software.

That paper also points out that to move the anticipated traffic to DSNET1, operating at the full 56 Kb/s, 37 hours would be required to distribute the single AMPMOD file, TRANSFER, from the Logistical Program Support Agency (LPSA) at Letterkenny Army Depot (now known as SIMA) to the Major Subordinate Commands (MSCs). Thus, use of DSNET for this purpose is clearly impossible without some form of data compression.

4. INFORMATION THEORY AND DATA COMPRESSION

Information theory is the study of efficient coding and its effect on transmission speed and error probability [Ingels]. Three closely related branches of information theory are error detection and correction, cryptography, and data compression.

4.1 ERROR DETECTION AND CORRECTION

Error detection and correction is the branch of information theory that strives to minimize differences between original and resulting data. Although error detection and correction is an important field in its own right, we must assume for this paper that we have an error-free communications channel.

4.2 CRYPTOGRAPHY

Cryptography ensures that communication is private, that a message is from the proper source, and that messages are not altered during transmission. Cryptography provides secure communications channels with as little opportunity as possible for intentional or unintentional disclosure of the information being communicated, traffic analysis, or disruption of service from external sources. To accomplish these objectives, an encrypted message usually contains a minimum of recognizable or redundant patterns.

4.3 DATA COMPRESSION

Data compression minimizes the number of bits required to encode information. Data compression often reduces the bandwidth needed to transmit information or increases storage capacity. Removal of redundancy from encoded data is essential to successful data compression.

4.4 CHARACTERISTICS OF DATA COMPRESSION METHODS

Data compression techniques are categorized by several characteristics. One of the more important categories is the fidelity with which the encoded and decoded data agree with the original data. Methods may be distinguish by the size of the symbols substituted in the message to be compressed. Some writers characterize data compression algorithms by the static or dynamic aspects of the symbol table they use. Others characterize methods by their compression factor, which may be very data dependent, or by the amount of computer time required to compress and decompress the information. Some very interesting techniques have been developed for special data compression purposes.

4.4.1 Data Compression Fidelity

The information content varies with the form of presentation. For instance, if the form of text presentation is visual (as in facsimile), the human reader will be able to understand imperfectly formed characters and interpolate for some missing data. Similarly, in audio and video compression, the information to be preserved by the compression algorithm is limited to the information that can be perceived by a human on the other end of the link. For this reason, data compression techniques for different presentation forms are designed for specific applications. Some common data compression techniques are "lossey," in that the decoded data are not identical to the original data but preserve all "relevant" information.

It is interesting to note that one of the early common uses of data compression was in the CCITT Group III standard for facsimile transmission. This field, which is quite distinct from host-to-host data transfer, has grown rapidly with the availability of inexpensive devices. However, losses are tolerated because facsimiles work with a scanned image (a digitized sample of an analog signal). In facsimile transmission, the source is a document, and data compression works on a scanned image of that document. A single isolated speck on the original scanned page is assumed to be noise and is discarded by the data compression algorithm. In this case, the loss of a single bit from the facsimile data enhances the scanned image.

In AIM data bases, the loss of even a single a single bit cannot be tolerated. Thus, discussion in this paper will be limited to those data compression algorithms that are "lossless." For further discussion of lossey data compression techniques, see the reference list.

4.4.2 Data Compression Symbols

Many authors characterize methods by the length of the symbols an algorithm processes, regardless of whether the algorithm uses variable length symbols in the original data or in the compressed data, or both. Run-length encoding can be accomplished if fixed length symbols are used in both the original and the compressed data. Huffman encoding uses variable length encrypted symbols to represent fixed-length original symbols. Other methods encrypt variable-length original symbols into fixed- (or variable-) length encoded data.

4.4.3 Data Compression Symbol Tables

Another distinguishing feature of data compression methods is the source of the symbol table. Some data compression algorithms operate on a static symbol table, or a fixed dictionary of compression symbols. Because the dictionary is fixed, it need not be transmitted with the data. Such algorithms are dependent on the format and content of the data to be compressed; however, a fixed dictionary is usually optimized for a particular data type, whereas for other types of information the compression ratio is significantly lower.

Some data compression algorithms are relatively independent, and some make two passes at the data. The first pass determines the frequency of the symbols that will be processed and builds a symbol table based on that frequency. The custom symbol table is then transmitted, and the second pass uses the custom symbol table to encode and decode data.

Adaptive data compression methods build a custom symbol table as they compress the data. Such algorithms encode each character based on the frequency of preceding characters in a message. The decoding algorithm builds an identical dynamic table as the information is decompressed. Adaptive methods usually start with a minimal symbol table to bias the algorithm toward the type of data they are expecting.

4.4.4 Data Compression Efficiency

The ratio of the size of the compressed image to the size of the original image is called the compression factor. The compression factor is commonly stated as a percent or ratio. Thus a 50-% compression factor represents a 2-to-1 compression ratio, and an 80% compression factor represents a 5-to-1 compression ratio. The smaller the compressed image, the more efficient the algorithm.

In comparisons of data compression algorithms, the time each algorithm takes to compress the original information and to decompress the image back to the original data is an important measurement. A faster algorithm is more economical. Other factors also are important. The algorithm should not hinder the process. On-the-fly compression between application programs and magnetic storage devices should operate as quickly as the storage devices themselves. Likewise, if a compression algorithm is built into a hardware data-communications component, the algorithms should not prevent the full bandwidth of the communication media from being utilized.

4.4.5 Data Compression for Special Purposes

Many algorithms are designed for special purposes. For instance, in a broadcast environment, one may have huge resources with which to compress information that is to be transmitted from a single point to many destinations. The equipment at the destination may be limited because duplication of expensive equipment at each remote site is not economical. Hankamer discusses an algorithm optimized for such an environment [13].

