

Selected Papers in the Applied Computer Sciences 1994



U.S. Geological Survey
Bulletin 2103



AVAILABILITY OF BOOKS AND MAPS OF THE U.S. GEOLOGICAL SURVEY

Instructions on ordering publications of the U.S. Geological Survey, along with prices of the last offerings, are given in the current-year issues of the monthly catalog "New Publications of the U.S. Geological Survey." Prices of available U.S. Geological Survey publications released prior to the current year are listed in the most recent annual "Price and Availability List." Publications that may be listed in various U.S. Geological Survey catalogs (see **back inside cover**) but not listed in the most recent annual "Price and Availability List" may be no longer available.

Reports released through the NTIS may be obtained by writing to the National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161; please include NTIS report number with inquiry.

Order U.S. Geological Survey publications **by mail** or **over the counter** from the offices given below.

BY MAIL

Books

Professional Papers, Bulletins, Water-Supply Papers, Techniques of Water-Resources Investigations, Circulars, publications of general interest (such as leaflets, pamphlets, booklets), single copies of Earthquakes & Volcanoes, Preliminary Determination of Epicenters, and some miscellaneous reports, including some of the foregoing series that have gone out of print at the Superintendent of Documents, are obtainable by mail from

**U.S. Geological Survey, Information Services
Box 25286, Federal Center, Denver, CO 80225**

Subscriptions to periodicals (Earthquakes & Volcanoes and Preliminary Determination of Epicenters) can be obtained **ONLY** from the

**Superintendent of Documents
Government Printing Office
Washington, DC 20402**

(Check or money order must be payable to Superintendent of Documents.)

Maps

For maps, address mail orders to

**U.S. Geological Survey, Information Services
Box 25286, Federal Center, Denver, CO 80225**

Residents of Alaska may order maps from

**U.S. Geological Survey, Earth Science Information Center
101 Twelfth Ave. - Box 12
Fairbanks, AK 99701**

OVER THE COUNTER

Books and Maps

Books and maps of the U.S. Geological Survey are available over the counter at the following U.S. Geological Survey offices, all of which are authorized agents of the Superintendent of Documents:

- **ANCHORAGE, Alaska**—Rm. 101, 4230 University Dr.
- **LAKEWOOD, Colorado**—Federal Center, Bldg. 810
- **MENLO PARK, California**—Bldg. 3, Rm. 3128, 345 Middlefield Rd.
- **RESTON, Virginia**—USGS National Center, Rm. 1C402, 12201 Sunrise Valley Dr.
- **SALT LAKE CITY, Utah**—Federal Bldg., Rm. 8105, 125 South State St.
- **SPOKANE, Washington**—U.S. Post Office Bldg., Rm. 135, West 904 Riverside Ave.
- **WASHINGTON, D.C.**—Main Interior Bldg., Rm. 2650, 18th and C Sts., NW.

Maps Only

Maps may be purchased over the counter at the following U.S. Geological Survey offices:

- **FAIRBANKS, Alaska**—New Federal Bldg., 101 Twelfth Ave.
- **ROLLA, Missouri**—1400 Independence Rd.
- **STENNIS SPACE CENTER, Mississippi**—Bldg. 3101

Selected Papers in the Applied Computer Sciences 1994

Edited by D.A. WILTSHIRE

U.S. GEOLOGICAL SURVEY BULLETIN 2103

*This volume is published as chapters A–C.
These chapters are not available separately*



U.S. DEPARTMENT OF THE INTERIOR
BRUCE BABBITT, Secretary

U.S. GEOLOGICAL SURVEY
GORDON P. EATON, Director

For sale by U.S. Geological Survey, Information Services
Box 25286, Federal Center, Denver, CO 80225

Any use of trade, product, or firm names in this publication is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Published in the Eastern Region, Reston, Va.
Manuscript approved for publication June 6, 1994.

Library of Congress Cataloging in Publication Data

Selected papers in the applied computer sciences, 1994 / edited by D.A. Wiltshire.

p. cm. — (U.S. Geological Survey bulletin ; 2103)

Includes bibliographical references.

1. Computer science. 2. Earth sciences—Data processing. I. Wiltshire, Denise A.
II. Series.

QA76.S415 1995

550'.285—dc20

94-30441
CIP

FOREWORD

In 1988, the U.S. Geological Survey began publishing a compilation of papers on technical advances in the applied computer sciences. The series, titled *Selected Papers in the Applied Computer Sciences*, includes short reports that describe computer applications in support of earth-science investigations and research. This volume is the fourth in the series. The topics presented in the compilation are:

- Integration of three software packages to visualize a digital terrain flyby.
- Comparison of electronic scanning and hand-digitizing techniques for producing hydrologic maps.
- Techniques used in the design and implementation of a local-area network.

CONTENTS

[Letters designate the chapters]

Foreword

- (A) Using Geographic Information, Image Processing, and Animation Systems
to Visualize a Digital Terrain Flyby
By Robert G. Clark, John W. Jones, Thomas E. Ciciarelli, and
Daniel F. Stanfill IV
- (B) Digitizing Options for Hydrologic Maps: Scanning or Hand Digitizing?
By Andrew Cohen
- (C) Simple Mixed-Vendor Networking for Remote Sites
By Bruce R. Johnson

CHAPTER A

Using Geographic Information, Image Processing, and Animation Systems to Visualize a Digital Terrain Flyby

By ROBERT G. CLARK, JOHN W. JONES, THOMAS E. CICIARELLI,
and DANIEL F. STANFILL IV

SELECTED PAPERS IN THE APPLIED COMPUTER SCIENCES 1994

U.S. GEOLOGICAL SURVEY BULLETIN 2103



CONTENTS

Abstract	A1
Introduction	1
Background	1
Software Integration.....	2
ARC/INFO	2
Khoros.....	2
Surveyor	4
Products.....	5
Conclusions	5
References Cited	6

FIGURES

1. Diagram showing functions performed by each of the three software packages.....	A2
2, 3. Sample screen captures from Khoros showing—	
2. Surveyor glyph options	3
3. ARC/INFO glyph options	4
4. Diagram showing Khoros workspace used to produce terrain flyby	5
5. Digital orthophoto quadrangle combined with a digital elevation model.....	6

Using Geographic Information, Image Processing, and Animation Systems to Visualize a Digital Terrain Flyby

By Robert G. Clark,¹ John W. Jones,¹ Thomas E. Ciciarelli,¹ and Daniel F. Stanfill IV²

ABSTRACT

It is becoming more and more difficult to procure expensive analytical software because of decreasing Government and university funding. Therefore, this multidivision, multiagency project focused on integrating commercial software (ARC/INFOTM), free software (KhorosTM), and public software (Surveyor) to merge several U.S. Geological Survey data sets and produce terrain flybys. The merged data sets included a digital orthophoto quadrangle, a digital elevation model, and the transportation and hydrography digital line graph layers for a mountainous area in Idaho. Preprocessing of the data was performed with ARC/INFO, a geographic information system software. After merging of ARC/INFO digital orthophoto quadrangle and digital line graph images, routines constructed in Khoros produced red, green, and blue images and reformatted them for input to Surveyor. Surveyor software was then used to produce several digital orthophoto quadrangle and digital line graph terrain flybys. By integrating the three software packages, not only were the best features of each exploited but also money was saved.

INTRODUCTION

The U.S. Geological Survey (USGS) and the Jet Propulsion Laboratory (JPL) are investigating the feasibility of integrating free software (Khoros), inexpensive public software (Surveyor), and commercial software (ARC/INFO) on commonly used hardware for visualizing large earth-science data sets. Visualization can help earth scientists view their data and make new discoveries that are often difficult to detect with conventional analytical techniques.

Data sets used in scientific visualization often require preprocessing. The complexity of this preprocessing and the large size of the data sets add to the overall difficulty of the analysis. For example, the nonstandard digital orthophoto

quadrangle (DOQ) file used in this project required approximately 8 megabytes of storage and can be considered a large earth-science data set. In addition, a standard DOQ having a 1-m resolution can be as large as 50 megabytes. The software used to prepare and visualize these large spatial data sets typically has to simultaneously manipulate several large working data sets. In addition to keeping costs down, the capability to handle many large spatial data sets was one of the project's reasons for selecting ARC/INFO, Khoros, and Surveyor.

BACKGROUND

Seeing the need for preprocessing large earth-science data sets prior to visualization, the National Aeronautics and Space Administration (NASA), the funding organization for Surveyor, suggested embedding Surveyor into Khoros. In an initial meeting, the JPL Surveyor authors, NASA personnel, and a USGS representative discussed Khoros' capability to perform preprocessing (for example, application of image-processing algorithms) on data prior to any of Surveyor's processing. As a result, USGS and JPL researchers performed an initial software integration so that image-processing functions could be applied to terrain data before they were loaded into Surveyor for rendering.

The Khoros visual programming environment is an image-processing system that can distribute its processing over a local-area network or a wide-area network. The JPL Surveyor system is used to generate terrain flybys. In this project, data management (for example, data importing) and some data manipulation (for example, data set gridding and regridding) were performed by an off-the-shelf commercial geographic information system (GIS) software, ARC/INFO. After the DOQ, digital elevation model (DEM), and digital line graph (DLG) images were preprocessed in ARC/INFO, Khoros was used to merge the ARC/INFO DLG (that is, roads and streams) rasterized images with the DOQ image and produce red, green, and blue (RGB) images. In turn, these preprocessed Khoros RGB images were used to construct flight paths and create animation frames in Surveyor. This process flow is illustrated in figure 1.

¹U.S. Geological Survey, Reston, VA 22092.

²Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109-8099.

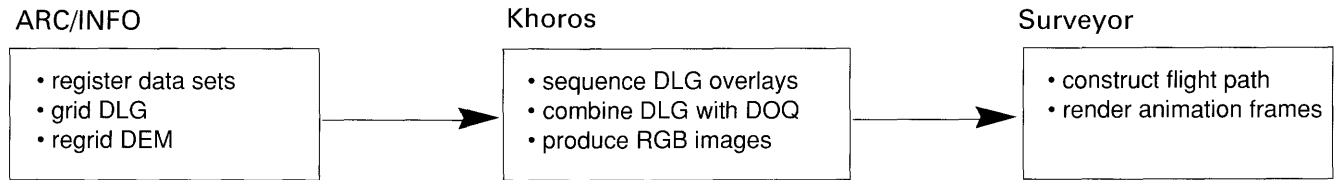


Figure 1. Functions performed by each of the three software packages.

SOFTWARE INTEGRATION

Software integration was accomplished in two steps. The first step was to develop a shell script to convert Khoros VIFF-formatted files to JPL's Video Image Communication and Retrieval (Vicar)-formatted files and then construct Surveyor pyramid-structured databases from these Vicar-formatted files. This shell script was used as a model for the development of a "C" program (that is, a program developed in the "C" programming language). This program uses the same programs that the shell script does. By using existing programs of the JPL and the University of New Mexico, maintenance is minimized. For example, if one program was constructed to replace existing programs, it would require some reprogramming every time a file format or data type was changed. One of this project's programs made function calls to three already existing programs: one that converted Khoros VIFF-formatted files to a raw 8-bit data file, another that converted a raw 8-bit data file to a Surveyor Vicar-formatted file, and a third that constructed a Surveyor pyramid database. This "C" program also included a function to start Surveyor with a configuration file, a "worldfile" that Surveyor used to point to its pyramid database. With the generated configuration file, the user can start building a terrain flyby immediately and does not have to identify databases, load the databases, or specify initial application parameters.

The second step in the development of a Khoros-Surveyor interface involved building a Khoros glyph for the "C" program mentioned above. A glyph is a graphical representation (an icon) of a Khoros program that can be manipulated in Khoros (the facility where glyphs can be selected and operated on either individually or as a group). The Khoros glyph was developed to represent a small "C" program using the Khoros utility Composer. The resulting Khoros-Surveyor glyph is illustrated as an open glyph in figure 2. The user specifies the Khoros input files and the Surveyor pyramid database levels. Alternatively, the input files may be specified by connecting the glyphs symbolically. A second glyph was also constructed as part of the software integration effort. This glyph, *arc2viff*, converted ARC/INFO raster files to Khoros VIFF-formatted files. In addition to taking the ARC/INFO raster file name as input,

this second glyph required the ARC/INFO header file name associated with the ARC/INFO raster file. The header file provides the number of rows, the number of columns, and the data type of the raster file. The output file name also had to be specified. Figure 3 illustrates the open *arc2viff* glyph.

ARC/INFO

GIS's are robust in data management and manipulation functions such as data registering and resampling (Raper and Maguire, 1992). ARC/INFO is a widely used GIS software; the USGS has many network copies, each of which simultaneously services several users. For this project, the ARC/INFO effort was performed by the USGS's Reston GIS Research Laboratory. ARC/INFO was used to construct raster files to the same dimensions, resolution, and cell size as the DOQ. First, the DOQ was imported into ARC/INFO as an 8-bit band-interleaved-by-line (bil) file (Environmental Systems Research Institute, Inc., 1991). An ASCII header that indicated the georeferencing, cell size, and number of pixels per line for the DOQ was created with a text editor. ARC/INFO then used this information to read and reformat the DOQ to an ARC/INFO grid file (that is, an 8-bit raster file). Next, the DEM data were read into ARC/INFO by using the "demlattice" command. The gridded DEM data set originally had a 30-m resolution. The DLG layers were also imported into ARC/INFO by using the "dlgarc" command. Registration of these data sets was verified by displaying the raster (DOQ and DEM) and vector (DLG) layers simultaneously. The "linegrid" command was used to convert the DLG layers to raster files having the same resolution as the DOQ (4 m). The DEM data were resampled to the same resolution by using the grid "resample" command. After all data sets had been rasterized and resampled in ARC/INFO, the DEM and DLG data were exported as ARC/INFO bil files.

KHOROS

Khoros was selected for this project because of several important attributes: a visual programming environment,

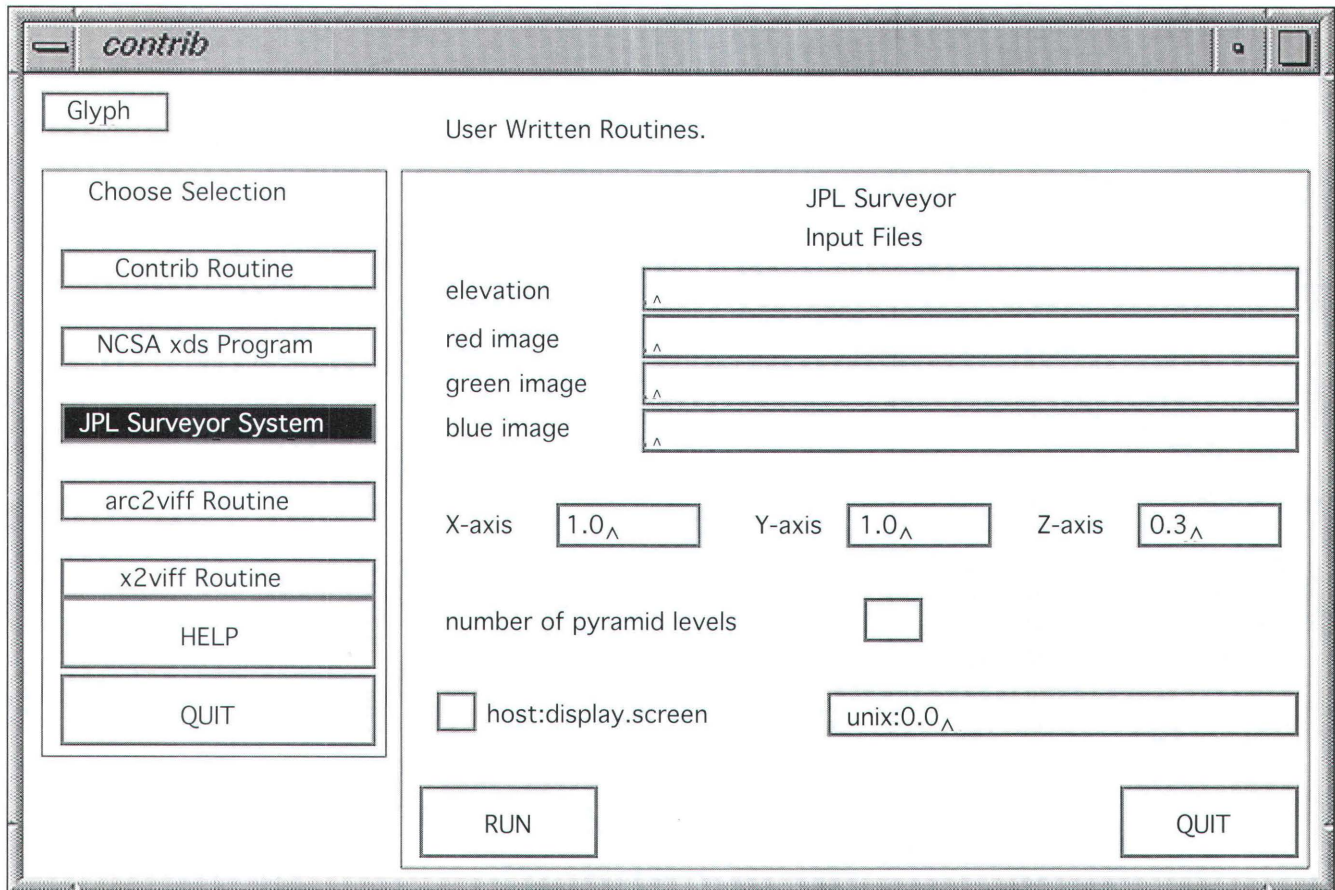


Figure 2. Sample screen capture from Khoros showing Surveyor glyph options.

automatic generation of user interfaces, and image-processing facilities (Rasure and others, 1990). Khoros is copyrighted but is available free of charge via Internet to file transfer protocol (ftp) program anonymous users. It is used worldwide throughout the Internet community. This project used Khoros to process the ARC/INFO raster data sets into Surveyor Vicar-formatted data sets. After the ARC/INFO registered data sets were loaded into Khoros, basic data merging was performed. The three Khoros-merged RGB raster images and the DEM image were loaded into Surveyor. Figure 4 illustrates the Khoros glyphs required to perform the data merging and Surveyor initialization. This Khoros workspace was used to construct RGB images from the DOQ and the two DLG data layers. The DOQ is an 8-bit raster data set and was imported into Khoros with the arc2viff glyph. The DEM data are in a 16-bit raster file and were also imported into Khoros by using the arc2viff glyph. Several masks were built for each of the two DLG data layers (that is, transportation and hydrography) to ensure they would be correctly colored by Surveyor.