5. USES OF DATA COMPRESSION

This section discusses some of the more common uses of data compression. The ways in which data compression techniques have been beneficially used in actual applications will be demonstrated.

5.1 TECHNOLOGIES BENEFITING FROM DATA COMPRESSION

Communication and storage technologies have benefitted greatly from data compression methodology.

5.1.1 Data Compression for Communication

Data compression methods are designed to reduce the number of bits required to store or transmit information in the original data and to allow information in the decompressed copy of the data to be re-created.

By reducing the size of a message, the effective bandwidth of the communications channel can be increased. Data compression is usually obtained by substituting a shorter symbol for an original symbol in the message. The symbols may be characters, words, phrases, or any other unit that may be stored in a dictionary of symbols. Data compression efficiency varies from application to application, but compression factors as high as 98% have been reported for specific applications. General purpose techniques can frequently accomplish 50% to 60% compression factors. Lelewer and Hirschberg [20] present a more complete, albeit technical, discussion of data compression techniques.

5.1.2 Data Compression for Storage

The obvious advantage for data storage technology is that small items require less storage space. Thus, the effective storage capacity of any storage medium is increased if the data are compressed. There is, however, with current technology, another important implication of data compression on storage technology.

Storage devices are among the slower components of modern computer technology. Transfer of information to and from magnetic media is relatively slow compared with the speed of internal processing. Thus, many vendors have discovered that by building data compression directly into the writing and reading processes of the storage device, they have not only increased the amount of information that can be stored on the device but have also sped the transfer of the data to and from the device. Thus, data compression makes storage devices seem not only bigger but faster as well.

5.2 EXAMPLES OF DATA COMPRESSION USE

Specific examples of how data compression has been used in the past and how it can be used in the future will be discussed in this section.

5.2.1 Early Uses of Data Compression

One of the earliest forms of data compression was null suppression. In paper tape, a row of holes was punched to represent each character of the data. But if one made a mistake, it was not possible to undo the wrong holes. So one defined a "null" character by punching holes in every possible position for a character on the tape. Null characters could be ignored by any program. Thus, if you made a mistake, you simply went back to that character position on the tape and punched every hole. People soon realized that these null characters represented no information at all and that, even after being loaded into computer memory, the null characters could be discarded without information loss.

In the days of computer cards, most text files were a collection of fixed-length card images. Blank spaces were frequently left at the end of each card to prevent lines from breaking in the middle of words. These blanks represented no information. One of the earliest forms of textual data compression supported variable-length records so the insignificant blanks at the end of each record could be truncated without information loss.

In the early 1980s, microcomputers became widespread; with that growth came the concept of public-domain software and "shareware," which is software that authors share with other users. A network of bulletin board systems flourished, using personal computers as the host, and dial-up telephone lines were used as the data-transfer medium. To share information, microcomputer users used inexpensive (and slow) modems over dial-up circuits. Microcomputer users have long understood the value of data compression. Usually, a person attended the operation on one (or both) end(s) of the line, and the time spent transferring a file was not only bothersome but costly if the file transfer was long distance. Microcomputer users quickly learned that if they could squeeze the blanks out of a file, it would transfer more quickly (and cheaply).

Eventually, more and more sophisticated data compression algorithms were developed for microcomputers. Mainframe data compression applications simply have not kept pace with developments in the microcomputer field.

5.2.2 Current Examples of Data Compression

IBM's Data Facility Data Set Services, available on GSA contract for IBM 4381 hosts, provides optional run-length compression in producing a backup dump of Direct Access Storage Device (DASD) or Mass Storage System (MSS) virtual volumes. Individual data sets may be specified for backup, and the resulting dump data set should be smaller than the original data set; however, the compression factor will probably be smaller than that of a more sophisticated data compression algorithm.

For years, backup-tape devices have had a data compression scheme built into either the hardware or the software. Because these schemes are totally transparent to the user, they frequently were not even mentioned in the product specifications. Both microcomputer and mainframe backup systems commonly have data compression built-in, but the resulting data set usually requires special software and thus does not lend itself to use with data communications.

Similarly, wide area networks (like DDN) have data compression on their internal node-to-node trunk lines, but the customer need not be told. As more and more network customers use their own data compression, the redundancy of the network's input data will be reduced, and the effectiveness of the network's data compression algorithms will decrease. At that time, the network customer who does not compress his or her own data will be paying a premium rate for an inferior service.

Other standards that have become universally adopted are the Microcom Networking Protocol Class 5 and the CCITT V.42bis data compression standard, both of which are found in newer commercially available modems. Products conforming to these standards claim as much as 34K baud throughput from a 9600 baud modem. V.42bis is a lossless data compression algorithm, supplemented by error detection and correction, so the resulting data will most likely be precise duplicates of the original data.

5.2.3 Future Prospects of Data Compression

Multimedia computing is a buzz word of the industry. Multimedia integrate audio and visual information with computer-generated information. For completeness, multimedia must be discussed, even though the techniques are lossey and inappropriate for computer data base technology.

Still-image-compression software products based on the standard proposed by the Joint Photographic Experts Group (JPEG) are readily available now. Hardware chip sets also have been developed, and the adopted standard probably will not deviate from the proposal significantly. All the manufacturers of software and chip sets have promised to support the actual standard, in addition to their current products, when the standard is finally adopted.

One of the most promising areas for data compression is the prospect for transmitting full-motion video over digital circuits. The Motion Picture Experts Group (MPEG) is developing a standard for video compression, with the idea that integrated circuit chips should be able to provide real-time full-motion video without requiring unreasonable amounts of digital data transmission and storage.