The transportation layer was colored red by assigning the red channel a value of 200, the green channel a value of 0, and the blue channel a value of 0 wherever the DLG transportation data layer overlaid the DOQ. A red value of 200 was selected instead of a bright value of 255, because it would be less likely to "bleed" in various types of visualization methods. In this way, every original ARC/INFO transportation value of 2 was converted to 200 in the Khoros glyph vsubstit. The hydrography DLG data were colored blue by assigning the red channel a value of 0, the green channel a value of 0, and the blue channel a value of 200 wherever an original ARC/INFO hydrography value was 3. This procedure too was accomplished by using the Khoros glyph vsubstit. Wherever the transportation layer intersected the hydrography layer, the transportation layer took precedence, and the intersecting pixel was colored red. This order of precedence for the two DLG layers was handled primarily by the two glyphs, vand and vsub (located approximately in the center of fig. 4). The result of the processing of these two glyphs was used as a mask for the red

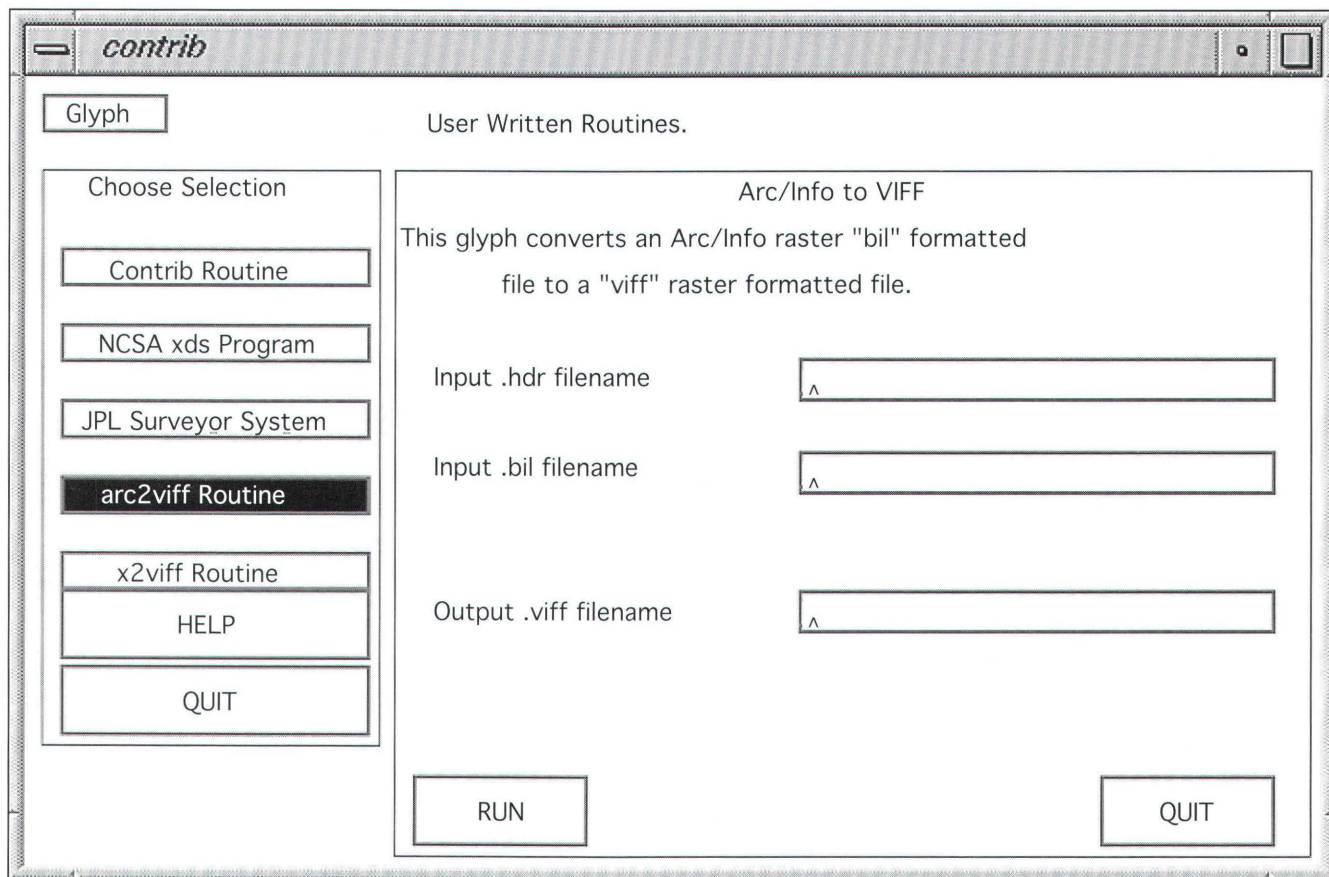


Figure 3. Sample screen capture from Khoros showing ARC/INFO glyph options.

channel in the subsequent vreplace glyph. The vreplace glyph applies the DLG overlays to the DOQ according to its color and color mask values. The two vor glyphs in figure 4 each combine one DLG data layer with the DOQ image. All pixels having a 0 in a DLG data layer were set to the DOQ pixel's values. A DOQ pixel has the same value for each of the RGB channels to keep the DOQ in a gray scale. The vfloor glyph constructs a black mask (that is, a zero pixel value) for the DLG data layer pixels and is used to help define the red, green, and blue colors of the two DLG data layers (Rasure and Argiro, 1991). Finally, Surveyor uses the Khoros-formatted files (that is, elevation and RGB images) as input to the Surveyor glyph located at the right side of the workspace illustrated in figure 4. Figure 5 is a view from Surveyor showing the DLG transportation layer colored in red and the hydrography layer colored in blue. The last glyph to execute, the Surveyor glyph, starts the Surveyor system.

SURVEYOR

Surveyor is one of the few terrain flyby software packages that can distribute its rendering to multiple machines. In addition, the size of the images Surveyor can process is limited only by the size of the operating system's virtual memory address space and not by array size within the Surveyor software or the size of the system swap space. To construct the DOQ flyby, the frames were rendered on five Silicon Graphics Incorporated (SGI) IRIS Indigos and two multiprocessor Sun Microsystems machines. A higher quality level for the pyramid was achieved by using an oversampling of nine rays per pixel instead of the default of two rays. The pyramid is defined as a precomputed series of reduced-resolution images scaled down from an original by user-selectable filters. The size of the output frames was set to 640×480. The frames were then converted from Vicar file format to SGI RGB file format. Once the frames were in RGB format, various products were produced for the scientific community.

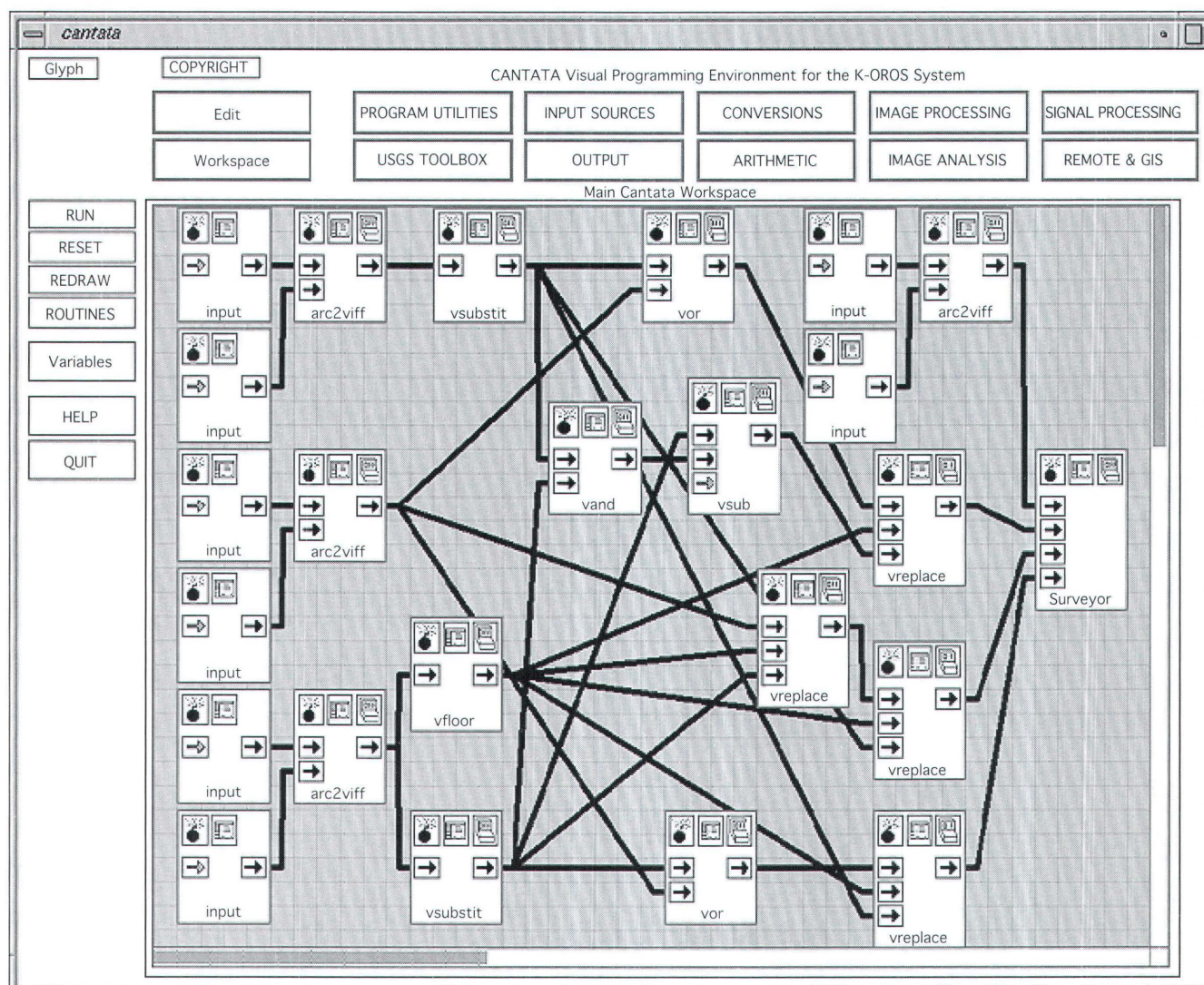


Figure 4. Khoros workspace used to produce terrain flyby.

PRODUCTS

Several products were created by this project. Videotapes created by using SGI's "moviemaker" application program were used as a learning tool and discussion point for further investigations. In addition, the SGI RGB-formatted frames were converted to an SGI movie format for display in the USGS Cartographic Technology Laboratory and in the USGS Technology Information Centers in Reston, Va., and Menlo Park, Calif. The SGI movies in the Technology Information Centers have an audio channel. A Moving Picture Expert Group movie was also produced. Color hard copy of several perspective views of the constructed model are displayed in the USGS Reston Scientific Visualization Laboratory. Finally, Surveyor and arc2viff glyphs are available from the authors.

CONCLUSIONS

This project demonstrated that free, low-cost public, and commercial software can be interfaced and used to visualize earth-science data. Large spatial data sets were merged and portrayed in a realistic fashion. Several products were made available to the scientific community. With these visualizations, possible inconsistencies between different types of data for a given location could be investigated. These techniques and the software developed can be applied to other spatial locations as well as to other types of data. The feasibility of visualizing earth-science data by combining the best features of several software packages residing on nonspecialized, relatively inexpensive hardware was demonstrated. This low-cost processing frees more funding for other earth-science research issues.

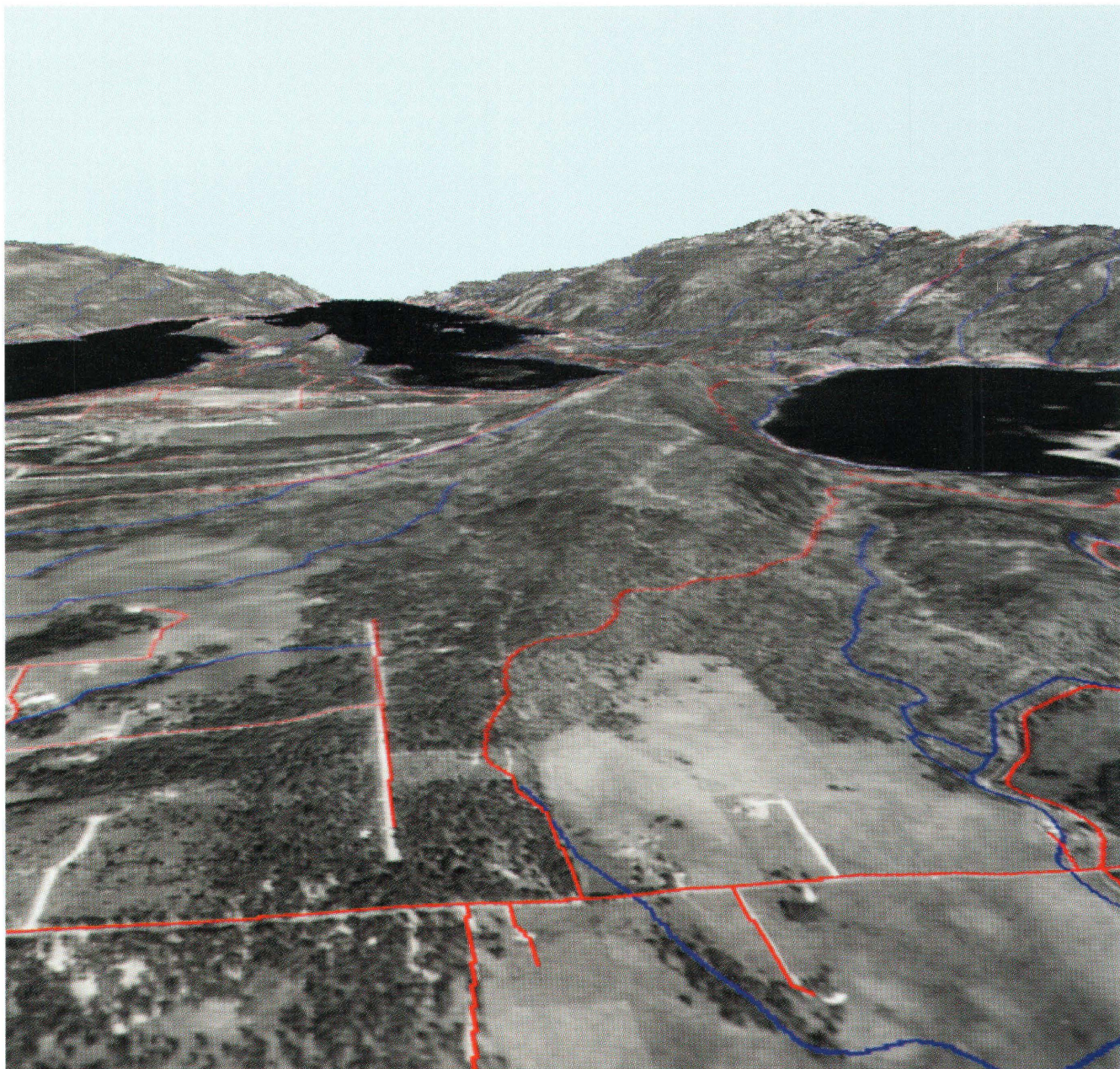


Figure 5. Digital orthophoto quadrangle combined with a digital elevation model showing the transportation digital line graph in red and hydrography in blue.

REFERENCES CITED

- Environmental Systems Research Institute, Inc., 1990, Understanding GIS, the ARC/INFO method: Redlands, Calif., Environmental Systems Research Institute, Inc., variously paged.
- Raper, J.F., and Maguire, D.J., 1992, Design models and functionality in GIS—Computers and geosciences: An International Journal, v. 18, no. 4, p. 387.
- Rasure, John, and Argiro, Danielle, 1991, Khoros user's manual: Albuquerque, University of New Mexico, variously paged.
- Rasure, John, Argiro, Danielle, Sauer, Tom, and Williams, Carla, 1990, Visual Language and software development environment for image processing: International Journal of Imaging Systems and Technology, v. 2, p. 183–199.

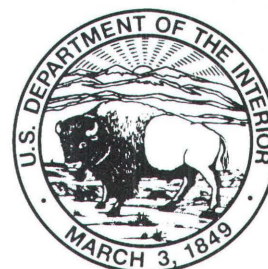
CHAPTER B

Digitizing Options for Hydrologic Maps: Scanning or Hand Digitizing?

By ANDREW COHEN

SELECTED PAPERS IN THE APPLIED COMPUTER SCIENCES 1994

U.S. GEOLOGICAL SURVEY BULLETIN 2103



CONTENTS

Abstract	B1
Introduction	1
Approach	1
Results and Discussion	1
Linework	1
Accuracy	3
Eye Strain	3
Comfort	3
Lost Data	3
Correction	3
Time Required	3
Summary	3

TABLES

1, 2. Time required for conversion of hard-copy images to digital files by—	
1. Contex 8000 scanner and Pixeltrak vectoring software	B2
2. Altek AC40 digitizer	2

Digitizing Options for Hydrologic Maps: Scanning or Hand Digitizing?

By Andrew Cohen

ABSTRACT

Geographic information systems require that spatial data be digitized into specific formats to allow comparison of data from different sources. Two methods of digitizing map features were compared in terms of time requirements for data entry, ease of data entry, and accuracy of results. Scanning the data set electronically was generally more efficient for a series of hydrologic and planimetric maps than hand digitizing the same material, and scanning planimetric maps required more time than scanning hydrologic maps of the same quadrangles.

INTRODUCTION

Two methods of digitizing maps to produce ARC/INFO™ coverages for geographic information systems (GIS) were compared for time, ease of entry, and accuracy. The first method used a Context™ 8000 scanner coupled with PixelTrak™ vectorizing software; the second used an Altek™ model AC40 hand-digitizing board. Map separates on scale-stable mylar film showing hydrologic and planimetric features were obtained for each method.

Basic steps in the scanning method included (1) scanning the mylar separate, (2) converting the resulting raster image file to a vector file, (3) converting the vector file from scanner units to an ARC/INFO coverage in real-world coordinates, and (4) checking the accuracy by overlaying plots of digitized work on the original separates. Time results are summarized in table 1.

Basic steps in the hand-digitizing method included (1) hand digitizing the map features directly into an ARC/INFO coverage by using the Altek digitizer and (2) checking the accuracy by overlaying plots of digitized work on the original separates. Time results are given in table 2.

The scanning method was then applied to map separates depicting two different data sets—planimetric and hydrologic—to determine which of the two sets would yield a faster production time. Both were scanned and vectorized, and the total time needed to produce a final ARC/INFO coverage was recorded. This comparison was not done for the hand-digitizing method because choosing lines to be

digitized from either data set requires the same amount of time. All aspects of the test were performed and recorded by the same person.

APPROACH

The data feature to be digitized was hydrology, including shorelines. Two 7.5-min quadrangles in the Champlain Valley of northeastern New York State (Churubusco, N.Y., and West Chazy, N.Y.) were selected because they contain an average amount of detail. Intermittent streams, indicated by dashed lines, were digitized as solid lines.

Two sets of materials were tested. The first set consisted of planimetric separates of standard U.S. Geological Survey (USGS) 7.5-min 1:24,000-scale quadrangles on clear mylar; these were obtained from the New York State Department of Transportation and contained all features found on the published maps except topographic contours. The second set consisted of hydrology separates for the same quadrangles; these were obtained from the USGS National Mapping Division and contained only the blue-line (hydrology) features found on the published maps.

RESULTS AND DISCUSSION

All items discussed are user and (or) site specific and are not meant to describe general conditions.

LINEWORK

If a feature is represented by a solid line, the scanner (PixelTrak software) is able to complete a large percentage of its vectorizing in a semiautomatic mode; for example, it automatically traces that particular line until it reaches an intersection, at which time the operator must take control. If a feature is represented as a dashed or dotted line, the operator must take control at each line break and digitize the vertex by hand. In the hand-digitizing method, all vertices are entered manually.