6. AIM SYSTEM CONSIDERATIONS

For data compression to be used effectively, the entire system must be understood. A complete analysis of the needs of the system must be made, including all requirements. In AIM's situation, the major requirement is to ensure that classified data bases on widely dispersed systems remain identical by using limited bandwidth data transmission facilities.

6.1 IMPLEMENTATION OF DATA COMPRESSION

Data compression may be implemented in hardware or software. Some systems rely on a combination of hardware and software; these systems implement an algorithm that can be accelerated by a special high performance chip set that performs the repetitive operations of the algorithm.

6.1.1 Hardware Data Compression

Hardware data compression devices that implement data compression on a point-to-point communications link, either for single or multiplexed ports, can be purchased from commercial sources. If such a device were to be used on a dedicated classified link, it would need to be installed between the host computer and the encryption devices at each end of the communications link. Such point-to-point compression devices routinely produce 4-to-1 reductions in the volume of textual data and thus provide an apparent factor of 4 increase in bandwidth.

No known hardware data compression devices will operate with a network unless they are an integral part of the network design. The network requires that each packet of information have a header that contains the address to which the information is to be delivered. If a hardware data compression device is used, it will compress the entire packet, including the address header, and the network will not be able to read the address and will not know where to deliver the packet.

Like the encryption devices in common use, the data compression devices expect to be talking directly to an equivalent device on the other end of the link and thus alter the entire incoming data stream, including any packet addressing or routing information. Unless intermediate network nodes can decompress and interpret the addressing information, they cannot implement network routing functions. Representatives of Aydin Monitor Systems state unequivocally that data compressors cannot be used with AYNAC equipment that the AMPMOD network uses.

Even if hardware data compression could be used between the AMPMOD host computer and the network packet switch node, the network node can process only 56 Kb/s of information. Accelerating the communications link to the network node beyond that limit would require that service be upgraded to another class and would serve no useful purpose.

A set of hardware chips that implement a data compression scheme has recently been developed. InfoChip Systems, Inc., of Santa Clara, California, claims to have developed the first lossless and noiseless data-compression and -decompression co-processor for disk and network controllers. When used as a component in the disk controller hardware, the co-processor allegedly increases total data-storage capacity of personal computers and workstations by 15 times or more. The company also claims that the chips can be used to multiply bandwidths or increase the capacity of any type of local area network. The company claims compression ratios of between 4 : 1 to 15 : 1 for CCITT Group III facsimile documents, of 3 : 1 to 8 : 1 for data bases, of 1.5 : 1 to 7 : 1 for image files, of 2 : 1 to 12 : 1 for CAD/CAM files, of 2 : 1 to 3 : 1 for word processing files, and of 1 : 1 to 2 : 1 for executable code. Using such devices as these, vendors are rapidly developing new products, such as remote bridges with data compression.

Such hardware gives hope that many more inexpensive data compression products—perhaps even an equivalent to the AYNAC with data compression as an integral feature—will reach the market soon. When that happens, the Army Acquisition Information Management Program should be positioned to quickly adapt the new hardware that include data compression to their needs.

6.1.2 Software Data Compression

Software data compression is the execution of a program that reduces the size of a data set in such a way that a decompression program could create another data set identical to the original. The smaller data set could then be communicated between systems, and a copy of the original data set could be reproduced on the receiving system by a decompression program.

Software data compression must be conducted on a machine that can analyze the data and before the data are packeted for the network. As Jung and Hake (1989) observed, no viable software is available for this purpose in the IBM host computer environment. AIM could, of course, develop custom software, but that would take time and money. Algorithms exist and implementations are readily available in the microcomputer arena to obtain a 4-to-1 compression ratio on textual information. Adapting such algorithms to an IBM MVS environment should not be difficult; why someone has not yet done it is very interesting.

In theory, AIMNET software data compression would run a compression program on the originating host computer that would pack the information into a smaller-size file. This smaller file would then be transmitted through the communications link in compressed form, and a decompression program at the receiving host would convert the data back into its original form. This form of software data compression would be totally independent of the communications medium used and should cause no problems for DSNET1.

6.2 COMMUNICATION MEDIA AND DATA COMPRESSION METHODS

6.2.1 Data Compression on Dedicated Leased Lines

The use of data compression on point-to-point leased lines is a simple matter. One can buy hardware boxes that, when inserted pairwise in a point-to-point data stream, can provide apparent increased bandwidth over a communications channel. If additional throughput is required on the current leased lines, it is recommended that PM-AIM purchase a pair of hardware data compression devices for installation at each end of the line and thus avoid the cost of upgrading the line to a higher speed.

Simplex Communications Corporation of Ann Arbor Michigan is one company that offers such products. Appendix B lists other commercially available products that use data compression in some niche of the computer and communications industry.

6.2.2 Data Compression on DSNET1

Opportunities for data compression in conjunction with AMPMOD's or PM-AIM's use of DSNET are limited. For the purpose of this discussion, the network includes the software that provides the destination address and packetizes the information to be sent. The address information on each packet of information must be preserved for the network to route the data to the proper destination. The address information is part of each packet header prepared by the networking software running in the host computer. For the addresses to be preserved, the compression must either take place before execution of the networking software that prepares the packets for the network, or it must be done in such a way that the address and control information is preserved, and only the data are encoded. Neither of these options is possible with existing hardware data compression devices.

Software data compression, running in the host computer before the data are passed to the networking software, could reduce the size of the data and thus provide an apparent increase in throughput over the DSNET. Although such software is common in the microcomputer environment, no commercially available package for the IBM mainframes could be found.