Table 1. Time required for conversion of hard-copy images to digital files by Contex 8000 scanner and Pixeltrak vectoring software.

[All values are in minutes]

Quadrangle	Procedure						Total
	Scan ¹	Vectorize ²	Transfer ³	A/I Conv ⁴	QA/QC ⁵	Correction ⁶	
Planimetric separate:							
Churubusco, N.Y.....	5	134	7	17	19	25	298
			7	17	20	8	(4 h 58 min)
			7	17	15		
West Chazy, N.Y.	5	170	7	17	22	40	302
			7	17	17		(5 h 2 min)
Hydrology separate:							
Churubusco, N.Y.....	5	152	7	17	10	10	253
			7	17	20	8	(4 h 13 min)
West Chazy, N.Y.	5	160	7	17	10	10	250
			7	17	17		(4 h 10 min)

¹Scanning consists of loading material into the scanner, performing the prescan functions (such as adjusting light thresholds and dots per inch), actual final scanning, and moving the resulting .RLC (run-length compressed format) raster output file to the PixelTrak partition of the personal computer.

²Vectorizing consists of loading the scanned .RLC file into PixelTrak, selecting and vectorizing each desired line, and saving the resulting collection of vectors in a .DXF (Drawing Exchange File Format) file.

³Transfer refers to physically putting the .DXF file on a diskette, hand carrying it to the personal computer that is linked to the Data General (DG) network, and loading the .DXF file into the ARC/INFO workspace on the DG server.

⁴ARC/INFO conversion is the process of converting the PixelTrak file, which is in scanner units and referenced only to an origin on the scanner, to an ARC/INFO coverage in some useful map projection (usually Universal Transverse Mercator units) and referenced to the New York State grid so it can be used in conjunction with other coverages. This process is largely automated but still requires some user interaction.

⁵During the quality-assurance/quality-control process, a paper plot is made of the A/I coverage and overlain on the original material to check for line accuracy, additions, and deletions. This procedure is standard industry practice.

⁶If any discrepancies are found, the original .DXF file in PixelTrak is corrected. The process then repeats from footnote 3.

Table 2. Time required for conversion of hard-copy images to digital files by Altek AC40 digitizer.

Quadrangle	Time for each process (in minutes)			
	Digitize ¹	QA/QC ²	Correction ³	Total
Planimetric separate:				
Churubusco, N. Y.	236	24	64	385
		20	24	(6 h 25 min)
		17		
West Chazy, N. Y.	210	30	75	417
		25	60	(6 h 57 min)
		17		

¹Digitizing consists of mounting original material on the Altek digitizing board, logging onto the Data General server, creating a new empty coverage (already in a useful map projection, usually Universal Transverse Mercator), and entering each vertex of the desired line into such a file by hand.

²Quality assurance/quality control is the quality-checking process. A paper plot is made of the ARC/INFO coverage and overlain on the original material to check for line accuracy, additions, and deletions. This procedure is standard industry practice.

³If any discrepancies are found, the corrections are made by returning to the Arcedit session described in footnote 1. The process then repeats from this step.

ACCURACY

The linework resulting from the scanning method was excellent. The software divides the thickness of a line into several pixels and selects those representing the middle of the line. If the process yields an erroneous result, the operator can immediately correct the situation. By contrast, erroneous digitizing on the hand-held digitizer is generally not noticed until a check plot is made.

EYE STRAIN

The hand-held digitizer can cause more eye strain than the display monitor used in scanning. The operator must carefully view the linework through a small eyepiece on the keypad, often under poor lighting. The scanner displays the work on a large color monitor and can enlarge the image several times to minimize eye strain.

COMFORT

Continuous use of the hand digitizer is extremely tedious and thus can lead to errors. The operator found digitizing by scanner more comfortable than digitizing by hand.

LOST DATA

Frequently during hand digitizing, the processing would abnormally abort and interrupt the digitizing, and

lost data had to be redigitized. If the operator uses too little pressure on the entry key, data are inadvertently omitted. This problem becomes major when thousands of data entries must be made. Using the scanner eliminates this problem.

CORRECTION

If lines must be corrected during the quality-assurance/quality-control process, the linework generated by the hand-digitizing method must be split into separate lines to remove an erroneous section. Also, where nodes are too far apart to "snap" together, gaps can result. The scanner eliminates this problem also.

TIME REQUIRED

Scanning the hydrology separate to produce a final ARC/INFO coverage took significantly less time than scanning the planimetric separate, because vectorizing the scanned features of a planimetric separate requires that the operator identify the hydrologic features by referring to a paper map—a time-consuming step not needed for hydrology separates.

SUMMARY

Results (tables 1 and 2) indicate that, for a digitizing job containing an average amount of detail, the scanning method yielded more accurate results in a shorter time than the hand-digitizing method.

CHAPTER C

Simple Mixed-Vendor Networking for Remote Sites

By BRUCE R. JOHNSON

SELECTED PAPERS IN THE APPLIED COMPUTER SCIENCES 1994

U.S. GEOLOGICAL SURVEY BULLETIN 2103

The design, selection of standards, hardware, and software, and installation of a local-area network in Spokane, Wash., are described



CONTENTS

Abstract	C1
Introduction	1
Goals	1
Starting Facilities	2
Restrictions and Protocols	2
Central Server versus Peer To Peer	3
Ethernet and 10Base-T	4
TCP/IP and EtherTalk	5
NFS and AppleShare	5
SMTP, BinHex, and MIME	6
PostScript	6
Design Summary	6
Hardware	6
Hubs	7
Network Interface Adapters	8
Wiring	9
Router and Modem	10
Printer and Plotter Servers	10
Printers	12
UPS	12
Software	13
TCP/IP and AppleTalk	13
NFS and AppleShare	14
PC-NFS	14
ChameleonNFS	15
NFS Share	15
Electronic Mail	15
UNIX Daemons	16
<i>lpd</i>	16
<i>sendmail</i>	16
<i>pcnfsd</i>	17
<i>popper</i>	17
Internet Name Resolution	17
Domain Name Service	17
DNS and Dial-Up Connections	19
Primary Name-Server Solution	19
Secondary Name-Server Solution	20
Cache-Only Name-Server Solution	20
DNS Name Look-Up	20
Modifying <i>named</i> 's Search List	21
Running DNS without NIS	21
Services Provided	21
Printing and Plotting	21

File Sharing	22
Remote Login and File Transfer	22
Electronic Mail and Document Attachment	22
Backup	22
Summary	23
References Cited	23
Appendix A. Installing <i>named</i> 4.8.3 And Modifying DNS Search Lists	27
Obtaining <i>named</i> , Version 4.8.3	27
Installing <i>named</i> , Version 4.8.3	27
Building a Resolver Library	27
Updating the Shared Library	27
Compiling <i>named</i>	28
Modifying DNS Search Lists	28
Appendix B. DNS without NIS on SunOS 4.x Systems	29

FIGURES

1. Drawings of possible network configurations	C4
2. Diagram showing Spokane field office network configuration	7

Simple Mixed-Vendor Networking for Remote Sites

By Bruce R. Johnson

ABSTRACT

Building a multiple-vendor local-area network at a remote site can be complex and confusing. Simplicity and suitable standards are the keys to network designs that are straightforward to install and have low maintenance requirements. Why build such a network? The reasons for building a local-area network at the U.S. Geological Survey's Spokane field office and connecting that network to the Internet include the following: (1) providing access to laser-quality printing and large-format plotting for all computer users from their desktop computers, (2) providing file-sharing services to all users, regardless of computing environment, (3) providing remote login and remote file-transfer capability to all users, (4) providing local and remote electronic mail and document-transfer capabilities to all users, and (5) providing automated backup services for all office computers. Computing environments in use in Spokane at the time of this network design included UNIX workstations, Macintosh desktop computers, and PC-compatible computers running either DOS or DOS plus Windows.

The fundamental design philosophy of this system is simplicity. To keep the design as simple as possible, hardware and software standards were adopted where they would not adversely affect users. This philosophy is consistent with a peer-to-peer network configuration using Ethernet as the standard hardware-interface protocol and 10Base-T as the standard wiring protocol. A pair of 10Base-T Hubs was connected to give the network a multiple-star form. As far as possible, all similar desktop computers used identical network hardware and software. A dial-up connection to the Internet was made through a dedicated router and a pair of high-speed modems. Network printers were attached through dedicated print servers. TCP/IP and AppleTalk were adopted as standard network communications protocols and Network File System and AppleShare as standard file-sharing protocols. Other standard systems adopted include SMTP for electronic mail transport, BinHex and MIME for electronic mail document attachment, and PostScript as a page description language for network printer files.

Most design goals have been met. All users have access to laser-quality printing, large-format plotting, file-sharing, electronic mail with document transfer, and the Internet. Key network pieces were tested for compatibility

before committing to purchases by borrowing evaluation units from vendors. Because of this testing, all purchased products were usable in the final network environment. Some difficulties encountered in configuring the systems to communicate properly with the Internet over a dial-up connection were resolved, and the network can now be administered as a part-time job function.

INTRODUCTION

Building a local-area network (LAN) to interconnect computer equipment from multiple vendors is difficult under the best of circumstances. It can be even more of a challenge if the location is remote from both wide-area network connections and computer system professionals. At the beginning of 1993, the Spokane field office of the U.S. Geological Survey (USGS) experienced both of these challenges. There were no computer professionals in the office, and there was no known local access to wide-area network connections. Nevertheless, network services were important enough to proceed with building a LAN. Because there were no computer professionals in the office, major system requirements were simplicity of design, ease of installation, low cost, and low maintenance.

GOALS

There were several reasons for committing the time and money to installing this LAN, most of which involved improving the efficiency of the office staff in their normal tasks and improving the quality of their products. The first goal was to provide all computer users with equivalent access to laser-quality printing and large-format plotting. At the beginning of this project, there were several laser printers in the office, but few users had direct access to them from their desktop computers. Another goal was to provide local and remote file-sharing capabilities for all users. File-sharing capabilities at the beginning of this work consisted mostly of copying information to floppy disks and carrying or mailing the disks to the computer where the information was needed. Although this method is effective (if slow) for sharing small files, it becomes ponderous for large data files.

The USGS's Branch of Western Mineral Resources (WMR) includes offices in Menlo Park, Calif., Tucson, Ariz., Reno, Nev., and Spokane Wash. At the time of this network design, the position of chief of WMR had recently been moved from California to the Spokane office. Because the rest of the branch administrative staff remained in the California office, the branch could save significant time and resources by using electronic mail between the two sites. Because other offices in WMR had access to the Internet, connection to the Internet became a system design goal to provide electronic mail (e-mail) between all Spokane users and the rest of the branch. A further goal was to provide a system for attaching binary files to e-mail messages so that word processor documents could easily be transferred to and from the rest of the branch. Finally, the office needed an automated system that would back up files from all computers within the office to a centralized tape drive without individual user intervention.

STARTING FACILITIES

At the beginning of this project, the following computer hardware was available within the Spokane field office:

1. A UNIX-based workstation that had been purchased to provide a platform for geographic information system (GIS) software and to be the central facility for the hoped-for LAN. The workstation—a Sun SPARCstation II—has 32 MB of internal memory and approximately 1.5 GB of disk storage. The operating system is Sun's version of UNIX, SunOS 4.1.2. The Sun has the following peripherals: a 150-MB ¼-inch tape drive; a 2.3-GB 8-mm tape drive; a 600-MB removable-cartridge, magneto-optical disk drive; a CD-ROM reader; a laser printer; a 19-in. high-resolution color graphics monitor; and a 9600-baud modem. A second UNIX workstation was purchased during the project. This system—a Data General Avion—has 28 MB of internal memory and approximately 1.3 GB of disk storage. The operating system for this computer is Data General's version of System V UNIX. Although each workstation uses a version of UNIX, the operating systems are significantly different, particularly at the systems administration level.
2. Two shared desktop computers, an Apple Macintosh IIcx and an IBM PC (Intel 8088 processor). These machines were in an open space where all staff members had access to them. The Macintosh has an 80-MB hard disk and is still in use as a shared resource in the computer laboratory, primarily for word processing, spreadsheet data analyses, and graphics. The original PC has been replaced by a Gateway 2000 486/33 computer having approximately 340 MB of hard-disk storage and a 90-MB removable-cartridge Bernoulli Box. The Gateway is used as a GIS platform and for access to laser printing

and large-format plotting by users who are not connected to the LAN.

3. Desktop computers in individual scientists' offices and in local administrative offices. These computers are a mix of Macintosh models using the Macintosh operating system and IBM PC's and PC-compatibles using either the DOS operating system alone or DOS with Microsoft Windows. In this report, the terms DOS computer and Windows computer will be used to refer to the respective PC-compatible computers, where necessary.
4. A variety of output peripherals, including dot-matrix printers attached to most desktop computers; a Hewlett-Packard LaserJet II laser printer attached to the shared Gateway computer; an Apple LaserWriter laser printer accessible by several Macintosh desktop systems; and a large-format pen plotter and 24×36-in. digitizer, both attached to the shared Gateway computer.

Some equipment was shared by several users at the beginning of this project. A smart port-sharing device allowed three PC-compatible users to share the LaserJet II printer and the pen plotter. The LaserWriter laser printer was located in a group of three open-space offices and was shared initially by two Macintosh users with a small Apple-Talk/PhoneNet network. Later, this network was expanded to include a PC compatible running TOPS software.

As a further complication, one staff member located offsite, approximately 20 mi away, required the same access to facilities as the rest of the staff. Initially, the only connection available was by 9600-baud modem to the Sun workstation. The modems provided access to the workstation but not graphics capability or links to other facilities such as laser printing and plotting. Several desktop systems also had modems attached, so that remote access was possible for some users, generally at 1200 or 2400 baud.

RESTRICTIONS AND PROTOCOLS

A fundamental requirement of this LAN design was to keep the system as simple as possible. Fulfilling this requirement meant adopting hardware and software standards that would not adversely impact users. Deciding where to apply standards and where not to always involves balancing user flexibility with system administration overhead. Some examples may be instructive. Because the users in this office had already invested time and resources in desktop computers from several vendors and with diverse operating systems, it would have been difficult and counterproductive to attempt to impose a single hardware/operating system standard. It would have been similarly counterproductive to attempt to impose primary-application software standards on all users. Many users had been using the same word processor, spreadsheet program, or other software for years. There was no reason to incur the training penalty of switching users to a standard application suite. The network

design had to be flexible enough to work with the existing hardware and software. On the other hand, most users had little or no experience with file- or printer-sharing software, so it made more sense to adopt standard software packages for these functions. The same was true of the hardware connections between computers. Users had little experience with network connections, and the type of connection utilized had little impact on the functionality of their computers. Minor limitations in the choices allowed each user significantly increased the ease of administering the network.

CENTRAL SERVER VERSUS PEER TO PEER

The first major design decision was whether to set up the network with a central server or with a peer-to-peer configuration. Most networks are currently designed with central servers. The central server is usually a high-speed system configured with large mass-storage devices that acts as the main data-storage locality for the network. Central servers run a network operating system that keeps track of resources on the network and allows client computers on the network to communicate with the server and with each other. The server is generally used as a mail repository for the entire network, usually drives high-speed, high-quality printers, and is typically used as the computer that runs network versions of application software. Because each function provides a service to the rest of the computers on the network, the central computer is called a server; computers that use these services are called clients. On large networks, it is common to split the central server functions between two or more computers. Thus, there may be one or more file servers (for data storage), mail servers, print servers, and application servers.

Building a central server system is particularly easy if the network consists of computers of the same type (same vendor or, at least, the same operating system). The variety of computers in use at this site made the adoption of the central server model difficult. Until recently, it was not possible to link the four operating environments in use in this office—UNIX, Macintosh, DOS, and DOS plus Windows—in a central server network and still maintain equal access across all systems. At least one vendor is now offering a central server network operating system with client modules for all four operating environments. However, introducing such a system would require the administration of yet another operating system and the purchase of another computer.

Central server systems are also particularly useful if many users run the same application software or require access to a common database. It can be advantageous to install a network version of a widely used software on a server so that the applications can be reached from each client computer. Having only one version of each application

in use reduces administrative problems. Because users at this site run many applications, there would be little to gain from network versions of application software. Users at this site do not require frequent access to shared databases.

A secondary requirement of this design was to purchase as little additional hardware and software as possible. Setting up a central server would have required purchasing an expensive computer with a large disk-drive capacity. It would also add complexity to the network and be an extra burden for the network administrator.

A peer-to-peer network relies on all computers connected to the network to act as servers for one another. In a pure peer-to-peer network, each computer may act as a file server, mail server, print server, application server, and so on as well as be a client for each of those services. For example, if one computer on the network has a dot-matrix printer attached to it and another computer has a laser printer attached to it, both computers may act as print servers and print clients. If either computer (or any other computer on the network) needs to send a document to the laser printer, the sending computer becomes a print client, and the computer with the attached laser printer becomes a print server. The same is true for the computer with the attached dot-matrix printer. Each computer may also act as both client and server simultaneously if it sends a document to its attached printer. Because there is no network operating system overseeing the network operations, each computer in a peer-to-peer network must provide portions of a network operating system's capabilities. Some operating systems include capabilities such as file sharing; for other systems, additional capabilities must be added.

The peer-to-peer configuration was selected for further investigation because it seemed simpler and because it seemed unlikely that funds would be available for the purchase of a dedicated server. The primary task was to determine whether all required functions of a network operating system were available either included with or as additions to the four existing operating environments and whether all pieces of hardware and software would work together. It was also important to keep the goal of simplicity constantly in mind. A peer-to-peer system that would require more additional complexity and more administrative overhead than the addition of a central server would not be a step forward. A preliminary investigation into the capabilities of existing operating environments and the availability of network additions suggested that a peer-to-peer network could be made to work.

Standard protocols were selected (as discussed below) on the basis of availability for all four operating environments, technology, and price. Evaluation software and hardware borrowed from vendors were tested to compare additions to each operating system and to confirm that all required hardware and software would operate correctly together. Such testing is an essential step in the design of any network. Although most vendors attempt to ensure

compatibility between their products and standard systems, there are too many variables in any network to guarantee that every piece will work as designed. Most of the effort in this evaluation was spent testing file-sharing software, printer-sharing software and hardware, and their interactions with the most commonly used applications. As a result of this testing, all purchased hardware and software are usable.

The final network configuration is a compromise between central server and peer to peer. Although there is no large central server running a separate network operating system, there are central mail servers (the two UNIX workstations) and centralized print servers (dedicated hardware), and there will probably be a centralized backup server. File sharing and remote-computer access are distributed throughout the network.

ETHERNET AND 10BASE-T

To connect any two pieces of computer equipment, the interface between the pieces of equipment must be defined. A computer interface is usually defined through a set of rules, or protocols, to which all manufacturers agree. There are many interface protocols in use in the computer industry; selecting well-established, consistent protocols throughout a network is particularly critical if equipment is from multiple vendors. Selecting the physical and data-link (hardware) protocol to interconnect the Spokane LAN was the first of several standard-protocol decisions. At the time, three widely used hardware standards were in use: LocalTalk, Token-Ring, and Ethernet.

The Apple LocalTalk interface is built in to all Macintosh computers; selecting LocalTalk saves the cost of buying network interfaces for the Macintosh computers. LocalTalk interface adapters are available for PC-compatible computers. However, connecting the UNIX workstations to a LocalTalk network requires the purchase and installation of bridges between the workstations' native Ethernet and LocalTalk. A benefit of using LocalTalk is the ability to use common telephone wire for the network; in addition, there was some experience at this site in setting up LocalTalk networks. A disadvantage of using LocalTalk is the data-transfer speed. Maximum speed on LocalTalk networks is 230 Kbit/s. LocalTalk networks can be configured as linear (serial) strings of computers, as star patterns, or as complex combinations of linear strings of computers (fig. 1).

Token-Ring network hardware is always configured as a ring or a combination of rings. It is much faster than LocalTalk, having a maximum speed of either 4 or 16 Mbits/s, but requires purchasing interface adapters for all computers on the network, including the UNIX workstations. At the time of this design, the Token-Ring protocol

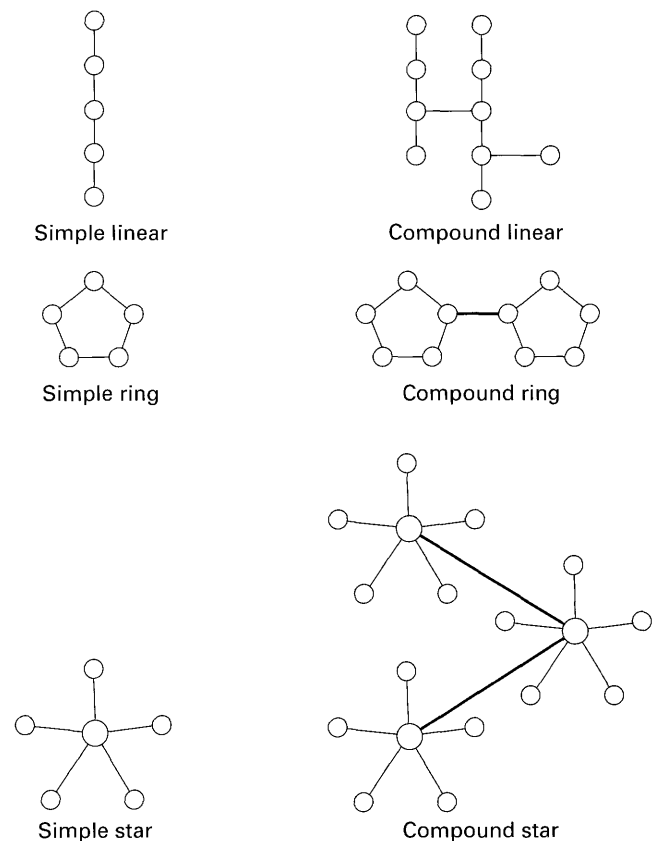


Figure 1. Possible network configurations.

was new and was reputed to be sensitive to wiring quality and interference.

The Ethernet hardware protocol is the oldest of the three considered. It is nearly as fast as the fastest Token-Ring system, having a maximum speed of 10 Mbits/s, and is available from many vendors. The hardware is stable and priced lower than the other two types. To install an Ethernet LAN requires purchasing interface cards for PC compatibles and Macintoshes; UNIX workstations come with Ethernet interfaces installed. Ethernet can be wired in a serial pattern, as a star, or as a combination of serial and star.

The Ethernet protocol was selected primarily because of its widespread use, adequate speed, and availability across all hardware platforms. Ethernet can be wired with thick coaxial cable (so-called thick-net or 10Base-5 wiring), with thin coaxial cable (thin-net or 10Base-2 wiring), with fiber-optic cable (10Base-F wiring), with unshielded, twisted-pair cable (UTP or 10Base-T wiring), or with any combination. Coaxial and fiber-optic cables are more expensive and much more difficult to install but allow longer distances between hosts. For example, thin-net protocols allow as much as 200 m between adjacent hosts. The cable and installation cost is much lower for a 10Base-T system. However, 10Base-T systems require a central

repeater or hub, a star configuration, and line lengths of 100 m or less.

10Base-T systems are always wired in a star pattern having a hub at the center (fig. 1). The star can be expanded by adding additional stars with the hubs connected by either 10Base-T cable or coaxial cable. One advantage of the star configuration is that each host is connected directly to a hub by a single cable. If a cable is faulty, only one host is affected. Similarly, if a host has trouble maintaining its network connection, there is only one cable to troubleshoot. In a linear configuration, each host is dependent on the wiring links of all other hosts between it and the remainder of the network.

TCP/IP AND ETHERTALK

Once the decision was made to use Ethernet and 10Base-T for the physical and data-link (hardware) protocols, a transport protocol was needed. Although many proprietary transport protocols can coexist with Ethernet, the de facto standard is TCP/IP. TCP/IP is also the standard transport protocol in use throughout the Internet. Because of the intended use of the Internet for remote communications and in the interest of keeping the design as simple as possible, TCP/IP is used for all local host-to-host communications. TCP/IP software is a standard part of the UNIX operating system, so no additional software was required for the UNIX workstations. TCP/IP is not a standard part of any desktop operating environments currently in use, but adequate TCP/IP software is available in the public domain for each operating environment.

EtherTalk, Apple's adaptation of the AppleTalk transport protocol for Ethernet hardware, is available on all Macintosh computers having Ethernet interfaces installed. After the network was installed, the EtherTalk protocol was run simultaneously with TCP/IP (the two transport protocols can run simultaneously on the same Ethernet hardware). Macintosh users who have no other need for TCP/IP can use EtherTalk to share files with other Macintosh users. They can also print to a network printer attached to a print server that utilizes the EtherTalk protocol.

NFS AND APPLESHARE

After the transport protocols were selected, applications that would operate across the network were considered. The first application was file sharing. The goal was to provide file sharing across the entire local network for all users. Each computer on the network should be able to act as both a file server and a file-sharing client. With the data owner's permission, it should be possible to make any file or directory of files available on the network to any subset of the local users. Also, it should be possible for any user

having the appropriate permissions, at any computer on the network, to use data as if they were on a local disk.

On UNIX-based systems, Network File System (NFS) is the industry standard protocol for sharing files across a network. Each computer running NFS can allow remote users access to any file, directory, or file system and can provide restrictions to that access by remote computer, user, or group of users. Similarly, NFS allows remote file systems to be mounted to the local file system and used as if the files were on local disks. NFS is in widespread use and is available as commercial software additions for all non-UNIX operating environments.

Most software testing for this network design was directed at the compatibility of PC-compatible-based NFS software with other applications. All software packages tested included TCP/IP software, and most were capable of running both as DOS applications and under Windows. They also included software for remote-system login (*telnet* or equivalent) and for remote-system file transfer (*ftp* or equivalent). Most packages failed to meet the design goals in providing server modules, because most can be used only as clients in the NFS environment. Thus, although remote disks could be mounted and then appear as local disks (NFS client function), local disks could not be made available to other users at remote computers (NFS server function). The same was true of most other software functions; *telnet* and *ftp* client functions were supported, but *telnet* and *ftp* server functions were often not.

Additional problems with most of these NFS packages were use of large amounts of DOS memory and lack of compatibility with critical application software. The ideal solution would have been to find one NFS package that would work for all PC-compatible users. Unfortunately, at the time of this evaluation, no available package met all design goals. However, two packages were found that, between them, satisfied most users' needs. These packages are described in more detail below.

A similar problem exists on Macintosh systems. Apple Filing Protocol (AFP), the native file-sharing protocol for Apple computers, is part of the Macintosh operating system. The AppleShare program, available on all Macintosh computers, is a file-sharing client only. It can be used to mount files from any AFP server. The file-sharing server extension, called File Sharing by Apple, is a part of the System 7 Macintosh operating system. File Sharing allows the Macintosh to be set up as an AFP server. As long as the user wants to share files only with other Macintosh users, AFP provides both client and server file sharing equivalent to NFS. However, if the Macintosh user wants to share files with users on either UNIX or PC-compatible computers, NFS software must be added to the system. NFS software that provides client-only file-sharing capabilities is commercially available for the Macintosh.

Some users do not need the capabilities of NFS file sharing but would like to be able to use other remote

services. For these users, TCP/IP software and client versions of *telnet* and *ftp* are available in the public domain for both PC-compatible and Macintosh computers.

SMTP, BINHEX, AND MIME

Like other network functions, electronic mail service is divided into a client portion (often called a User Agent or UA) and a server portion (often called a Message Transfer Agent or MTA). The mail program that a user sees as his or her mail interface is a User Agent. The Message Transfer Agent is the behind-the-scenes server, usually a daemon, which runs on the mail server and is responsible for e-mail delivery.

The industry-standard UNIX message transfer protocol and the standard used throughout the Internet is the Simple Mail Transfer Protocol (SMTP). SMTP is included with nearly all UNIX systems and can be understood by most proprietary electronic mail systems. Because of its nearly universal acceptance by other systems, SMTP was chosen as the standard for e-mail message transfer. The initial design decision was to set up the two UNIX workstations as mail servers and to use the mail software on each for both the User Agent and the Message Transfer Agent portions of the e-mail system. This arrangement requires each user to log on to a workstation to send or receive e-mail. Because the workstations are always kept running, there is always a mail server (MTA) available to receive e-mail. During the network installation, software was found that allows users to run User Agents on individual desktop computers and thus to send and receive e-mail at their own computers while still having the workstations act as the mail servers (MTA's).

There are several standard protocols for attaching documents to e-mail messages for transport to remote hosts. The oldest protocol in widespread use is probably the `uuencode/uudecode` pair that is used on UNIX systems. This protocol was originally designed for use with `uucp` (UNIX-to-UNIX copy), an early method of file transfer on UNIX systems. `uuencoded` files are still found in many UNIX archives.

BinHex (Binary to Hexadecimal conversion) is the most widespread protocol in use between desktop computers at the time of this writing. It is mostly used for transport of documents and data files between Macintosh computers. However, User Agents capable of encoding and decoding BinHex files are also available for PC-compatible and UNIX computers. These systems are capable of translating each other's data files. Not only can PC-compatible computers exchange data files, but it is also possible to attach a DOS data file to an e-mail message and send it to a Macintosh computer, or vice versa; the file can be read if the receiving application is capable of importing data files from the sending application. BinHex can encode any type of

data file that exists on either DOS or Macintosh computers. It was adopted as a transport protocol because of its widespread use.

The other data-file-transport protocol adopted in this design is Multipurpose Internet Mail Extensions (MIME). MIME is not currently as widely used as BinHex but is potentially more powerful. It uses a more flexible method of identifying the type of data encoded in the file and is capable of identifying not only the usual data types but also audio and video data files. Its data identification method also has more potential for future growth of data types. MIME is available as a function of User Agents that run with all operating environments in current use. It is rapidly becoming a standard on the Internet.

POSTSCRIPT

At the time of the network design, a PostScript-compatible laser printer had recently been installed on the shared PC-compatible port-sharing device. Nearly all users reconfigured their software to print to the new printer through PostScript-compatible drivers. The PostScript page description language offers increased flexibility in printed output design through scalable type and compatibility with nearly all software that is in use on the PC-compatible computers. For these reasons and because the Macintosh computers use PostScript for their printing, PostScript compatibility was adopted as a requirement for all future network printer purchases.

DESIGN SUMMARY

A desire for simplicity and equality of access led to the adoption of a modified peer-to-peer network configuration. The widely used, industry-standard Ethernet hardware protocol using 10Base-T wiring and the TCP/IP transport protocol were selected. The EtherTalk protocol is supported for Macintosh users who have no need for TCP/IP. The NFS protocol was selected on the basis of its availability on all platforms, and AFP is supported for Macintosh users who required file sharing only with other Macintosh users. For electronic mail, SMTP is the standard selected for all mail transport; BinHex and MIME are supported for attaching data files to e-mail messages. Finally, PostScript compatibility is required of all network printers. Throughout the design process, the emphasis was on creating a fully functional LAN as simply as possible.

HARDWARE

The following descriptions of the hardware and software used at this site are intended as concrete examples of one system that works; many other combinations would

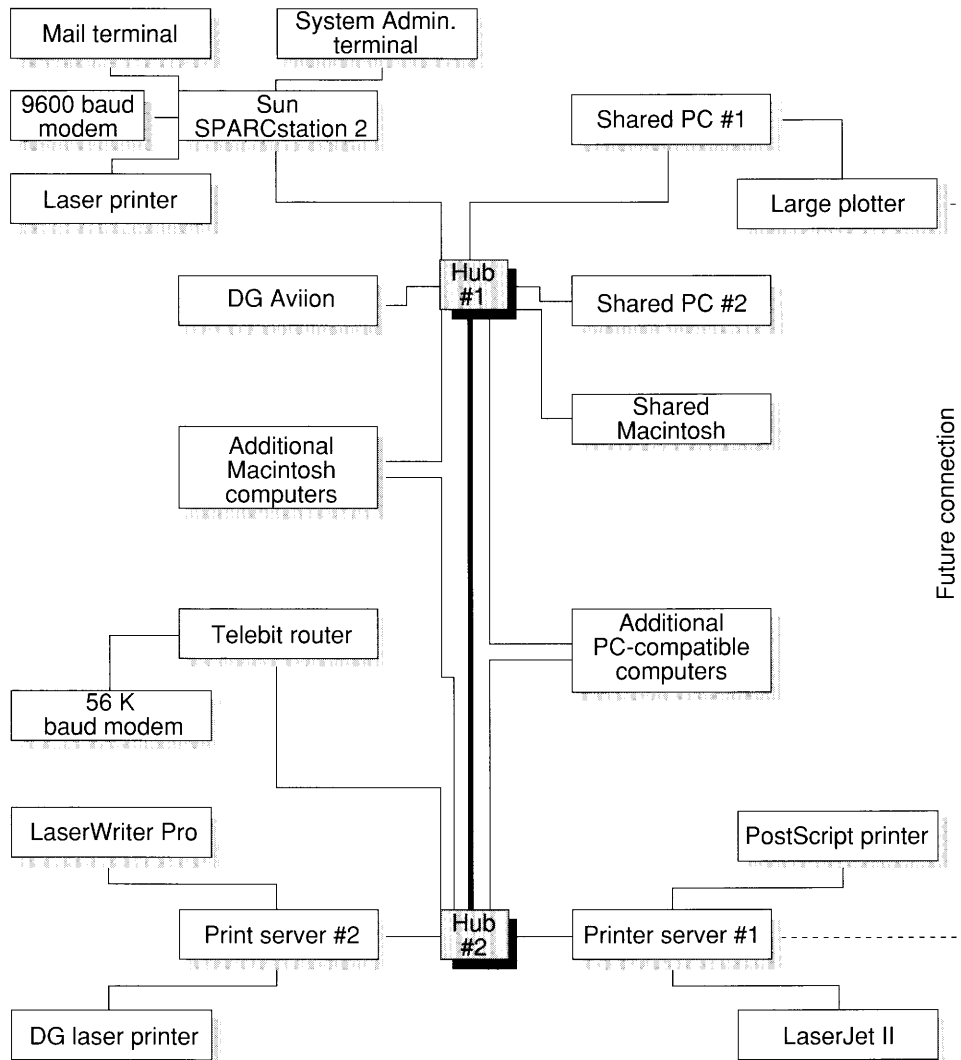


Figure 2. Spokane field office network configuration.

work equally well. Because these components are known to work together, this system provides a point of departure for designing a LAN for similar sites. Figure 2 shows the current physical configuration of the Spokane LAN.

HUBS

The heart of any 10Base-T Ethernet network is the repeater, or hub, which lies at the center of the star pattern (fig. 2). The capacity of a hub is based on the number of satellite hosts that it can support; hubs are available with a wide range of capacities, from special-purpose hubs that support four hosts or fewer to large devices capable of supporting hundreds of hosts. Because hubs can be chained together to increase capacity at any time, it is not necessary to purchase initially all the capacity that will ever be needed. However, the price per supported host generally decreases with increased capacity, so it makes sense to plan

for at least the near-future expected network size. (The Spokane LAN was expected to grow to approximately 20 hosts within a year.) Because of the need to test a small network before committing to a design and because there is a wide selection of hubs available that support 12 to 14 hosts, a 14-host hub was purchased for initial testing. A second, identical hub was added at the time the LAN was installed, and a third hub can be added in the future if the LAN grows beyond the capacity of the first two hubs.

When selecting the hub capacity required for a LAN, the designer must count all hosts that are to be connected to the network. A host, in a network design, is any piece of hardware that has an Ethernet cable connected to it. Each computer to be connected to the network is a host. Some computers have more than one network connection available. If these additional network connections are used, then each additional connection is also counted as a host. Other devices that are not normally considered computers can also be hosts on a network. On the Spokane LAN, shared

printers are connected to the network through dedicated hardware print servers. Each print server is directly connected to the network and is counted as a host. Similarly, routers, bridges, or additional hubs chained to the initial hub are counted as hosts. The initial Spokane LAN design called for 20 hosts, including 16 computers, a print server, a router, and the additional hub (counts as two, one for each end of the connecting cable).

The most common 10Base-T Ethernet hub available has 12 ports for remote hosts. These ports are usually reached either through a single 50-pin RJ-21 connector or through 12 individual RJ-45 modular sockets. The single 50-pin connector is most useful if the hub is to be permanently installed with standard telephone equipment. Individual modular sockets allow the hub to work in a stand-alone mode and offer more flexibility in moving connections. Some hubs allow the option of switching between the single connector and the individual connectors. Besides the 12 remote host ports, many hubs also include one or two non-10Base-T ports—usually an AUI port for connecting thick coaxial cable, a BNC connector for thin coaxial cable, or both. If it is possible to run coaxial cable to one or two of the hosts to be connected, these ports can be used to connect additional hosts to the hub. More commonly, however, these ports are used to connect between hubs or to connect hubs to a preexisting, coaxial-cable network. For this LAN, two hubs were connected through their BNC ports by using a short length of thin-net cable; thus, all 24 10Base-T ports were freed for other hosts.

Hubs were evaluated for this project on the basis of their compatibility with Ethernet and 10Base-T standards, the type and number of host connections available, the warranty offered, and the price per host connection. The hub selected was an Asanté model 10T that offered a total of 14 host connections, including 12 10Base-T ports, one BNC port, and one AUI port. The Asanté also comes with interchangeable back planes, so that it can be changed from individual RJ-45 modular sockets to a 50-pin RJ-21 connector. After initial testing proved that the LAN would operate as planned, a second model 10T hub was purchased.

NETWORK INTERFACE ADAPTERS

After the Ethernet hub, the next most important piece of hardware to consider is at the other end of the Ethernet cable, the Network Interface Card (NIC). Each host on the network must have an interface to the network. For some hosts, such as print servers and routers, the interface is installed at the factory. For other hosts, including most desktop computers, the interface must be added as an expansion card. The card usually plugs into a computer's expansion slot. A socket for connecting the computer to the network is on the exterior of the NIC. For a 10Base-T network, it is important that all NIC's have an RJ-45 external

socket. Although adapters are available that allow other types of NIC's to be connected to a 10Base-T network, using them adds to network expense and complication.