Software packages to support data compression during micro-to-mainframe file transfer (e.g., BLAST), and also micro-based 3270 terminal emulation with data reduction (e.g., Packet/Flash), are readily available; however, these packages are not applicable to the DSNET environment.

6.3 OTHER OPTIONS

Other options for improved performance in the objective phase of PM-AIM implementation include the addition of a network capability to transfer information to multiple addresses simultaneously. Thus, to update the data base, the host could send a message once, with an address list, and the network would deliver the message to each addressee. Currently, the originating host must make a copy of the message for each address in the address list and transmit a copy of the message to the network for each addressee.

6.3.1 Reduction of Frequency of Updates

The amount of data to be communicated between sites could be reduced if the data base were updated less frequently; however, this has detrimental effects on the timeliness of the data available at other sites and is therefore excluded from serious consideration until such time as the requirement for additional bandwidth exceeds the requirement for timeliness of data.

6.3.2 Distribution of Partial Updates

Distributing only pertinent data offers the greatest opportunity for increased efficiency of the telecommunications links in the target phase. However, this option would require substantial changes in the way things are done. Instead of transmitting entire data bases on a regular basis, one need transmit only the changes made since the last transmission. Because the information that is changed in the data base is a small portion of the entire data base, the required data transmission volume could be greatly reduced. This solution is advantageous in that none of the applications packages on the host computers would need to be changed.

However, this solution is not without drawbacks. If the partial update method were adopted, an additional procedure would be needed to verify that the two data bases were identical or to resolve differences if they were not. Otherwise, the data bases would eventually be unsynchronized, and the entire system would fail.

Although reducing the amount of data to be sent is most promising in terms of minimal cost of communications links, personnel are very reluctant to implement this solution; they already have a working system, and the amount of effort and cost required to redesign the system to accommodate partial updates and synchronization checks is unknown. It is imperative that any changes to a vital system as large as AIMNET be well thought out and well tested.

6.3.3 Distributed Data Base Design

The alternative to redesigning AIMS to use distributed data bases, and thus transmit fewer data, is not feasible within the current or target phase and must still be considered an objective. The only choice in the target phase is to implement some form of data compression.

7. CONCLUSIONS

It appears that, although hardware data compression could increase the bandwidth of the existing leased lines, there is no equivalent technique for improving the throughput of DSNET1 unless the network is upgraded to a higher bandwidth. Since this is outside the control of PM-AIM, it is suggested that a long-term effort be initiated to identify the techniques necessary to implement smarter, rather than faster, data communication.

The opportunities to reduce the volume of data to be transmitted are endless, but the best ideas can be obtained from those who know the system best. Potential AIMNET users seem to want to share ideas about how the AIMNET system could be smarter. Many have already suggested how efficiency could be improved but have no authority to request that equivalent changes be made at other locations. An open forum for sharing such ideas would be in the long-term interest of AIM.

Much of the effort required to remedy this situation will be in the education and motivation of the staff to look for least-cost solutions rather than implementing expedient solutions that are known to work. The process of changing the environment in which the existing personnel work cannot be done overnight, and a long-term effort is indicated.

8. RECOMMENDATIONS

AIM should immediately start identifying the minimal portions of their data bases that must be communicated to accomplish a mission. The most rewarding option is the reduction of the amount of data to be sent over DSNET; only the required data should be transmitted, and redundancy should be eliminated at the logical functional level. After having eliminated redundancy at that level, AIM should once again check for existing commercially available data compression programs and should use a commercially available program if one exists. If none exists, AIM should consider implementing a data compression algorithm via custom programming on their host computers.

In the current phase, PM-AIM can explore the effects of a reduced frequency of update and can install hardware data compression devices on dedicated communications links. No options are available for the DSNET1 implementation.

In the target phase, PM-AIM should explore the possibility of reducing the amount of data that needs to be communicated by sorting, eliminating duplicate or replaced transactions, or sending only the changes. A process should be incorporated to ensure that the data bases do not become unsynchronized. The technological developments should be closely monitored for new advancements that would enable new capabilities. The development of a data compression program applicable to PM-AIM's particular situation would be a worthwhile task, if there is no technological breakthrough that makes such a program unnecessary.

In the object phase, PM-AIM should develop a true distributed data base. Transactions should be processed as near their origin a possible, with the results of all transactions visible to all other parties in real time.

REFERENCES

Hake, K. A. (November 1990). "An Examination of Electronic File Transfer Between Host and Microcomputers for the AMPMODNET/AIMNET Classified Network Environment." ORNL/TM-10874.

Jung, L. and Hake, K. A. (September 1989). "Telecommunications Network Study of the AMPMOD System." ORNL/TM11249.

DATA COMPRESSION REFERENCES

[1] Anderson, J.B., and Mohan, S. Feb. 1984. "Sequential Encoding Algorithms: A Survey and Cost Analysis," *IEEE Trans, Commun.* **COM-32**(2) 169-176.

[2] Apostolico, A., and Fraenkel, A. S.March 1987. "Robust Transmission of Unbounded Strings Using Fibonacci Representations," *IEEE Trans. Inf. Theory* **33**(2), 238-245.

[3] Bell, T., Witten, I.H., and Cleary, J.G. Dec. 1989. "Modeling for Text Compression," *ACM Computing Surveys* **21**(4) p. 557(35).

[4] Bellman, R. 1957. *Dynamic Programming*, Princeton University Press, Princeton, NJ.

[5] Choueka, Y., et al. 1986. *Huffman Coding Without Bit-Manipulation*, Tech. Rep. CS86-05, Weizmann Institute of Science, Dept. of Applied Mathematics, Rehovot, Israel.