Unfortunately, it is not possible to use the same NIC in every computer on a mixed-vendor network. PC-compatible computers and Macintoshes have different types of expansion slots and therefore require different NIC's. In addition, there are differences in expansion slots within the range of both the PC compatibles and the Macintoshes currently in use at this site. The best compromise that could be achieved was to try to locate a vendor carrying a line of NIC's that could be used for all computers. In practice, two vendors were required—one with a line of NIC's suitable for PC-compatible computers and a different vendor with a line suitable for Macintosh computers.

Besides the hardware connection to the Ethernet, the NIC must also provide a software interface to all applications that need access to network resources. The network driver that usually comes with the NIC provides this interface. The driver must communicate with the NIC hardware and the lowest level of network software—here the TCP/IP and AppleTalk software. Some network software includes drivers for NIC's from major vendors. In practice, then, the NIC's must be evaluated while the network software is being evaluated. It is critical that these two links in the network chain be compatible; if they are not, no other applications can use network resources. It is well worth the time required to obtain evaluation copies of both hardware and software to be sure that they are compatible before investing in multiple copies of each.

There are several other factors to consider when evaluating PC-compatible NIC's. PC-compatible computers in use include both 8-bit bus and 16-bit bus ISA (Industry Standard Architecture) expansion slots. Although NIC's designed for the 8-bit bus will operate correctly in computers with a 16-bit bus, the advantage of the 16-bit bus—data transfer speed—would be lost. NIC's designed for the 16-bit bus will not work in an 8-bit bus computer. A second performance issue for PC-compatible NIC's is the amount of memory on the card and how that memory is accessed. Three types of NIC memory access were in use at the time of this evaluation: shared memory, Direct Memory Access (DMA) memory, and input/output (I/O) port memory access. Shared-memory access provides faster response than either DMA or I/O port access.

NIC's must communicate with the host computer through three interfaces. First, the NIC needs to be able to interrupt the processor through an Interrupt Request (IRQ) line. There are not many IRQ's available on PC compatibles, and some IRQ's are already in use by various other peripherals attached to the system (disk drives, communication ports, pointing devices, and so on). The NIC also needs to send and receive commands through one or more of the computer's I/O ports. Finally, the NIC may require the use of part of the memory address space as

buffer space. Ideally, the NIC could use any available IRQ line, any available I/O ports, and any portion of available memory. The more restrictive the NIC is, the harder it will be to standardize on its use for all systems. In addition, NIC's that allow their configuration to be changed by software after they are installed are much easier to work with than those that require the system case to be opened so that hardware switches can be changed.

For the Spokane LAN, the Western Digital EtherCard line of PC-compatible NIC's was adopted as a site standard. The EtherCard Plus 10-T is used for 8-bit bus PC-compatible computers, and the EtherCard Plus Elite 16-T is used for the 16-bit bus computers. Since our original purchases, Western Digital has sold their EtherCard line of products to Standard Microsystems Corporation (SMC); SMC still sells the cards under the same names. The model 10-T NIC contains 8 KB of shared memory, and the Elite 16-T contains 16 KB of shared memory. Both NIC's have two external ports—an AUI jack and an RJ-45 socket. Both cards have external LED's that show the network connection status and include software drivers that are compatible with the network applications in use.

A similar evaluation was undertaken to find a single line of NIC's that would be compatible with all Macintosh models in use at the site. Issues of the amount of memory on the card and the type of memory access are the same as they are for PC-compatible NIC's. The software driver compatibility issue is also the same. Macintosh NIC's are easier to install than PC-compatible NIC's because interrupts, ports, and memory allocations are configured automatically.

For this LAN, the Asanté line of MacCon+ Ethernet NIC's was adopted as a site standard. The MacCon+ IIET is used for all Macintosh II computers having the NuBus expansion bus. This NIC has a 32-bit interface, 16 to 64 KB of shared memory, and both AUI and RJ-45 external ports. Some newer Macintosh systems, such as the Macintosh LCII and the Centra 610, use the Processor Direct Slot (PDS) expansion bus. For these systems, a similar expansion card is also available in the Asanté line. Like the PC-compatible NIC's, these cards have external network status indicators and come with software drivers that are compatible with the network application in use. Finally, there are systems such as the Apple PowerBook notebook computer that have no internal expansion capability. These systems are connected to the Ethernet through NIC's that attach to their SCSI ports.

Other hosts on the network (including the UNIX workstations, the print servers, and the router) came with network interfaces installed at the factory. Some of these interfaces were equipped with RJ-45 sockets for direct connection to the network; others have only AUI connectors for connection to a thick coaxial cable network. For this LAN, AUI connectors were fitted with Allied Telesis CentreCOM 210TS transceivers with external LED network status indicators; each is contained in a small box that attaches

directly to an AUI connector and has an RJ-45 socket on the opposite end.

WIRING

Once hubs and NIC's have been selected, the next requirement is cabling. Five hosts were cabled together for the initial Spokane LAN testing. The hosts included the Sun workstation, two PC-compatible computers, one Macintosh computer, and the hub. All equipment was in two adjacent rooms, so that cable could be strung along the floor for temporary connections. The cable used was standard unshielded, twisted-pair Ethernet cable with four color-coded pairs of wires. RJ-45 modular plugs were installed by using an industrial-strength hand crimper. Inexpensive plastic crimpers are available but are not recommended; significant, consistent pressure is required to make reliable connections. Although four-pair wire is the Ethernet standard, only two pairs of wire are used for each connection. No problems were encountered with any temporary cabling installed for the testing phase.

Before the LAN installation, the office staff moved into newly renovated office space. Because a new telephone system was to be installed, there was an opportunity to have the network cable strung throughout the office space along with the new telephone cable. After discussions with several communications experts (mainly Skip Dutro, written commun., 1993), we decided that standard, Level 3, 24-gauge, four-pair telephone cable could be used for the 10Base-T Ethernet installation. This decision allowed the telephone system designers to request network outlets along with the normal telephone outlets and allowed the telephone installers to string cable without regard to the type of outlet involved. The incremental cost of installing additional outlets was small enough that network outlets were placed in every office. Several outlets were installed in large offices and those that were known to need multiple connections. Several outlets were also installed in each wall of computer laboratory space. Using standard telephone cable and placing many outlets provided increased flexibility for future expansion and for shifting of personnel and computers between offices.

All network cabling was routed back to a central telephone closet that is accessible to the network administrator. The telephone installation team brought the wiring out to a series of blocks with a separate RJ-45 socket for each cable. At that point, the network administrator installed two 10Base-T hubs on the closet wall and connected them with a short length of thin-net coaxial cable. The cable is approximately 2 m of RG-58-U coaxial cable connected to each hub through a BNC T-connector. The empty leg of each T-connector contains a 50-ohm terminator. Once the hubs were connected to each other, short patch cords were made with RJ-45 modular plugs on each end. These patch cords

connect the RJ-45 sockets leading to each individual office to the hubs.

The first step in installing the LAN in the new space was to reconnect the test-phase computers by using the newly installed wiring and the hubs in the telephone closet. Several problems encountered were ultimately diagnosed as faulty Ethernet patch cords, either at the hub or between the computer and the wall outlet. An inexpensive Ethernet cable tester proved indispensable in solving these problems. The tester is used to check for continuity through the cable and for correct pin use in the RJ-45 modular plug at each end. The same tests could be made easily with any continuity tester; however, a dedicated tester makes the job faster and less error prone.

This experience also points out the advantage of adopting a step-by-step approach to installing the LAN. It would have been more difficult to track down the faulty cables if more new LAN segments had been installed before the testing. For the remainder of the LAN installation, every patch cable was tested before use. Repairs to faulty cables were easily made by replacing one of the RJ-45 plugs. The cable tester was also useful for tracking down a faulty cable between the telephone closet and one of the offices. The ease with which this faulty cable was identified and replaced confirmed the value of the star LAN configuration.

ROUTER AND MODEM

To this point in the hardware discussion, the emphasis has been on the local network. To provide the desired remote services, access to the Internet was also needed. Several options for Internet access were evaluated. Much of the evaluation of options and advice on hardware acquisition came from the USGS's Information Systems Division (ISD) personnel (Jackie Soares, written and oral commun., 1993). Options evaluated included a local (within 30 mi) dedicated telephone line to a non-USGS Internet site, a long-distance dedicated telephone line to a USGS Internet site, and dial-up telephone access.

At the time of this evaluation, the closest Internet site that could be used as an access point was a non-USGS site approximately 25 mi from the office location. The primary advantage of using a dedicated telephone line is speed; a dedicated T-1 line transmits data at a rate of 1.54 Mbits/s. The primary disadvantage is cost; the line is always up and is charged by time and per mile. Although it would have been less expensive to lease a dedicated line for the 25-mi local connection than it would have been for the long-distance connection, the initial expense would have been higher because of the need to upgrade equipment at the local Internet site. A long-distance dedicated-line option was ruled out because of recurring cost. As the volume of data transmitted through the link increases, the long-

distance dedicated line may become more cost effective at some point.

Our choice for a link to the Internet was a pair of high-speed modems having a dial-up telephone connection. After the initial hardware purchase, costs accrue only when the data link is used. The hardware selected was in use at the time of this evaluation on a successful similar link between the USGS's Hawaiian Volcano Observatory and ISD's Menlo Park, Calif., computing center. Because the equipment was already in place in California, a link was created between the Spokane office and the same computing center in California. The modems selected (one for each end of the link) were Telebit Netblazers (model T3000), which operate at 56 Kbit/s.

A router was also needed to drive the modem and to provide a link between the modem and the local network. The router is a small desktop computer having an Ethernet interface and a serial port, which is dedicated to the function of sending and receiving Ethernet packets through the modem. It examines all packets on the local network and routes those that are not addressed to local hosts to the Internet through the modem. The router also places all incoming messages on the local network. The router handles all commands to the modem to dial the phone for outgoing messages and to answer the phone for incoming messages. In addition, to prevent the expense of many short telephone calls, the router can be programmed to hold the connection for a fixed minimum time. The router selected for this network was a Telebit Netblazer (model ST), which can be expanded to include a second Ethernet interface if necessary. It can also make use of a second high-speed modem. The dual-modem mode effectively doubles the data transmission rate, because the router can transfer different portions of large data files over each modem simultaneously. This mode also requires two modems and two phone lines at the other end of the link.

PRINTER AND PLOTTER SERVERS

To provide laser-quality printer service and large-format plotter service to all local users, a method of connecting printers and the plotter to the LAN was needed. At the time of the LAN design, there were three laser printers and one large-format plotter to connect to the network. One printer had been used by a small group of Macintosh users connected through an AppleTalk network by means of PhoneNet hardware. The other two laser printers and the plotter had been used by a small group of PC-compatible users connected through a smart port-sharing device. A design requirement was to continue providing existing services to the current users while expanding access to these devices to all users.

There are several possible methods of connecting printers and plotters to Ethernet networks. One method is to

purchase network interface expansion devices for each printer/plotter. Ethernet expansion devices and the associated TCP/IP software are available for many, but not all, printers. This method would probably provide the fastest possible interface because each device would be directly connected to the network. Ethernet expansion devices were not available for all of our printers and would have been more expensive than other methods.

A second possibility is to use existing computers as print servers. Some TCP/IP and NFS software packages that were tested included print server modules. To use a print server module on a desktop computer, the printer is first connected to the computer on a local serial or parallel port. Then the print server software is started, and the printer's name and location are "published" on the network. Users on other computers can then direct their printing to that printer through the computer acting as a print server. This technique is probably the least expensive but has some serious performance drawbacks. First, the print-server computer must be turned on and running the server software any time that a user needs access to the printer. Although some print-server software evaluated could run in the background so that the server could be used for other tasks, all server software used significant portions of memory and much of the system's processing power while transferring files to the printer. Because of the performance decrease and the need for constant access, it would probably be necessary to dedicate a computer to act as a print server for this technique to be effective. Because the print server does not require much processor speed or large disk storage, this method of providing printer service may be the most cost effective for sites that have unused, out-of-date computers available.

A technique that could be used for the LaserWriter (and any other printer having a built-in LocalTalk interface) would be to link the LocalTalk network of such devices to the Ethernet through a gateway or bridge. Gateways and bridges are small special-purpose computers having at least two network interfaces. Their function is to translate data packets from one network protocol to another and then retransmit the data on the second network. The difference between the two is that bridges retransmit all data received on one network to the other network; gateways check the destination of the data packet and only retransmit those whose destination is on the other network. Using a gateway properly can reduce the volume of network traffic on both networks. The drawback of this technique for the Spokane LAN was primarily cost. Because there was only a single printer having a LocalTalk interface installed at the time of this evaluation, adding a bridge or gateway to the network would have been more expensive than other solutions. If a site has many LocalTalk-ready devices to connect, it may be cost effective to install a bridge or gateway.

The solution selected for Spokane was to install dedicated print servers. A dedicated print server is a small box that contains an Ethernet interface and one or more ports for

peripheral devices. The box may also contain a small special-purpose computer with memory for buffering print files. A print server is connected to the Ethernet just like any other host. It has a unique network address and host name. The printers and (or) plotters are connected to its peripheral ports. Dedicated print servers are available having one to four serial, parallel, or LocalTalk ports. To provide for expansion beyond the four existing devices, two print servers were added to the network. The initial design called for splitting the output devices evenly between two identical servers, all users being able to send output to any device.

The specifications of all dedicated print servers for which data could be obtained were studied, and evaluation units were obtained from five vendors. All five units were extensively tested with the small test network and with several types of DOS, Windows, and Macintosh software. Several factors were considered in the evaluation, the most important being compatibility with existing hardware and software. Because we were standardizing on PostScript-compatible printers, the print server had to be PostScript compatible and able to translate non-PostScript ASCII files into the PostScript language for printing, if necessary. This requirement allowed the use of PostScript printers that did not automatically translate ASCII files. Two servers were unable to make such translations. A third server was eliminated from the evaluation because it was priced well above the others. The final two contenders passed all compatibility tests from the PC-compatible computers and from the UNIX computers. However, both servers displayed problems when attempting to print large graphic files from the Macintosh, and neither included an EtherTalk interface. Thus, it was not possible to obtain a single print server that was completely satisfactory for both Macintosh and PC-compatible users. We decided to restructure the design so that the two printers used by the PC-compatible users and the large-format plotter were attached to one print server; a separate solution was sought to connect the Apple LaserWriter to the network.

In the final evaluation of print servers to use with the PC-compatible and UNIX computers, the following capabilities were considered significant: (1) number and type (serial or parallel) of device ports, (2) cost per device port, (3) availability of line printer daemon (*lpd*) software running on the server, (4) password protection, (5) availability of UNIX-to-DOS, DOS-to-UNIX, and ASCII-to-PostScript translation filters, (6) internal job redirection, and (7) performance. The first two items are obvious, but the others require explanation. An *lpd* server that runs on the print server allows computers that do not have print-server software, such as DOS computers, to send print files directly to the print server. Without an *lpd* daemon on the print server, the print-server task must be split between the dedicated device and another computer that queues the print job before sending it to the dedicated server. For the test

network, the Sun workstation was used for printer queuing. The disadvantage of this system is that the workstation must be online for other computers to use the print services. An advantage of this system is that all workstation print-queuing controls are available to users and system administrators. Because of the differences between the end-of-line characters stored in text files on UNIX and DOS systems, either a UNIX-to-DOS or a DOS-to-UNIX filter is required to print from both computer types to the same printer. A useful feature found on some print servers is the ability to automatically shift print jobs from a busy printer to one that is not busy. If the printers are similar (for example, both PostScript compatible) and are located in the same general area, this feature can greatly reduce print turnaround time. Finally, the print servers were evaluated on the basis of their data-transfer speed. Most servers have maximum transfer rates of 100 to 125 K characters per second (cps) through their parallel ports and 38.4 or 76.8 Kbaud through their serial ports.

The print server selected for the PC-compatible and UNIX systems was an AXIS model AX-5 having two parallel ports and two serial ports. The two laser printers are attached to the parallel ports, one serial port is used for the large plotter, and the remaining serial port is available for future expansion. This unit does not have built-in *lpd* server software, so the Sun workstation is still used for print-job queuing. The server does have the required filters and can operate at 125 K cps through the parallel ports and 76.8 Kbaud through the serial ports. The AX-5 also includes the busy-printer redirection feature. In addition, it allows eight logical printer definitions to be distributed among the four physical devices. Each logical printer can be configured differently; if more than one set of printer features and filters is required to print differing data files to the same printer, more than one logical printer can be defined for each physical device.

A suitable Macintosh-compatible print server was also identified—the Dayna EtherPrint-T Plus. Although this device is marketed as a print server, it is closer in function to a gateway. The server has connections for both Ethernet and LocalTalk networks and can support four LocalTalk devices. The server does no filtering or reformatting of data files, nor does it contain memory for spooling printer files. All print files are assumed to be in printer-ready format, and print-file spooling is done on the originating computer. The printers attached to the Dayna print server appear in the Chooser window of Macintoshes on the network, just as if the printers were connected directly through a LocalTalk cable. A Dayna EtherPrint-T Plus print server was purchased for the network, and the LaserWriter printer was attached to it. Later, a second laser printer from Data General was attached to the Dayna server.

PRINTERS

Initially, four Macintosh users were using the LaserWriter printer for all their printing. By the time the network was installed, seven users were using two printers on the Dayna server for all their printing. Because the printers were causing significant print delays, particularly with large graphic files, a faster Apple LaserWriter Pro (network) printer was substituted for the original LaserWriter on the network.

When the PostScript-compatible laser printer became available to the PC-compatible users, Hewlett-Packard LaserJet II use nearly ceased. The only use still made of the LaserJet was by users who ran programs that did not have PostScript drivers. As many users migrated either partially or completely to Windows-based applications, LaserJet use further declined. To get more use out of the investment in the LaserJet printer, it was decided to convert it to PostScript use. Initially, a PostScript card was installed in one of the printer's font cartridge slots. This conversion worked well, allowing users to send either PostScript or Hewlett-Packard PCL-4 files to the printer. Unfortunately, the printer had to be sent a set of instructions as a header to the print file to change the printer from one file format to the other. The printer was also several times slower than the dedicated PostScript-compatible printer. As use continued to drop, it became apparent that the only way to achieve an appropriate level of printer use was to convert it to a full-function PostScript printer with a fast PostScript interpreter. Installing a combination PostScript emulation cartridge and a RISC-based accelerator expansion board in the printer converted it to a high-speed PostScript printer that could share the printing load with the original PostScript-compatible printer.

UPS

Frequent power outages raised a concern about losing all network services if the power were to go off, even briefly, in the computer room. Because all PC-compatible and UNIX print services and all mail services were routed through the Sun workstation and because all remote services must pass through the router and high-speed modem, those devices were identified as requiring power protection. Two Uninterruptable Power Supplies (UPS) were purchased. One provides approximately one-half hour of power protection to the critical portions of the Sun workstation (computer and storage devices), and a second, smaller UPS provides approximately one-half hour of power protection to the router and the 56-Kbit/s modem.

SOFTWARE

The reason for designing, testing, and installing network hardware is to provide better services to the users. To use these network services, most application programs require one or more layers of interface software. In addition, software modules must be added to computers that act as network servers.