[6] Connell, J. B. A. July 1973. "Huffman-Shannon-Fano Code," *Proc. IEEE* **61**(7), 1046-1047.

[7] Fischer, T.R., and Dahm, P.F. May 1985. "Variance Estimation and Adaptive Quantization," *IEEE Trans. Inform. Theory* **IT-31**(2), 428-433.

[8] Forney, G.D. Jr. Mar. 1973. "The Viterbi Alogirthm," pp. 268—278 in *Proceedings of the IEEE* **61**(3).

[9] Forney, G.D. Jr., et al. Sept. 1984. "Efficient Modulation for Band-Limited Channels," *IEEE J. Selected Areas Commun.* **SAC-2**(5), 632-647.

[10] Gallager, R. G. Nov. 1978. "Variations on a Theme by Huffman," *IEEE Transactions on Information Theory* **IT-24**(6) 668-674.

[11] Gibson, J.D. Apr. 1980. "Adaptive Prediction in Speech Differential Encoding Systems," pp. 488-525 in *Proceedings of the IEEE* **68**(4).

[12] Gray, R.M. Jan. 1977. "Time-Invariant Trellis Encoding of Ergodic Discrete-time Sources with a Fidelity Criterion," *IEEE Trans. Inform. Theory* **IT-23**(1), 71–83.

[13] Hankamer, M. June 1979. "A Modified Huffman Procedure with Reduced Memory Requirements," *IEEE Trans. Commun.* **27**(6), 930–932.

[14] Huffman, D. A. Sept. 1952. "A Method for the Construction of Minimum-Redundancy Codes," pp. 1098–1101 in *Proceedings of the Institute of Electronic Radio Engineers* **40**(9).

[15] Iyengar, V., and Kabal, P. 1988. "A Low Delay 16 kbits/sec. Speech Coder," pp. 243–246 in *Proceedings of the IEEE International Conference on Acoustics, Speech, Signal Processing*, New York.

[16] Jayant, N.S. Sept. 1973. "Adaptive Quantization with a One-Word Memory," *Bell Syst. Tech. J.* **52**(7), 1119–1144.

[17] Jayant, N.S., and Noll, P. 1984. *Ditial Coding of Waveforms*, Prentice-Hall, Englewood Cliffs, NJ.

[18] Johnson, J.T. June 29, 1990. "The Big Squeeze," *Personal Computing* **14**(6), p. 139(3).

[19] Larmore, L.L., and Hirschberg, D.S. July 1990. "A Fast Algorithm for Optimal Length-Limited Huffman Codes," *J. Assoc. for Computing Machinery* **37**(3), p.464(10).

[20] Lelewer, D.A., and Hirschberg, D.S. Sept. 1987. "Data Compression," *ACM Comput. Surv.* **19**(3), 261–296.

[21] Marcellin, M.W., and Fischer, T.R. Jan. 1990. "Trellis Coded Quantization of Memoryless and Gauss-Markov Sources," *IEEE Trans. Commun.* **Com-38**(1).

[22] Marcellin, M.W., Fischer, T.R., and Gibson, J.D. Jan. 1990. "Predictive Trellis Coded Quantization of Speech," *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-38**(1).

[23] Oppenheim, A.V., and Schafer, R.W. 1975. *Digital Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1975.

[24] Pearlman, W.A. Aug. 1982. "Sliding-Block and Random Source Coding with Constrained Size Reproduction Alphabets," *IEEE Trans. Commun.* **COM-30**(8), 1859–1867.

[25] Pearlman, W.A., and Chekima, A. May 1984. "Source Coding Bounds Using Quantizer Reproduction Levels," *IEEE Trans. Inform. Theory* **IT-30**(3), 559–567.

[26] Regan, S.M. June 1990. "LZW Revisited:Speeding Up an Old Data Compression Favorite," *Dr. Dobbs Journal* **15**(6), p. 126(3).

[27] Reininger, R.C., and Gibson, J.D. Jan. 1985. "Backward Adaptive Lattice and Transversal Predictors in ADPCM," *IEEE Trans. Commun.* **COM-33**(1), 74–82.

[28] Schwartz, E. S., and Kallick, B. Mar. 1964. "Generating a Canonical Prefix Encoding," *Commun.* **ACM** 7(3), 166-169.

[29] Seiter, C. Jan. 1991. "The Big Squeeze," *Macworld*, 128-135.

[30] Sieminski, A. May 1988. "Fast Decoding of the Huffman Codes," *Inf. Process. Lett.* **26**(5), 237-241.

[31] Shannon, C.E. 1948. "A Mathematical Theory of Communication," *Bell Syst. Tech. J.* **27**(3/4), 379-423; 623-656.

[32] Stewart, L.C., Gray, R.M., and Linde, Y. Apr. 1982. "The Design of Trellis Waveform Coders," *IEEE Trans. Commun.* **COM-30**(4), 702-710.

[33] Storer, J. A. 1988. *Data Compression Methods and Theory*, Computer Science Press, Rockville, MD.

[34] Tanaka, H. Jan. 1987. "Data Structure of Huffman Codes and Its Application to Efficient Encoding and Decoding," *IEEE Trans. Inf. Theory* **33**(1), 154-156.

[35] Ungerboeck, G. Jan. 1982. "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. Inform. Theory* **IT-28**(1), 55-67.

[36] Viterbi, A.J., and Omura, J.K. May 1974. "Trellis Encoding of Memoryless Discrete-Time Sources with a Fidelity Criterion," *IEEE Trans. Inform. Theory* **IT-20**(3), 325-332.

[37] Warner, W.C., June 21, 1990. "Compression Algorithms Reduce Digitized Images to Manageable Size," *EDN* **35**(13), p. 203(8).

[38] Wilson, S.G., and Lytle, D.W. May 1982. "Trellis Encoding of Continuous-Amplitude Memoryless Sources," *IEEE Trans. Inform. Theory* **IT-23**(3), 211-226.

[39] Witten, I. H., Neal, R. M. , and Cleary, J. G. June 1987. "Arithmetic Coding for Data Compression," *Communications of ACM* **30**(6), 520-540.

APPENDIX A

DATA COMPRESSION TECHNIQUES

A brief glossary of data compression methods and terminology follows this appendix. For more details of compression methods, see the references.

Run Length Encoding. Converts any number of repeated symbols into a count plus one occurrence of the symbol.

Variable Length Codes. Substitutes codes of varying length for characters in text.

Huffman Coding. Used for a given character distribution, by assigning short codes to frequently occurring characters and longer codes to infrequently occurring characters.

Static Huffman Coding. Uses a fixed table based on a representative set of data.

Dynamic Huffman Coding. Reads the data twice, once to determine the frequency of occurrence of characters and to build the table, and once to encode the information. Huffman's minimum redundancy encoding minimizes the average number of bytes required to represent fixed length characters in a text.

Adaptive Huffman Coding. Encodes each character on the basis of the frequency of preceding characters in a message.

Variable Length Encoding. Looks for repeated words and phrases as well as characters.

Lempel-Zif-Welch. Replaces frequently used symbol strings with fixed length codes.

Data Dependency. Takes advantage of statistical information about the relationships between characters (e.g., "q" is usually followed by "u").

APPENDIX B

COMMERCIALLY AVAILABLE DATA COMPRESSION PRODUCTS

IBM MVS

Compact/LZW
High-speed data compression of records, groups of records, or entire files.
Prime Factors, Inc.
1470 E. 20th Ave.
Eugene, OR 97403
(503) 345-4334

Data Compression/Expansion
Subroutines for user programs. Compresses alphanumeric fields by as much as 25%.
International Business Machines Corp. (IBM)
Old Orchard Rd.
Armonk, NY 10504

DFDSS
IBM Data Facility Data Set Services (DFDSS) provides the ability to use run-length compression in creating a DUMP data set to backup DASD or MSS virtual volumes. Individual data sets may be specified for backup.
International Business Machines Corp. (IBM)
Old Orchard Rd.
Armonk, NY 10504

SHRINK/MVS
Stand-alone, or integrated into Sterling Software's Storage Automation Management System (SAMS), this product allows the user a choice of compression ratios (from 50 to 80 percent) in a VSAM exit routine.
Sterling Software
Systems Software Marketing Division
11050 White Rock Road, Suite 100
Rancho Cordova, CA 95670-6095
(916) 635-5535

IBM PC

PKPAK v3.6
A shareware product previously known as PKARC
PKZip 1.02, available for a minimum $25 single-user fee; users who want documentation can pay $47 or more.
PKWare Inc.
7545 N. Port Washington Rd.
Glendale, WI 53217
(414) 352-3670

ARC+Plus version 7.0 for DOS and OS/2
Features enhanced data compression priced at $89.95.
System Enhancement Associates
(201) 473-5153

Newspace
File compression program priced at $69.95.
Isogon Corp.

Fontspace
File compression program priced at $89.95.
Isogon Corp.

Cubit
File compression program priced at $69.95.
Softlogic Solutions

SQUISH Plus v1.05
File compression utility uses data compression primarily for access to the fixed disk; priced at $99.95.
Sundog Software Corp.

SQZ Plus
File compression program; priced at $99.95.
Symantec Corp.

Pak 2.10
Utility program.
NoGate Consulting

Pack/File 2.33
Utility program priced at $24.95.
Shortcut Software (products)

ARC 6.02
Utility program priced at $50.
System Enhancement Associates, Inc.

LZEXE
Compresses .exe files, a public domain utility program.
Fabric Bellard

LHarc 1.13
Public domain product, LHARC employs adaptive Huffman coding with an LZ encoder to achieve a high rate of compression with a faster decompression process.

ICE

Zoo 2.01

PC TOOLS Deluxe v5.5
Includes option to compress files and subdirectory structures into BACKUP.nnn files.
Central Point Software

Norton Backup 2.0
Comes with data compression software.
Norton

FASTBACK PLUS v2.09
Will compress files and subdirectory structures into backup files.

COREfast v2.00
Provides backup to COREFAST.nnn files with data compression.

CLIP + v4.1
Backs up and restores programs, copies files to the backup media as discrete files, retaining the original directory structure. Text files will be compressed using an algorithm known as E40 (it compresses English to about 40% of its starting size).

Stacker
Data compression software utilities. Priced at $129.
Model 9703 and Model 9704 Data Compression Coprocessors implement the QIC-122 standardized algorithm for data compression and expansion for tape backup. The hardware card (with a 9703 coprocessor) sells for $229.
Stac Electronics
Carlsbad, CA.
(800) 522-7822

3-1/2 inch backup tape drive with data compression
Colorado Memory

Expanz! Coprocessor Board
$199 each—half-height data-compression card for PCs can triple the memory capacity of any storage device because it's designed to handle block, serial and packet transfers. Expanz! compresses data as they are written to a disk and decompresses them as the disk is read.
InfoChip Systems, Inc.
Santa Clara, CA
(408) 727-0514

MACINTOSH DATA COMPRESSION UTILITIES

Compactor
$25 shareware.
Bill Goodman
109 Davis Ave.
Brookline, MA 02146
(617) 731-4802

StuffIt Classic 1.6
$25 shareware.
Aladdin Systems Inc.
Deer Park Center, Suite 23A-171
Aptos, CA 95003
(408) 685-9175

StuffIt Deluxe
File compression program, $99.95.
Aladdin Systems Inc.
Deer Park Center, Suite 23A-171
Aptos, CA 95003
(408) 685-9175

Disk Doubler
$79 (file compression software).
Salient Software, Inc.
3101 Avalon Court
Palo Alto, CA 94306
(415)-852-9567 or
(800) 326-0092

Diamond
$125, data compression software.
Sextant Corp.