TCP/IP AND APPLE TALK

The base layer of software required by all applications is the network transport protocol software. For this installation, the standard is TCP/IP, which is a combination of two protocol standards. IP is the Internet Protocol, which specifies how data packets are to be constructed and addressed for transmission on the Internet. TCP is the Transmission Control Protocol, which specifies how files and messages are divided into data packets for transmission by IP and then reassembled at the receiving end. Combined, they have become a standard base layer for data movement over the Internet. Other applications that control functions such as the formatting of electronic mail or the transfer of data files operate on top of the TCP/IP layer.

Because TCP/IP is an Internet standard, versions are available, both commercially and in the public domain, for nearly any computer platform. Several TCP/IP packages were used during the testing of NFS software (see below); all worked adequately. Each Network File System (NFS) package that was tested included TCP/IP. If NFS software is used, the version of TCP/IP that comes with it should be adopted as the network standard. If NFS software is not required, then a single TCP/IP package should be adopted for each computer platform. A widely used public domain TCP/IP package is available on the network from NCSA. Several other network applications are usually included with TCP/IP. At least three applications are important to gain full access to the Internet: *ping*, *ftp*, and *telnet*.

ping is more commonly used by network administrators than by users, but it is a necessity for setting up an Ethernet network. *ping* is a simple utility that transmits an IP request to a remote host asking if the remote host is available. If the network is intact, the remote host is on, and the remote network interface is operating correctly, it returns an answer indicating that all is well. After adding a new host to a network, most network administrators then "ping" the new host from an established host. A successful test shows a functioning network connection and functioning NIC.

File-transfer software (*ftp*) is an everyday Internet workhorse. *ftp* is often disguised with an easy-to-use shell and may go by other names on different systems, but a version of *ftp* is found under the user interface of nearly all file-transfer software. *ftp* programs allow network users to

transfer data files of any kind between computers on the Internet without having to have an account on the remote computer. System administrators who have information to share with the Internet community place that information in special portions of their computer's storage space that are then made accessible to the public. When a user wishes to copy a data file or files from a remote system to the local system, the user starts up an *ftp* client program on the local computer. The *ftp* client contacts the remote computer through the network and exchanges information with an *ftp* server running on the remote computer. The user then selects the data file(s) to be copied, and the two *ftp* programs complete the copying process.

A third important utility for full Internet access is a *telnet* program. *telnet* probably comes in even more disguises than *ftp*, but all work the same way. *telnet* and its relatives allow users to work on a remote computer just as if it were a local computer. This class of software is also commonly called terminal emulation software because it is used to make a local computer act as if it were a terminal attached to a remote computer. The user requests a *telnet* connection to a remote computer, and the program then creates a screen on the local computer that looks like the terminal being emulated. The program connects to the remote computer through the network, and the user sees the normal login prompt for the remote computer. If the user has a current account on the remote computer, he or she may then log in and run programs as if the remote computer were local. The simplest *telnet* programs are usually able to emulate only a standard ASCII terminal; the most common emulation is of a VT-100 terminal. These programs are able to display only text on the emulation screen. More sophisticated terminal emulation programs can emulate several types of terminals, including one or more standard graphic terminals such as the Tektronix 4100 and 4200 series. Using an emulator in graphics mode requires a high-speed network connection because the volume of data required to continuously redraw a graphics screen is large.

There are other network protocols in use on the Internet; the most common, other than TCP/IP, is the UDP/IP combination. UDP (User Datagram Protocol) is often substituted for TCP by applications that send short messages across the network. If the message is short enough to fit within one IP data packet and if the application does not need to guarantee that the packet arrives at the destination, UDP is a faster and more efficient protocol than TCP. Normally, users and network builders need not worry about UDP; it is typically included with TCP/IP software and used automatically when appropriate.

A concern during the design of the Spokane network was whether TCP/IP and AppleTalk could coexist on the same network. Tests were conducted to determine if Macintosh users could use either TCP/IP or AppleTalk for printing and file sharing. Large printer files were sent simultaneously to the LaserWriter from a Macintosh using Apple-

Talk and to another laser printer from a PC compatible using TCP/IP. No problems were encountered during the tests or during subsequent daily operations.

NFS AND APPLESHARE

Standards were adopted for file-sharing protocols based on software that was in use at the time of the network design. Because NFS software is a standard part of most UNIX operating systems and because there is no standard file-sharing protocol for PC-compatible computers, NFS was adopted as the primary standard for all file sharing. Because file sharing between Macintosh computers was already being done with AppleShare plus File Sharing (AFP), it was adopted as a second standard for Macintosh-to-Macintosh file sharing.

PC-NFS

Selecting a standard NFS software package for the PC-compatible computers required the most extensive testing of the design process. Complicating the process was the need for compatibility with all user applications and the desire to support the same software package under both DOS and Windows environments. File-sharing software should be transparent to the user. Once the user has selected part of a remote file system to mount as a local drive, all normal operating-system utilities and user applications should treat the remote files as if they were local. This process is simple when the two computers involved are running the same operating system but becomes more complicated when the operating systems are different. Operating systems generally differ in their rules for naming files and in the methods used to store files. For example, Macintosh computers store all data files in two separate parts—a data fork and a resource fork—whereas UNIX and DOS computers store data files as single files. Also, the Macintosh operating system, UNIX, and DOS all have different rules for assigning names to data files, DOS being the most restrictive. File-sharing software must translate between these differences with as little impact on users and application programs as possible.

The evaluation process used for NFS software was similar to that used for print servers. Literature was obtained from as many vendors as possible, and evaluation copies were requested from eight whose products seemed most likely to meet the requirements. Of those eight, four were eliminated for various incompatibility problems that were apparent from reading the detailed software documentation. The remaining four were tested for compatibility with a representative suite of current user applications.

Factors that were important for this evaluation included (1) ease of installation and configuration, (2) use from both DOS and Windows environments, (3) inclusion

of both client and server modules for at least *ftp* and NFS, (4) remote-printing client module, (5) amount and type of memory required, (6) availability of site licensing and cost per user, and (7) compatibility with critical applications. One critical application was ArcView. For all four packages tested, all normal DOS and Windows programs were equivalent—that is, if one DOS program worked with the file-sharing and print-sharing modules, then all other DOS applications also worked correctly. The same principle was true for Windows applications.

However, ArcView for Windows computers was different. The primary user application that runs on the two UNIX workstations at this site is Arc/Info from Environmental Systems Research Institute (ESRI). At the time of this evaluation, a new product that had just been released by ESRI allowed users on remote Windows computers to manipulate Arc/Info data files and display the results on the local screen. This program, ArcView (version 1), runs under the Windows environment. (Although ArcView is also available for UNIX computers, the following discussion refers to the Windows version only.) ArcView provides easy access to Arc/Info data for users who are not expert at running Arc/Info and who are not familiar with the UNIX environment. ArcView makes direct network connections to Arc/Info data files that reside on remote UNIX and DOS computers in much the same fashion as other Windows programs. The problem that was uncovered by this testing was that ArcView had built-in expectations about the way remote data-file names would be translated from the remote operating system to DOS. Sun's PC-NFS, the NFS package that was used by ESRI during the creation of ArcView (version 1), was the only NFS system tested that met ArcView's requirements.

Unfortunately, PC-NFS did not meet several other requirements. In particular, it did not include an NFS server module. As of version 5.0, PC-NFS included a DOS *ftp* server module. The *ftp* server can be configured as a DOS background process under Windows but is not a true Windows utility. PC-NFS also requires more memory than several other packages tested. If all modules are loaded, including the module required for using PC-NFS with Windows, PC-NFS uses more than 150 KB of DOS memory. Without the Windows module, it uses more than 100 KB. It works in both DOS and Windows environments, includes a remote printing module, is moderately easy to install and maintain, offers site licensing, and is the only package compatible with all tested user applications. Although PC-NFS was not the highest-rated NFS package in these tests, it was adopted as the standard package for users who foresaw a need to use ArcView and for those who operated primarily in the DOS environment.

At the time of the Spokane network installation, nearly all PC compatibles were running PC-NFS (version 5.0) without serious problems. Because PC-NFS lacks an NFS server module, PC-compatible users are not able to share

files directly between PC-compatible systems. All file transfers between PC compatibles must use a file-transfer system such as *ftp*, and file sharing is available only for files stored on the UNIX machines. As newer versions of ArcView and various NFS packages become available, these conditions may improve.

CHAMELEONNFS

A second NFS package, ChameleonNFS (version 3.10) by NetManage, had the highest rating in this evaluation for users who were willing to run all network applications in the Windows environment and did not need access to ArcView. Like all other systems tested except PC-NFS, ArcView did not operate correctly with ChameleonNFS. Chameleon is a Windows-only package; it does not include any DOS modules. Because it does not run in the DOS environment, it uses little DOS memory (less than 10 KB). The remainder of the program is contained in Windows DLL files that are loaded into memory only as they are needed. Another advantage of ChameleonNFS is that it includes both client and server modules for *ftp*, NFS, printing, and mail. ChameleonNFS was the easiest package tested to install and configure; it also had the most straightforward user interface.

NFS SHARE

As mentioned above, AFP is used for file sharing between Macintosh computers, and remote printing is handled by an Ethernet-to-LocalTalk print server (gateway) and the print services included with the Macintosh operating system. Macintosh users also need file sharing with non-Macintosh computers. To accomplish this file sharing, the Macintosh also needs to add an NFS module to its operating system. Because ArcView is not yet available for the Macintosh, the testing program was simpler. At the time of this evaluation, NFS Share (version 1.3) by InterCon was available for the Macintosh. NFS Share makes use of System 7 access privilege and alias capabilities and is entirely satisfactory for file sharing between Macintosh computers and UNIX workstations. Because neither the PC compatibles nor the Macintoshes currently run NFS servers, there was no testing of file sharing directly between the Macintoshes and the PC compatibles.

ELECTRONIC MAIL

An important network design goal was to provide electronic mail (e-mail) access for all staff members. The first requirement for e-mail service is to have a computer that is always on to operate as a mail server. If each desktop computer is used as a mail server for its owner, mail service is erratic at best. Outgoing e-mail could be handled adequately by a desktop mail server, but the server would often be unavailable for incoming e-mail. Some mail systems make

several attempts to deliver mail and then discard it if the system is unavailable; other mail systems return the message to the sender if the receiving mail server is unavailable. In addition, some users work at more than one computer, and some computers have several users. Keeping track of where each user's mail should be stored would be difficult if each computer acted as a mail server. To avoid these problems, several computers that are nearly always on are selected to be mail servers and to run Message Transfer Agents for everyone on the staff. For the Spokane network, the two UNIX workstations are used as mail servers, and the staff is divided between the two workstations, so that each user has only one mail server, to which that user's e-mail address directs all mail. A benefit of communicating through e-mail is that the recipient does not have to be available when the message is sent. As long as the mail server is available when the message arrives, the message is stored in the recipient's mailbox until he or she is available to read it. If the local network is connected to other networks through a dial-up connection, as the Spokane network is, outgoing mail can be held at the central mail server in a queue. The mail system can be programmed to periodically make the dial-up connection and send all queued mail with one telephone call instead of making a separate call each time a message is sent.

There are several e-mail access methods in use on the network. Users who do most of their computer work on a UNIX workstation use the normal UNIX e-mail facilities, including the `mail` User Agent program for those who work from a standard UNIX shell and several other e-mail applications for users who work from a UNIX windowing system. To assure that all users have access to their e-mail, every staff member has an account on one of the workstations. To accommodate users who have either no desktop computer or a desktop computer that is not yet connected to the network, an ASCII terminal has been attached to a UNIX workstation and placed in a common-use area. Any staff member can use this mail terminal to log in to the workstation to send or receive e-mail without interrupting those who are using the workstation for other tasks.

Users who have desktop computers connected to the network have two additional options for sending and receiving mail. The first option is similar to using the shared mail terminal, except that the users can remain at their desktop computers. By using a terminal emulation program to log in to the mail server, users can reach the normal UNIX mail User Agent from their own office. Users' mailboxes are still maintained on the mail server.

A second option for network users is to obtain a User Agent program to run on their desktop computers and interact directly with the UNIX Message Transfer Agent on the mail server. The major advantage of this approach is that users need not learn anything about the UNIX system to have access to e-mail. All e-mail access is through a program that uses the desktop computers' native interfaces.

Although the program may be new to users, the interface should be familiar enough that little training is required to become a competent e-mail user.

Eudora.—Public-domain e-mail User Agent software packages were tried on both Macintosh and PC-compatible computers. Because a public domain User Agent proved satisfactory for both Macintosh and Windows computer users, no commercial e-mail packages were evaluated. The e-mail system that is used is Eudora by QualComm. It is available for both Macintosh (version 1.4) and Windows (version 1.4b18) environments. In addition to the public-domain version of Eudora, there is also a commercial version available that includes more advanced features. Standardizing on a single program that is available for both Macintosh and Windows computers makes the teaching and maintaining of e-mail systems significantly simpler. Eudora is easy to install and configure under both operating environments and offers a consistent, user-friendly interface that is easy to learn.

Eudora also provides a partial answer to another network design goal—document transfer. It is partial because PC-Eudora is only available for the Windows environment; DOS users do not have mail-attachment capabilities and must send documents using file-transfer techniques (see *ftp* above). Eudora has the capability of attaching documents (or other data files) to an e-mail message and encoding and decoding the document using BinHex or MIME. BinHex is in common use at other USGS offices in this region; therefore, we have had more experience using BinHex to attach documents. With BinHex, not only can users transfer files of any type between computers running the same operating system, but they can also transfer files between computers running different operating systems. The only requirement is that the receiving system have some application that understands the file format. A simple example is the ability to transfer a document created in a word processor on a Macintosh to a remote Macintosh that has the same word processor. A more interesting example is the ability to transfer a document created by one word processor on a PC compatible to a remote Macintosh that runs a different word processor. If the receiving computer's word processor can open a file formatted for the sending word processor, the system will work. Of course, the encoding and decoding are not confined to word-processor documents; any data file type can be encoded, sent, and decoded at the receiving end. The file can be a document, a graphic file, a program, a video or audio segment, and so on. Many UNIX User Agents can also transmit and receive BinHex and MIME attachments.

UNIX DAEMONS

To this point, discussion of software running on the UNIX systems has been confined to utilities that are

normally included with the UNIX operating system. There are several other utilities that must be present on the UNIX systems to allow the software described above to operate. Most of these utilities are installed on the UNIX system as daemons. A daemon, in the UNIX world, is a small program that runs continuously as a background task. Each daemon is usually started at the time the system is initialized or booted. After the daemon initializes itself, it remains in the background waiting for a request for its services. When a request occurs, the daemon may handle the request itself, or it may start up a larger program, hand the request to the larger program, and then go back to waiting for another request.

lpd

The line printer daemon, *lpd*, is normally included with UNIX systems. *lpd* watches the storage areas where print files are spooled. Whenever a file is copied to a spooling area (there are usually separate spooling areas for each printer), *lpd* goes into action. It translates the print file if necessary, attaches any special information to the front or back of the file, and oversees the transmission of the file to the printer. Because the UNIX systems are used to spool all print files for the PC-compatible laser printers and for the large plotter, *lpd* must be present and operating on the UNIX workstation for remote printing to work from the PC compatibles or from the UNIX systems. The Macintosh computers spool their own print files and are therefore not dependent on *lpd* on a UNIX system.

sendmail

The *sendmail* daemon is also normally included with the operating system. *sendmail* is the most common UNIX Message Transfer Agent and is required for the electronic mail system's operation. Each User Agent mail program, whether on a UNIX system or on a desktop system, ultimately transfers messages to *sendmail* for delivery. It is *sendmail* that deciphers the recipient's address, requests an Internet address look-up if the recipient is not on the local machine, and eventually sends the message. Although *sendmail* is normally included with all UNIX systems, it may not be running, and it may not be configured correctly as it is distributed. A *sendmail* command must be inserted in the mail server's start-up scripts to start *sendmail* automatically each time the system is booted.

There are many options for the *sendmail* daemon that can be set by modifying the *sendmail* configuration file. Of particular interest for networks having a dial-up Internet connection are options that allow outgoing mail to be held in a queue and then sent periodically as a batch of messages. Queuing outgoing mail saves on telephone costs by preventing the router from making a separate telephone call for

every e-mail message that is sent. The default setting for *sendmail* is to send every outgoing message when it is received. There is an option ("C") that can be set in the *sendmail* configuration file to force *sendmail* to send every outgoing, flagged message to the mail queue instead. To make the "C" option work, the mailer definition flag "e" (for expensive) must be set for all offsite mail. If these options are set, then a queue-emptying time must also be set, by using the "-q" option, when *sendmail* is started. Usually, a queuing time of 1 hour is sufficient. Adding "-q1h" to the command that starts *sendmail* will tell *sendmail* to attempt to send all messages in the mail queue once every hour. More information about *sendmail* and its configuration has been presented by Costales and others (1993).

pcnfsd

A daemon called *pcnfsd*, which is required for NFS file sharing to work correctly between desktop computers and the UNIX systems, is not included with the UNIX operating system but is usually provided with the desktop computer's NFS software. *pcnfsd* must be properly loaded and configured on each UNIX system that is going to act as an NFS server for the desktop systems. This process involves copying the *pcnfsd* software to the UNIX system, modifying configuration files to adapt to the local system, compiling the daemon, and then starting the daemon as a background process. After the daemon is working correctly, a command is added to the system's start-up scripts so that the *pcnfsd* daemon is always started whenever the operating system is reinitialized.

popper

Another daemon that is not part of the UNIX operating system is required to allow a desktop e-mail User Agent to interact with the UNIX system's Message Transfer Agent. Several different daemons are available; the most commonly used daemon—and the one that is required by Eudora—is called *popper*. *popper* (version 1.831), a public-domain UNIX daemon that implements the POP1 and POP2 Post-Office Protocols, is available from the University of California at Berkeley (ftp.cc.berkeley.edu). After the software has been copied to the UNIX system, the process is much the same as that for installing *pcnfsd*. The only significant difference is that several system configuration files must be modified so that the operating system knows which ports are used by *popper*. Instructions for making these modifications come with the *popper* program.

INTERNET NAME RESOLUTION

When a user sends an e-mail message, how does the network know where to send it? If a user wants to create an

ftp or *telnet* connection to a remote computer, how does the software know how to find the correct remote computer? Brief mention was made earlier of the fact that each host must be assigned a unique Internet address. The following sections describe the Internet naming conventions, Domain Naming Service (DNS), and problems that arise in using these features with a dial-up connection. Albitz and Liu (1992) have provided further information on DNS and related services.

Internet addresses are composed of four decimal numbers, which can range from 0 to 255, separated by periods (for example, 112.250.66.1). Computers are perfectly happy using numeric Internet addresses, but, because numbers are difficult for humans to remember and because Internet functions may be moved between hosts, each host on the Internet is also assigned one or more unique names. Internet names are composed of strings of characters (usually words or abbreviations) separated by periods (for example, er.usgs.gov, hp.com). There is no fixed number of segments to an Internet host name, and host-name segments do not correspond to the numbers in the Internet address.