Ecomp
(Software implementation of JPEG.)
Electronics for Imaging
950 Elm Ave., Suite 195
San Bruno, CA 94066
(415) 742-3400

Compression Workshop
$550, software version of the JPEG compression algorithm fo Mac II.
C-Cube Microsystems, Inc.
399-A W. Trimble Road
San Jose, CA 95131
(408) 944-6300

MACINTOSH HARDWARE

CER NuBus Board
$995; scanned image compression.
Hurdler

Colorsqueeze
$179; color image compression.
Eastman Kodak Co.
343 State St.
Rochester, NY 14650
(800) 445-6325, Ext. 110
(800) 223-1650

PicturePress
Software, $199.
PicturePress Accelerator
Hardware $999—NuBus coprocessor board.
ImagePress Compression Software
$99; compress continuous-tone color images without visible loss of image data.
Storm Technology, Inc.
220 California Ave., Suite 101
Palo Alto, CA 94306
(415) 322-0506

ArcImage
NuBus Board and software; $1895; document scanning, compression, and storage.
First Financial Technology

Compression Master
$995—video compression NuBus board with CL550 chip.
C-Cube Microsystems, Inc.
399-A W. Trimble Road
San Jose, CA 95131
(408) 944-6300

Maple Picture Publishing
$400—monochrome image compression software.
Image Software A.S.
Norway

DiskDoubler Plus
Hardware support card for DiskDoubler software.
Salient Software
3101 Avalon Court
Palo Alto, CA 94306
(415)-852-9567

NeoTech Image Compressor
$1300—video compression NuBus board with CL550 DSP chip.
Neotech, Hampshire, England
Distributed in United States by
Advent Computer Products Inc.
449 Santa Fe Drive, Suite 213
Encinitas, CA 92024
(619) 942-8456

Digital Film
SuperMac CL-550 video compression system; less than $2,000; compresses 24-bit color images and full-motion video, as well as digital audio.
SuperMac Technology
485 Potrero Ave.
Sunnyvale, CA 94086
(408) 245-2202

VideoCube
About $3,595 with an on-board CL-550 compression chip; without the chip-based compression, the price would be about $1,380.
Lumiere Technology
30 Rue Barge
75015 Paris, France
1-40-56-97-13

REMOTE BRIDGES

3000 Series CB
Remote Ethernet bridge that compresses data by an average ratio of 4-to-1.
Cryptall Communications Corp.

MLB/6000 Relay Gold Marathon 5K Data/Voice Network Server
Supports as many as four wide area network lines, allowing integration of data, voice, and fax information on dedicated circuits. Prices for the Marathon 5K start at $3,000. Additional data and voice/fax expansion modules will be priced at $1,250 and $1,400, respectively.
Micom Communications Corp.
4100 Los Angeles Ave.
Simi Valley, CA 93063
(805) 583-8600

FastLane
Bridge, priced at $8,500, that links Ethernet and Token-Ring LANs over T-1 lines using data-compression technology.
Telco Systems, Inc.
Fremont, CA

MICRO-TO-MAINFRAME LINKS

Packet/Flash software
Used with Packet/3270. Emulation software costs $6,000 for a mainframe component and $75 per PC for the first 50 copies.
Mark Orenstein
Packet/PC, Inc.
Farmington, CT
(203) 678-1961

DataPacker
CICS terminal traffic reduction software.
H&M Software Systems
25 E. Spring Valley Ave.
Maywood, NJ 07607-2150
(201) 845-3357 or
(800) FOR-DEMO

CICS/CPR
Reduces CICS to 3270 terminal traffic volume.
MacKinney Systems

COMPRESSION CHIPS

New compression products are being developed that use the
ICs in conjunction with fast 32-bit floating point Digital Signal Processing (DSP) devices. Plessey Semiconductors, Inmos Corp and SGS-Thompson Microelectronics have announced FFT or DCT-based ICs. Intel and Datacube Inc are producing real-time image compression products.

IC-105 Data Compression Coprocessor
Costs $81 for 1,000-up, $50 for 100,000-up.
InfoChip Systems, Inc.
Santa Clara, CA
(408) 727-0514

Mystic chip coprocessor
Hewlett-Packard Co.

CL550 image compression integrated circuit
Costs $155 in quantities of 10,000 (JPEG-compliant DSP compression chip).
C-Cube Microsystems, Inc.
399-A W. Trimble Road
San Jose, CA 95131
(408) 944-6300

RC2324AC two-chip set
Costs $35 (incorporates the V/42 bis data compression algorithm).
Rockwell International Corp.

PUMA (Programmable Universal Micro Accelerator)
Chips and Technologies
(408) 434-0600

Inmos
(Offers DCT chips.)
Texas Instruments,
Cypress Semiconductor

Am95C71 Video Compression & Expansion Processor (VCEP)
AMD 7970 data compression device.
Advanced Micro Devices, Inc.

OTI95C71
(Manufacturing and marketing rights to Advanced Micro Devices' Am95C71 Video Compression & Expansion Processor.)
Oak Technology, Inc.

L64700 series
High-performance video compression processors provide full-motion video.
LSI Logic Corp.