When the Internet was in its infancy and few computers were connected, a large file maintained by central computers on the network contained the address and name of every host on the network. Each host had a simple, one-word name; when a new host was added to the network, the host's name and address were added to the file. The file was periodically copied to every computer on the network. When a user or an application wanted to find a remote host, it simply looked up the host's name in this hosts file and retrieved the host's address. Eventually, the network outgrew this simple system. It became difficult to find new, unique names for hosts, and copying the hosts file to every host on the network was using significant network resources. At that point, the Internet's Domain Name Service (DNS) was created.

DOMAIN NAME SERVICE

DNS is a scheme for dividing the responsibility for maintaining lists of host names and Internet addresses. The universe of possible Internet host names—the Internet name space—is divided into a hierarchical set of domains; a central computer, called a primary name server, in each domain is responsible for the host tables within its domain and for administering its portion of the name space. Each primary name server can also delegate responsibility for portions of the name space within its domain to other computers, which can then delegate in turn. The Internet domain name space can be compared to a pyramid. At the very top of the pyramid are a few computers (root name servers) that have host tables containing the names and addresses of all second-level name-server computers. Each second-level name server has host tables containing the names and addresses of

all the hosts at that level and all the name servers on the next level below and so on until we get to a host that does not have any hosts below it. At the time DNS was created, host names were expanded from a single word to the present multisegment pattern. Internet addresses maintained the same form (four decimal numbers separated by periods), but the form was given new meaning. Now, the left side of the address specifies which domain within the Internet the host belongs to, and the right side of the address specifies the host within the domain. The dividing line between the domain portion of the address and the host portion is not fixed; it varies depending on the domain's size.

The relationship between an Internet address and the domain name space is not readily apparent, and it is not possible to translate an address into a host name without looking the address up in a host table. However, there is a direct relationship between the host name and the domain name space. Consider, for example, the host named **xxxx.er.usgs.gov**. Each portion of this host name is equivalent to a domain. The **gov** portion of the name refers to a large domain named "gov," which contains all Government computers that are attached to the Internet. There are many Government computers; no single computer has a host table that contains the name and address information for all Government computers. Instead, the **gov** domain is subdivided, primarily by Government agency. Several computers in the **gov** domain have host tables containing the names and addresses of at least one computer in each of **gov**'s subdomains. One of those subdomains, **usgs.gov**, contains all USGS computers that are attached to the Internet. The **usgs.gov** subdomain is similarly subdivided by region. The computers in the USGS Eastern Region are in a subdomain called **er.usgs.gov**. Several computers in the Eastern Region have host tables containing name and address information for all hosts within the **er.usgs.gov** subdomain. Because the Eastern Region's domain is not further subdivided, the **xxxx** portion of the host name above refers to a single host. For all Internet-style host names, the first segment of the name refers to the host computer itself, and the rest of the name refers to the domain in which it resides. There are usually at least two name servers within each domain; large domains often have more than two. One name server is usually set up as a primary name server, and the others are set up as secondary name servers. The primary name server is the computer where all changes to the domain's host information are made. Secondary name servers periodically copy the host information from the primary server. In this way, the domain is protected from losing its name service if a name server is down, and the processing of name-service requests is spread out over several machines.

What does all this have to do with finding remote hosts? The first step for any application in finding a remote host is to translate the host's name into an Internet address. The process that occurs if the network is using DNS is as follows. When a user attempts to create a *telnet* connection

to a remote host, the local system first searches for its local-domain name servers. It then sends a message to a local-domain name server requesting a name-to-address lookup. The local name server looks in its host table to see if it already knows the address. There are three possible outcomes. The local name server may know the requested address because it is within the server's domain. The name server may also know the requested address because it has received a previous request for the same information. Each time a name server receives an address from another name server, it saves that information for a while in case there is another request for the same information. In either of these two cases, the local name server returns the address information to the requesting application. The third possible outcome is that the local name server does not have the information in its tables. It does, however, always have address information for several root name servers. The local name server tracks down the requested address information by starting at the right side of the requested domain name. For the example given above—a request for the address of **xxxx.er.usgs.gov**—the local name server sends a request to one or more root name servers asking for the addresses of name servers in the **gov** domain. When it gets this information, it sends a request to one or more **gov** name servers asking for the addresses of name servers in the **usgs.gov** domain. Again, it sends a request to **usgs.gov** name servers asking for the addresses of **er.usgs.gov** name servers. Finally, it sends a request to the **er.usgs.gov** name servers asking for the address of host **xxxx**. When that information is returned, the local name server forwards the address to the requesting application.

Once the host name has been translated into an Internet address, the application software can pass the required data to TCP or UDP, which then passes it to IP. IP makes up a data packet containing the remote host's address in the header and the data in the packet body. The packet is then sent out on the network and routed to the correct host on the basis of the Internet address. A remarkable Internet feature is its built-in redundancy. If individual network links are unavailable, the remainder of the network is generally unaffected. Because the naming information is spread throughout the network and because there are usually at least two computers in every domain that hold the domain's naming information, the loss of a single computer does not affect the system other than perhaps to slow it down slightly. A similar redundancy is built in to the physical network. Although small local networks at the bottom of the pyramid, such as the Spokane LAN, may have only a single link to the Internet, there are many interconnections above that level. There are usually many different physical routes that can be used between two hosts. The physical routing is dynamically allocated so that the user does not need to know anything about routing to use the network. Individual data packets of the same message often take different routes to reach their destination. Fortunately, TCP can reassemble

the original message or file even if the packets arrive from different routes and not in their original order.

The correspondence between a host domain name and a computer's Internet address is not permanently fixed. Nor is there always a one-to-one relationship between host name and address. Host names are often assigned on the basis of the task performed. For example, a computer may be set up to provide public access to a particular database. Later a second database may be added to the same computer but with a different host name. At this point, the single computer having a single address has two host domain names; the names do not even have to be in the same domain. As long as the appropriate name servers are kept up to date, a user can find each database by requesting access to the host name used for that database. Later, the volume of usage may increase to the point that the single computer can no longer handle the load. One of the databases may be moved to a second computer having a different address. The host name referring to the moved database is transferred to the new computer along with it. As long as the name server's host tables are updated to show this change, users need never be aware of it. For this reason, it is better to use the host name than to use addresses directly; host names tend to follow functions, whereas addresses tend to stay with individual computers. It is the function that is of interest to most users.

The process described above may seem tedious and time consuming, but it is extremely robust. Neither of two possible alternatives is satisfactory. The first alternative is to always use addresses instead of names. This approach works for most applications but is unsatisfactory for the reasons mentioned above. In addition, there are several applications, such as some e-mail programs, that do not work directly with addresses. The second alternative is to not run DNS. If DNS is not running in the local domain, then each computer must use its own local host table to look up addresses. Each time a user wants to connect to a remote system or send mail to a user on a remote system that is not in the local host table, the user must find the system administrator for the local system, have the system administrator look up the address in question, and add the host name and address to the local host table. Of course, every time a remote host name-to-address relationship is changed, the local host table becomes out of date. This system would be adequate if the users in a local domain confined most activities to their domain but is clearly not feasible for large domains or for domains where users make many external connections.

DNS AND DIAL-UP CONNECTIONS

DNS address look-ups normally proceed so rapidly in the background, as an application is started or a connection is made, that the user is unaware of the process. However,

the delays become an annoyance when all local-domain name servers are on the other end of a dial-up connection. The router must dial the telephone and make the connection to a name server on the other end of the link every time an address look-up is requested, even if the requested address is a computer on the same LAN. The Spokane LAN was originally installed in this way. A delay of about 30 to 45 s occurred each time a name-service request was made if any of the following took place: (1) any *telnet* or *ftp* connection request using host name, (2) any *ping* command using host name, (3) any NFS mount or login command, or (4) any time the windowing system was initialized on the UNIX workstations. The delays were an annoyance but did not affect the LAN's functionality. However, a more serious problem also existed. If anything went wrong with any part of the link between the LAN and the remote name servers, all functions listed above would cease to work. In effect, the local network would function only if the outside connection was reliable.

PRIMARY NAME-SERVER SOLUTION

There are several possible solutions to this problem. The first, and potentially most satisfactory, is to set up a subdomain for the local network. This solution requires at least two full-time computers on the local network that can run name-server software, a local network administrator who is able to maintain the new primary name server's host tables, and the delegation of new domain responsibility to the local site by the existing parent-domain administrator. At the Spokane site, the first two requirements were met, but the third was not. Therefore, this solution was not a viable option.

Where the new primary name-server solution is available, the procedure is straightforward. DNS is set up on at least two systems on the local network; one computer is configured as the primary name server for the new domain, and one or more other computers are configured as secondary name servers. The domain name selected is usually a subset of the parent domain. For example, if the parent domain is **wr.usgs.gov** (USGS's Western Region), the new domain name could be **sfo.wr.usgs.gov** for the Spokane field office. Each host on the local network is assigned a unique host name and an Internet address, and the name and address information is entered in the primary name server's host tables. Each host on the local network must be configured to use the new name servers for their name-service requests. The administrator of the parent network is then given the name and address information for all name servers in the new domain. Once these steps have been completed, all requests for local addresses are handled on the local network. Local network functioning is no longer dependent on the connection to the parent domain. The only requests that will go outside the local network are those that

involve services beyond the local domain. All requests for information that come from outside the local domain will be routed by the parent-domain name servers to the new local-domain name servers.

SECONDARY NAME-SERVER SOLUTION

A second solution to the dial-up name-server problem is to configure one or more computers on the local network as secondary name servers for the parent domain. This solution requires a single full-time local computer that can run DNS and a local administrator who can set the system up as a secondary name server. Because a subdomain is not created, all host names use the same domain name as the parent domain, and all host names and Internet addresses are assigned by the parent-domain administrator. The local network administrator does not need to update any host tables on the secondary name servers because they are automatically copied from the domain's other name servers. As in the first solution, all local hosts must be configured to use the new secondary name server for their requests. Once this process is complete, the system functions as if it were a new subdomain. The only differences are in the local network administrator's duties and in the periodic copying of host tables across the dial-up link. The Spokane LAN is set up with the Sun workstation as a secondary name server.

If the new secondary name server is registered with the parent domain's network administrator and if the administrator chooses to add it to the domain's list of secondary servers, it can share part of the domain's name-service burden beyond its local network. However, the secondary server can still function as the name server for its local network without being added to any other name server's list of name servers.

There is a configurable amount of time between subsequent requests to copy host tables from primary servers to secondary servers. The information in a secondary name server's host tables may be slightly older than that in a primary's host tables because some time is required to propagate the information throughout the domain.

All information stored in host tables other than on primary name servers has an associated time to live. If the connection between a secondary server and the rest of the domain's name servers is broken, there is initially no effect on local operations. However, when the time to live expires for any entry in the host tables, that entry is discarded. Eventually, the secondary server discards all its information. This process usually takes a week or more. If it is not possible to restore the connection within this time, the secondary server will fail to respond, and local network services will become unavailable. To restore local network service, the secondary name server must be temporarily reconfigured as a primary server, and the host tables must be restored.

CACHE-ONLY NAME-SERVER SOLUTION

There is another possible solution to the dial-up name-server problem, although it is not as satisfactory as the other two solutions. It is possible to configure a computer running DNS as a cache-only name server. The cache-only name server has the same requirements as the secondary name server—that is, a single full-time computer running DNS. The difference is that the cache-only name server never copies the entire domain's host table from the parent domain. Instead, it caches, or remembers, all address information it receives through name-service requests. If a request is made for an address that it has recently looked up, it immediately returns that address information from its cache. If the information has not been looked up recently, it must request the information from another name server. The main functional differences between a secondary server and a cache-only server are that the time to live for cache information is usually only about 24 hours and that the individual host data are replenished only when a request is made. Although a secondary name server can usually maintain its functionality for about a week, a cache-only server usually begins failing to respond within about a day. Also, users see many more delays with a cache-only name server on a dial-up connection than they do with a secondary server. Consider the user who makes a *telnet* connection within the local network every couple of days; with a cache-only server, the user is going to see a dial-up delay every time the connection is requested. With a secondary name server on the local network, the user will never see the dial-up delay because the server will maintain the information by periodically copying it as a background process.

DNS NAME LOOK-UP

A further problem with name-server dial-up connections is related to how domain names are processed for each request. Each computer that runs DNS is configured to know its own domain name. Thus, users can use shortcut names for hosts that are in the same domain. For example, if a user wants a *telnet* connection from the host **xxxx.er.usgs.gov** to the host **yyyy.er.usgs.gov**, the user can simply request a *telnet* connection to **yyyy**. The DNS software automatically adds **.er.usgs.gov** to the name **yyyy** to create the complete name to look up. DNS can go one step further. If the same user wants to connect to the host **zzzz.wr.usgs.gov**, the user can request a connection to **zzzz.wr**, and DNS will add **.usgs.gov** to the name and proceed with the look-up.

The problem is that DNS is not really quite as smart as it sometimes looks. The program that does all this work is not DNS but another daemon that is usually called *named* (*in.named* on SunOS systems). *named* has a set of rules that it uses to create complete domain names from partial

names. The default rules are optimized for systems having fast links to their domain name servers. The default rules also assume that, most commonly, the user will use a shortened version of the domain name. The first thing *named* does is add its known domain name to the requested name and search for that name. If that name is an unknown host, then *named* shortens the domain name that it adds to the request by eliminating the leftmost segment of the domain name, adds the remainder to the requested name, and sends out a second request. It continues this process until it runs out of domain name. The last thing that it attempts is to request the address for the host name as it was entered by the user. For example, if the user attempts to *telnet* to **yyyy.er.usgs.gov** from **xxxx.er.usgs.gov**, by the command *telnet yyyy*, *named* first attempts to look up **yyyy.er.usgs.gov**. If the local network has a primary or secondary name server that responds rapidly enough with the address, the process stops, and there is no perceivable delay. If the user enters the command *telnet yyyy.er.usgs.gov*, *named* first attempts to look up the address **yyyy.er.usgs.gov.er.usgs.gov**. Because this domain name does not exist, the local name server will not find it in its host tables. It will then attempt to contact a root name server with the request. Of course, contacting a root server involves sending a request out on the dial-up connection, and that request forces both *named* and the user to wait. Because *named* expects a rapid response, it attempts further requests before the dial-up connection can be completed. The actual default sequence of requests from *named* for the above example will look like this:

```
yyyy.er.usgs.gov.er.usgs.gov
yyyy.er.usgs.gov.usgs.gov
yyyy.er.usgs.gov.gov
yyyy.er.usgs.gov
```

Clearly, this procedure causes delays for every request other than those for a single host name within the local network. A possible solution might be teaching the users to only use single host names if they are making a local connection. Unfortunately, some applications create full domain names for the user before passing the request on to *named*. There is no way, short of reprogramming those applications, to prevent the full domain name look-up.

MODIFYING *named*'s SEARCH LIST

Fortunately, this predicament can be resolved by modifying the default behavior of *named*. *named* is a daemon that is usually included with UNIX operating systems. It is also included with some desktop computer TCP/IP software. Beginning with version 4.8.3, *named* has included the capability of modifying its default behavior by changing its "search list." The search list can be changed so that *named* only has two options. If the request is for a single-part host name, *named* will add its domain name; if the request is for

a multipart name, *named* only passes the name along as requested. To continue the example above, the modified version of *named*, given the request to look up **yyyy**, would add the domain name and send a request for **yyyy.er.usgs.gov**, as before. Given the request to look up **yyyy.er.usgs.gov**, it would make no changes and simply look up the name as given. If **yyyy.er.usgs.gov** exists in a local server's host tables, there will be no dial-up delay in either case. This modification eliminates the possibility of requesting partial names; a request to look up **zzzz.wr** will always fail. If the name server does not include *named* (version 4.8.3 or later) (SunOS 4.1.2 includes *named*, version 4.8.1), the most recent version of *named* can be obtained from UUnet. Appendix A contains detailed information about installing a new version of *named* and modifying the search list.

RUNNING DNS WITHOUT NIS

An additional problem can arise if the name server is running the Sun operating system—SunOS (version 4.x). It may also arise on computers running Sun's Solaris operating system. Sun's version of DNS assumes that Network Information Service (NIS or NIS+) is also running. As delivered with the operating system, DNS will not run correctly without NIS. One solution to the problem is simply to install and run both NIS and DNS on the name server. Sun advocates this solution, but NIS is complicated, uses a significant amount of system resources, and is not needed for small networks.

DNS can be run on Sun systems without NIS, but doing so requires a modification to the shared "C" library, *libc.so*. Sun will not offer any technical support for this modification, but the details of how to make the modification are available on the Internet. The Sun workstation on the Spokane network is currently running DNS without NIS and experiencing no apparent problems. Appendix B contains further details about how to run DNS without NIS on Sun computers.

SERVICES PROVIDED

Just as the network hardware was installed so that the network software could be used, the network software was installed so that application software could make use of network services. The application software exists only to allow users to accomplish their tasks. The following summary describes the Spokane LAN's status in terms of goals that were outlined at the time of the network design.

PRINTING AND PLOTTING

The network now provides transparent access to laser-quality printing for all users whose desktop computers are

attached to the network. As of this writing, 11 of 17 users have direct network connections to their desktop computers. Of the remaining six, three have no desktop computers in their offices, and the other three have not yet been connected to the network. All users also have access to laser-quality printing through three computers (one Macintosh and two PC compatibles) attached to the network in a shared computer room. Macintosh users all have access to a LaserWriter Pro, a high-speed network printer, and a Data General laser printer through a dedicated print server. All PC-compatible users, whether in the DOS or the Windows environment, have access to two PostScript-compatible laser printers through a dedicated print server that can switch a print job from one printer to the other if the first printer is busy. The Sun workstation has its own laser-quality printer and access through the spooling system to the network printers.

The large-format pen plotter is currently attached to a serial port on a shared PC compatible. It is currently available to all users by copying a data file to the shared PC compatible and then plotting from there. A port has been set aside on one of the print servers for the plotter. Spooling has been set up for the plotter on the Sun workstation. The plotter will be moved to the print server where all users will have direct access to it as a network service.

FILE SHARING

An original goal was to provide file-sharing capability between all computers on the network. File sharing should be distinguished from file transfer. File-transfer capabilities are summarized below; file sharing refers specifically to mounting a portion of a remote file system and then using it as if it were a local disk. This goal has been only partially satisfied. The two UNIX workstations have complete access to each other's file systems, subject to file-ownership restrictions. The Macintosh users have complete access to file systems on other Macintosh computers by using AFP and, if they run NFS Share, have complete access to file systems on the two workstations. PC-compatible users in both DOS and Windows environments also have complete access to the two workstation file systems if an NFS software package is running on the desktop computer. The most serious deficiency in the present system is the inability of most PC-compatible systems to share files with each other without going through an intermediary or using *ftp*. This inability is due to the lack of an NFS server module in PC-NFS, which is currently running on most PC compatibles. This problem will probably be solved in future software versions. Either PC-NFS will include an NFS server, or a newer version of ArcView will be compatible with other NFS software packages that do include NFS servers.

REMOTE LOGIN AND FILE TRANSFER

The remote login and file-transfer utilities meet all design goals. The two modules that are necessary for these functions, *telnet* and *ftp*, are available for all platforms as either public-domain or commercial programs. Several Internet navigation shells that run on top of *telnet* and *ftp* help users find their way around the Internet and retrieve information that they find. More information on using the Internet as an information resource has been given by Dern (1994), Fraase (1993), and Krol (1992).