Genesis chip set
Incorporates the functions of a 386SX processor, a display controller, a memory controller, and I/O ports.
Intel Corp.

AUDIO COMPRESSION

DS2270 Speech Recorder Stik
Digital audio recording subsystem; $96.25 in quantities of 500.
Dallas Semiconductor

STILL IMAGE COMPRESSION STANDARDS

CCITT Group 3
Facsimile compression.
Joint Photographic Experts Group (JPEG)

FULL MOTION VIDEO COMPRESSION STANDARDS

Motion Picture Experts Group (MPEG) video compression

Digital Video Interactive technology
100 : 1 data compression enables a HDTV image to be carried at less than 40M-bits/s.
Intel Corp.

The MC504 multiplexer
Ashton Tate

Piculator (Software)
Costs about $300; Radius Video Collage Engine (three-board set), sold exclusively through OEMs and systems integrators for approximately $10,000.
Radius, Inc.
1710 Fortune Drive
San Jose, CA 95131
(408) 434-1010

DigiCipher
Digital video compression.
General Instrument Corp.

Mirror III version 2.0
Costs $149; communications software with MNP Class 5 error-correction and data-compression protocols.
Softklone Distributing Corp.

MODEMS

Microcom QX/V.32C modem
(Using the Microcom Networking Protocol.)

V-series Ultra Smartmodem 9600
Costs $1,199; combines speed, error correction, and data compression into a sturdy, external device; supports both MNP 5 and CCITT V.42bis data-compression algorithms.
Hayes Microcomputer Products, Inc.
Atlanta, GA
(404) 449-8791

ProModem 9600M
9600 bps data modem with FAX send and receive supporting MNP-5 and V.42bis data compression.
Prometheus

Teleport/Fax modem
Costs $325; 2400 bps modem with MNP/Level 5 data compression and a fax-sending machine.
Global Village
Menlo Park, CA

Sendfax SC11046 chip set
Combines 4,800-bit/s Group 3 fax capabilities with a full-duplex, Hayes-compatible V.22bis 2,400-bit/s modem.
Sierra Semiconductor

73D224
2,400-bit/s modem chip set with error-correction and data-compression functions.
Silicon Systems

Rembrandt (high-bandwidth) and
Rembrandt II/06 (low-bandwidth) video codecs
Sophisticated devices that compress video signals to permit them to be economically transmitted over digital transmission networks.
Compression Labs, Inc.

Data Packer/DB2
License ranges from $15,000 to $57,500; provides multiple compression techniques, requires no additional address space and makes no modifications to DB2 or application programs.
BMC Software, Inc.
Sugar Land, Texas

MediaView
Data compression algorithm and a multimedia digital publishing system.
Los Alamos National Laboratory

DT2856 line-scan processor board
Costs $3,995; real-time image processing and data compression.

3200 family chip sets:

32FX16—optimized for fax applications, with on-chip support hardware for implementing DSP algorithms.

32CG160—includes BitBLT hardware, targeted toward page printers, graphics terminals, and document scanners.

32GX320—the highest-performance device, designed for high-end printers and fax machines.

National Semiconductor
2900 Semiconductor Drive
Santa Clara, CA 95052-8090
(408) 721-5000

WFFCA

Data compression program for Data General Computers.
:SYSMGR

Scitex

High-end color publishing system manufacturer, reportedly has used DCT compression techniques in its products for several years.

INTERNAL DISTRIBUTION

1. M. S. Bronzini
2. G. A. Dailey
3. J. L. Duncan
4. R. T. Goeltz
5. R. K. Gryder
6-15. K. A. Hake
16. R. B. Honea
17. D. E. Reichle
18. R. B. Shelton
19. B. E. Tonn
20. G. W. Westley
21. S. L. Yount
22. Central Research Office
23. Document Reference Section
24-25. Laboratory Records
26. Laboratory Records - RC
27. ORNL Patent Office

EXTERNAL DISTRIBUTION

28-37. Ltc. Wayne Bailey, Department of the Army, Project Manager AIM, Attn: SARD-IMO (AIM), Stop C-113, Ft. Belvoir, VA 22060-5895.
38-67. T. Branning, Department of the Army, Project Manager AIM, Attn: SARD-IMO (AIM), Stop C-113, Ft. Belvoir, VA 22060-5895.
68-70. D. Howard, Department of the Army, Project Manager AIM, Attn: SARD-IMO (AIM), Stop C-113, Ft. Belvoir, VA 22060-5895.
71. Dr. Bruce G. Buchanan, Department of Computer Science, University of Pittsburgh, 206 Mineral Industries Building, Pittsburgh, PA 15260.
72. Dr. Helen M. Ingram, Director, Udall Center for Studies in Public Policy, The University of Arizona, 803/811 East First Street, Tucson, Arizona 85719.
73. Dr. Allan Hirsch, Vice President, Environmental Sciences and Director, Washington Operations, Midwest Research Institute, 5109 Leesburg Pike, Suite 414, Falls Church, VA 22041.
74. Mr. Calvin D. MacCracken, President, Calmac Manufacturing Corporation, 101 West Sheffield Avenue, Englewood, NJ 07631.
75. Dr. Martin Williams, Professor, Department of Economics, Northern Illinois University, DeKalb, IL 60115.
76. Ms. Jacqueline B. Shrago, Director, Office of Technology Transfer, 405 Kirkland Hall, Vanderbilt University, Nashville, TN 37240.
77. Office of Assist Manager for Energy Research and Development, DOE/ORO, P.O. Box 2001, Oak Ridge, TN 37831-8600.
78-87. Office of Scientific and Technical Information, U.S. Department of Energy, P.O. Box 62, Oak Ridge, TN 37831.