ELECTRONIC MAIL AND DOCUMENT ATTACHMENT

All users now have access to electronic mail, both locally and through the Internet connection. Users who have desktop systems attached to the network have the choice of processing their mail (1) from a dedicated mail terminal attached to a workstation, (2) at their desks through a *telnet* connection to a workstation, or (3) at their desks by using Eudora (if they are running Windows or a Macintosh). Users who do not have a desktop network connection can use either the mail terminal or a shared computer to process their mail. The mail terminal can be used to process standard ASCII text messages only. Eudora users, in either Macintosh or Windows environments, can also attach data files of any type to their messages, encoded in either Bin-Hex or MIME format. The electronic-mail and document-transfer goals have been fully met for all UNIX, Macintosh, and Windows users; DOS users still lack document-transfer capability.

BACKUP

The final network design goal was to provide automated backup of all systems on the network. The appropriate hardware and software combination to implement this goal has not yet been found. The Sun workstation includes an 8-mm tape drive with a capacity of 2.3 GB per tape. This capacity is sufficient for an incremental backup of all disk drives in the office without changing tapes. An automated backup system could be constructed if software could be found that would copy data over the network from each operating system in use to the UNIX workstation. If NFS server modules become available for all desktop systems, it may be possible to write a UNIX shell program that would mount each desktop disk through NFS, back it up by using standard UNIX utilities, and then dismount it.

Meanwhile, the UNIX workstations have utilities for automated backup as part of their operating systems. Several desktop systems have automated backup systems making use of 1/4-in. tape drives or removable-cartridge hard-disk drives. It may also be possible to add a 1/4-in. tape drive

to a shared computer for backing up the remaining desktop systems over the network.

SUMMARY

Most design goals of the Spokane LAN have been met. All users have access to laser-quality printing, large-format plotting, file sharing, electronic mail and document transfer, and the Internet. The high-speed dial-up connection to the Internet caused some configuration difficulties, but they have been solved, and the single 56-Kbaud line's performance is sufficient for current levels of use. The remote Spokane staff member is still using the 9600-baud modem connection to the Sun workstation; however, once he is connected to the workstation, he has access to all network services. Most important, standard systems are used wherever

possible, and the network is simple enough that, once installed, it can be administered as a part-time job function by someone other than a computer specialist.

REFERENCES CITED

- Albitz, Paul, and Liu, Cricket, 1992, DNS and BIND: Sebastopol, Calif., O'Reilly & Associates, 381 p.
- Costales, Bryan, Allman, Eric, and Rickert, Neil, 1993, Sendmail: Sebastopol, Calif., O'Reilly & Associates, 792 p.
- Dern, D.P., 1994, The Internet guide for new users: New York, McGraw-Hill, 570 p.
- Fraase, Michael, 1993, The Mac Internet tour guide: Cruising the Internet the easy way: Chapel Hill, N.C., Ventana Press, 288 p.
- Krol, Ed, 1992, The whole Internet user's guide & catalog: Sebastopol, Calif., O'Reilly & Associates, 376 p.

APPENDIXES

APPENDIX A. INSTALLING *named* 4.8.3 AND MODIFYING DNS SEARCH LISTS

OBTAINING *named* (VERSION 4.8.3)

If a UNIX-based computer is used as a secondary or cache-only name server with a dial-up connection to the primary name server, it should be running *named* (version 4.8.3 or later). *named* is just one part of the DNS software that runs on UNIX systems. The complete package is called Berkeley Internet Name Domain (BIND). Whenever one portion of BIND, such as *named*, is updated, the rest of BIND should be updated simultaneously. The following description of obtaining a new version of BIND and installing *named* on a UNIX workstation is summarized from Albitz and Liu (1992), Sun Internal Bug Report #1043729, and Mark Curtis (oral commun., 1993). These sources are highly recommended for anyone attempting to make these changes. This example has been tested only on a Sun workstation running SunOS 4.1.2. The most recent version of BIND can be obtained over the Internet by anonymous *ftp* to UUnet as follows (commands typed by the user are in bold):

```
% cd /tmp           [change to a temporary storage area]
% ftp ftp.uu.net      [connect to UUnet]
Connected to ftp.uu.net
220 ftp.UU.NET FTP server . . .
Name (ftp.uu.net.:user): ftp
331 Guest login ok, send e-mail address
as password.
Password:             [enter your e-mail address]
```

At this point, a long welcoming message is displayed.

```
ftp> cd networking/ip/dns/bind [change to the
                                correct directory]
ftp> binary                 [change to binary mode]
ftp> get 4.8.3.tar.Z         [transfers BIND to local
                                system]
ftp> quit                   [finished with ftp]
```

At this point, BIND is in a compressed, archived format. To uncompress the file and unarchive the individual pieces, enter the following command:

```
% zcat 4.8.3.tar.Z | tar xvf -
```

There should now be several subdirectories under the temporary directory, including `bin`, `doc`, `include`, `man`, `master`, `named`, `res`, and `tools`. If so, downloading the new BIND has been successful.

INSTALLING *named* (VERSION 4.8.3)

There are several steps to installing a new version of BIND. A new resolver library must be built and installed, and *named* must be recompiled and installed. The resolver library is part of the operating system's shared "C" libraries. As a preliminary to building a new resolver library, the shared "C" library source code (part of the Shlib_Custom software category) must be included in the operating system installation. If it was not included in the original installation, it can be added by following the instructions in the Systems Administration Manual (see "*add_services*").

BUILDING A RESOLVER LIBRARY

The first step in the installation process is to build a new resolver library from BIND 4.8.3. From the temporary directory, enter the following commands:

```
% cd res
% chmod +w Makefile
Edit Makefile:
add -I../include -DBSD=43 -pic -c
to the line that begins with CFLAGS=. Then:
% make
```

The resolver should compile without errors, several object files (with `.o` extensions) should be created, and a library archive, `libresolv.a`, should be created. Then, run *ranlib* on the new archive:

```
% ranlib libresolv.a
```

The new resolver library should now be complete. To test that the new resolver modules will be usable, do the following test:

```
% ld -assert pure-text *.o
```

If *ld* issues no complaints, the object modules are probably safe to use. It is important to use this version of `libresolv.a` for all subsequent compilations.

UPDATING THE SHARED LIBRARY

After a new resolver library version has been built, the next step is to incorporate the new resolver in the shared library, `libc.so`. This step is necessary because many network programs do not use *named* for name look-ups. Instead, they incorporate the name resolver directly by linking to the shared library. The following commands should create a new shared library including the new resolver:

```
% cd /usr/lib/shlib.etc
% mkdir tmp           [create a new temp directory]
% cd tmp
% ar x ../libc_pic.a  [extract object files from
                        archive]
% rm __.SYMDEF         [__ represents two underscores]
% mv rpc_dtablesize. rpc_dtablesize.o
```

```
% mv rpc_commdata.rpc_commdata.o
% mv xccs_multibyte.xccs_multibyte.o
      [file name corrections required
      by bug in archive program]
```

If the system is running NIS, continue with the commands given below; if the system is to run DNS without NIS, follow the steps given in appendix B for modifying the shared library before proceeding with the following steps.

```
% ar x /tmp/res/libresolv.a [be sure this path
                             points to the new
                             resolver library]
% rm __.SYMDEF      [__ represents two underscores]
% rm mkstemp.o      [duplicate module]
% ld -assert pure-text *.o
```

All object modules to be compiled in the new library should pass this test. If they do not, check that the appropriate CFLAGS have been used for the compilation and that *ranlib* has been run on the resolver library.

```
% cd ..
```

Compare the list of object modules in the file *lorder-sparc* with the object modules in the *tmp* sub-directory. If new modules exist in the directory that are not in *lorder-sparc*, they must be added to the end of the list. Then

```
% make libc.so
```

A new library, *libc.so.x.y.z*, will be created in the current directory. The version number, *x.y.z*, will be one greater than the last version created. This new library should be tested before it is installed on the system. Check for an existing **LD_LIBRARY_PATH** environmental variable:

```
% setenv
```

If **LD_LIBRARY_PATH** exists, save the contents so that it can be restored after the tests.

```
% setenv LD_LIBRARY_PATH `pwd`
```

This command sets the environmental variable, **LD_LIBRARY_PATH**, to point to the new library. Any network commands that use the shared library for name look-up (*ping*, *ftp*, *telnet*, and so on) can then be used to test the library. Once the functioning of the new library has been confirmed, it can be installed. The user must be logged in as root to install the library:

```
# cp libc.so.x.y.z /usr/lib
# ldconfig
# unsetenv LD_LIBRARY_PATH
```

If the environmental variable **LD_LIBRARY_PATH** had a previous value, restore it at this point.

COMPILING *named*

The next step in the process is to build a new version of *named*:

```
% cd ../named      [change to new named directory]
% chmod +w Makefile
```

Edit Makefile:

```
add -I../include -DBSD=43 -pic -c
to the line that begins with CFLAGS=. Also, add
../res/libresolv.a to the line that begins with
RES=. Then edit the pathnames.h file to reflect correct
pathnames for the system. In particular, for SunOS 4.1.2
systems, the line
```

```
#define _PATH_XFER "/usr/libexec/named-xfer"
should be changed to
```

```
#define _PATH_XFER "/usr/etc/in.named-xfer"
Then
```

```
% make
```

Finally, if the compilation proceeds without errors, the new *named* and *named-xfer* daemons must be installed into */usr/etc*. The user must be logged in as root to perform this installation.

```
# mv named /usr/etc/in.named
# mv named-xfer /usr/etc/in.named-xfer
```

The new *named* daemon installation is now complete.

MODIFYING DNS SEARCH LISTS

The reason for modifying the resolver and the shared library is to be able to control the resolver's search list. (The default search list was described in the section "DNS Name Look-Up.") If version 4.8.3 or later of the resolver is running, the search list can be changed by modifying the resolver configuration file, *resolv.conf*. This file is usually found in the */etc* directory. To change the search list operation, add a search directive to the end of the configuration file. A search directive explicitly specifies the list of domains to be searched. The directive consists of the keyword, *search*, beginning in column one, followed by up to six domain names to be searched. The first domain listed in the search directive must be the system's default domain. If the default domain for the system is **wr.usgs.gov**, the following search directive would be equivalent to the default behavior without a search directive:

```
search wr.usgs.gov usgs.gov gov
```

To prevent the resolver from creating nonexistent host names by adding portions of the domain name to host names that are already complete, add a search directive to the name server's resolver configuration file, which contains only the system's default domain. For the example used above:

```
search wr.usgs.gov
```

This search directive, together with BIND 4.8.3, will allow a secondary name server to serve a local network without contacting the primary name server for every name look-up. Therefore, the local network having a secondary name server can maintain local operations even if the link to the primary name server is disabled.

APPENDIX B. DNS WITHOUT NIS ON SUNOS 4.x SYSTEMS

The version of DNS that is included with SunOS 4.x operating systems is intended to run only with NIS or NIS+ installed. To run DNS without NIS, the shared “C” library must be modified. To modify the library, the shared “C” library source code (part of the Shlib_Custom software category) must be included in the operating system installation. If it was not included in the original installation, it can be added by following the instructions in the Systems Administration Manual (see “*add_services*”). The information included in this appendix is primarily a summary of Sun Internal Bug Report #1043729, which is recommended for any user attempting to make these changes. The following commands should create a new shared library that can run DNS without running NIS:

```
% cd /usr/lib/shlib.etc
% mkdir tmp           [create a new temp directory]
% cd tmp
% ar x ../libc_pic.a   [extract object files from
                        archive]
% rm __.SYMDEF         [__ represents two underscores]
% mv rpc_dtablesize.rpc_dtablesize.o
% mv rpc_commonddata.rpc_commonddata.o
% mv xccs_multibyte.xccs_multibyte.o
                        [file name corrections required by
                        bug in archive program]
% ar x /usr/lib/libresolv.a
```

This library is the original resolver library. If a newer resolver library is used, see appendix A for instructions for creating the new resolver before continuing.

```
% rm __.SYMDEF         [__ represents two underscores]
% ld -assert pure-text *.o [test object modules]
% rm gethostent.o      [remove old host name resolver]
% rm strcasecmp.o      [remove redundant module]
% cd ..
```

Edit the `lorder-sparc` file to remove the reference to `gethostent` and to add references to new resolver modules. Apply the following patch to change `lorder-sparc`:

```
% diff -rc2 lorder-sparc.orig lorder-sparc
*** lorder-sparc.orig
--- lorder-sparc
*****
*** 150,154 ****
getwd.o
getnetgrent.o
! gethostent.o
ypxdr.o
```

```
ttyname.o
--- 150,161 ----
getwd.o
getnetgrent.o
! gethostnamadr.o
! sethostent.o
! res_query.o
! res_mkquery.o
! res_send.o
! res_debug.o
! res_comp.o
! res_init.o
ypxdr.o
ttyname.o
```

For systems running SunOS 4.1.2, add the entry `mbilib.o` to the end of the `lorder-sparc` file and make the following changes to Makefile:

Change the lines

```
libc.so:
ld -assert pure-text `${OBJSORT}
lorder-sparc tmp`
```

to

```
libc.so:
ld -assert pure-text `${OBJSORT}
lorder-sparc tmp` -ldl
```

Also, change the lines

```
libcs5.so:
ld -assert pure-text `${OBJSORT}
lorder-sparc tmp`
```

to

```
libcs5.so:
ld -assert pure-text `${OBJSORT}
lorder-sparc tmp` -ldl
```

Now make the new shared library

```
% make libc.so
```

A new library `libc.so.x.y.z` will be created in the current directory. The version number `x.y.z` will be one greater than the last version created. This new library should be tested before it is installed on the system. Check for an existing `LD_LIBRARY_PATH` environmental variable:

```
% setenv
```

If `LD_LIBRARY_PATH` exists, save the contents so that it can be restored after the tests.

```
% setenv LD_LIBRARY_PATH `pwd`
```

This command sets the environmental variable **LD_LIBRARY_PATH** to point to the new library. Any network commands that use the shared library for name look-up (*ping*, *ftp*, *telnet*, and so on) can then be used to test the library. Once the functioning of the new library has been confirmed, it can be installed. The user must be logged in as root to install the library:

```
# cp libc.so.x.y.z /usr/lib  
# ldconfig  
# unsetenv LD_LIBRARY_PATH
```

If the environmental variable **LD_LIBRARY_PATH** had a previous value, restore it at this point.

SELECTED SERIES OF U.S. GEOLOGICAL SURVEY PUBLICATIONS

Periodicals

Earthquakes & Volcanoes (issued bimonthly).

Preliminary Determination of Epicenters (issued monthly).

Technical Books and Reports

Professional Papers are mainly comprehensive scientific reports of wide and lasting interest and importance to professional scientists and engineers. Included are reports on the results of resource studies and of topographic, hydrologic, and geologic investigations. They also include collections of related papers addressing different aspects of a single scientific topic.

Bulletins contain significant data and interpretations that are of lasting scientific interest but are generally more limited in scope or geographic coverage than Professional Papers. They include the results of resource studies and of geologic and topographic investigations, as well as collections of short papers related to a specific topic.

Water-Supply Papers are comprehensive reports that present significant interpretive results of hydrologic investigations of wide interest to professional geologists, hydrologists, and engineers. The series covers investigations in all phases of hydrology, including hydrogeology, availability of water, quality of water, and use of water.

Circulars present administrative information or important scientific information of wide popular interest in a format designed for distribution at no cost to the public. Information is usually of short-term interest.

Water-Resources Investigations Reports are papers of an interpretive nature made available to the public outside the formal USGS publications series. Copies are reproduced on request unlike formal USGS publications, and they are also available for public inspection at depositories indicated in USGS catalogs.

Open-File Reports include unpublished manuscript reports, maps, and other material that are made available for public consultation at depositories. They are a nonpermanent form of publication that may be cited in other publications as sources of information.

Maps

Geologic Quadrangle Maps are multicolor geologic maps on topographic bases in 7.5- or 15-minute quadrangle formats (scales mainly 1:24,000 or 1:62,500) showing bedrock, surficial, or engineering geology. Maps generally include brief texts; some maps include structure and columnar sections only.

Geophysical Investigations Maps are on topographic or planimetric bases at various scales; they show results of surveys using geophysical techniques, such as gravity, magnetic, seismic, or radioactivity, which reflect subsurface structures that are of economic or geologic significance. Many maps include correlations with the geology.

Miscellaneous Investigations Series Maps are on planimetric or topographic bases of regular and irregular areas at various scales; they present a wide variety of format and subject matter. The series also includes 7.5-minute quadrangle photogeologic maps on planimetric bases that show geology as interpreted from aerial photographs. Series also includes maps of Mars and the Moon.

Coal Investigations Maps are geologic maps on topographic or planimetric bases at various scales showing bedrock or surficial geology, stratigraphy, and structural relations in certain coal-resource areas.

Oil and Gas Investigations Charts show stratigraphic information for certain oil and gas fields and other areas having petroleum potential.

Miscellaneous Field Studies Maps are multicolor or black-and-white maps on topographic or planimetric bases for quadrangle or irregular areas at various scales. Pre-1971 maps show bedrock geology in relation to specific mining or mineral-deposit problems; post-1971 maps are primarily black-and-white maps on various subjects such as environmental studies or wilderness mineral investigations.

Hydrologic Investigations Atlases are multicolored or black-and-white maps on topographic or planimetric bases presenting a wide range of geohydrologic data of both regular and irregular areas; principal scale is 1:24,000, and regional studies are at 1:250,000 scale or smaller.

Catalogs

Permanent catalogs, as well as some others, giving comprehensive listings of U.S. Geological Survey publications are available under the conditions indicated below from the U.S. Geological Survey, Information Services, Box 25286, Federal Center, Denver, CO 80225. (See latest Price and Availability List.)

"Publications of the Geological Survey, 1879–1961" may be purchased by mail and over the counter in paperback book form and as a set of microfiche.

"Publications of the Geological Survey, 1962–1970" may be purchased by mail and over the counter in paperback book form and as a set of microfiche.

"Publications of the U.S. Geological Survey, 1971–1981" may be purchased by mail and over the counter in paperback book form (two volumes, publications listing and index) and as a set of microfiche.

Supplements for 1982, 1983, 1984, 1985, 1986, and for subsequent years since the last permanent catalog may be purchased by mail and over the counter in paperback book form.

State catalogs, "List of U.S. Geological Survey Geologic and Water-Supply Reports and Maps For (State)," may be purchased by mail and over the counter in paperback booklet form only.

"Price and Availability List of U.S. Geological Survey Publications," issued annually, is available free of charge in paperback booklet form only.

Selected copies of a monthly catalog "New Publications of the U.S. Geological Survey" are available free of charge by mail or may be obtained over the counter in paperback booklet form only. Those wishing a free subscription to the monthly catalog "New Publications of the U.S. Geological Survey" should write to the U.S. Geological Survey, 582 National Center, Reston, VA 22092.

Note—Prices of Government publications listed in older catalogs, announcements, and publications may be incorrect. Therefore, the prices charged may differ from the prices in catalogs, announcements, and publications.

