NAT'L INST. OF STAND & TECH R.I.C.
A11103 520619

NIST
PUBLICATIONS

NISTIR 5757

Sharing Information Via the Internet - An Infoserver Case Study

Robert H. Bagwill

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Computer Systems Laboratory
Distributed Systems Engineering
Gaithersburg, MD 20899

NIST

QC
100
.U56
NO.5757
1995

NISTIR 5757

Sharing Information Via the Internet - An Infoserver Case Study

Robert H. Bagwill

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Computer Systems Laboratory
Distributed Systems Engineering
Gaithersburg, MD 20899

November 1995

U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary

TECHNOLOGY ADMINISTRATION
Mary L. Good, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Arati Prabhakar, Director

Contents

1 Overview

1.1 Goals

To increase the accuracy and timeliness of their business processes, many organization are moving towards electronic commerce. One of the application domains which can be prototyped and implemented relatively easily is the electronic dissemination of documents.

A variety of proprietary systems are available from commercial vendors, but the most rapid deployment of electronic document distribution has been by systems on the Internet using software based on open standards. The TCP/IP-based Internet, with its global coverage, relatively high-bandwidth and low latency, and wealth of freely available software, makes a particularly fertile ground for experiments in this area.

This report describes some of the experiences of the Distributed Systems Engineering Group has had employing this software.

> *Reader's Note:* a URL (**U**niform **R**eference **L**ocator) is a way of describing a method and location for accessing a particular file via the World Wide Web (WWW). URLs for various specifications are denoted in the text using this format:
>
> **URL: http://internet.org/dir/file**
>
> Unfortunately, given the volatility of the WWW, and its current lack of archiving protocols, the URLs given in this report probably will be out-of-date by the time it is published.

1.2 Requirements and Services

On the Internet, the most widely used protocols for distributing information are electronic mail, Usenet, FTP, Gopher, Z39.50, and World Wide Web. Each of the protocols has a client/server architecture, and each has its own advantages and disadvantages. By employing several of the components, the strengths of one service can compensate for the weaknesses of another.

Electronic mail (email) is becoming a integral part of the business process of many organizations. Although most businesses still use proprietary email systems for their isolated LAN's, many are beginning to use TCP/IP based email, and are subscribing to commercial networks that gateway to or parallel the Internet.

Some of the ways email can be used are:

- Email can be sent person-to-person.
- Private email lists can be used to send messages, documents, and software to a list of subscribers.

- Public email lists can be used to allow a members of a discussion group to broadcast messages to other members of the group.
- An automatic email response system can answer email requests for documents.

Some disadvantages of email are:

- large email lists are difficult to manage because of the number of subscribe and unsubscribe requests, and the number of bad addresses, bounces and other email errors.
- large email lists put a heavy load on the mail server.
- Information is "pushed" by the author, rather than "pulled" by the reader
- Readers have difficulty managing their email messages when they have subscribed to many email lists, and the lists have heavy traffic.
- Unless all the traffic is archived, it is difficult to reconstruct a past discussion.
- Email is delivered asynchronously using a store-and-forward architecture, so replies to a broadcast message can arrive at a given site before the original message.
- Internet email doesn't handle large or binary messages gracefully.

Some solutions to those disadvantages are:

- Mailing list management software allows readers to subscribe and unsubscribe via email messages, and can allow the list owner to management the list remotely
- Mail archive server software allows readers to fetch digests of list traffic or other files via email messages.
- Utilities like `split` and `UUENCODE` can be used to split large files and to encode binary files.
- The MIME standard can be used to support multi-file multi-format mailings.

Usenet news is a specialization of Internet email which creates a distributed forum similar to a bulletin board system (BBS) discussion group or chat line. Much of the technical information on the Internet is distributed and advertised over Usenet. A site can choose to subscribe to and forward any number of news groups. In addition, an organization can create its own internal Usenet discussion groups. But creating a limited subscription newsgroup, e.g., only Sun customers, is not easily accomplished. As a result, organizations usually use limited subscription email lists, or a private BBS to provide that type of service.

Some advantages:

- Offline readers are available.
- Threads of discusssion can be maintained.
- Traffic is archived.
- Archive size can be limited by size or number of messages.

Some disadvantages:

- Not all servers and clients support threading.
- Messages may be received out of order.
- Groups are subject to inappropriate postings.
- Expiration of messages means important messages can be lost.

The relevant specification is Internet RFC 977.

URL: gopher://ds.internic.net/00/rfc/rfc977.txt

FTP is the oldest of the Internet browsers. The FTP user sees the remote system in terms of directories and files. The user can change from one directory to another, list the contents of the directory, and get or put files to or from the directory (subject to the permissions set on the directories and files).

The user starts the FTP client application on the local system, and specifies the name or address of the remote system. The FTP client connects to the FTP server on the remote system, and begins a session.

Anonymous FTP allows users who do not have an interactive user account on the remote system to login to a special anonymous account. The anonymous FTP user is restricted to an anonymous FTP directory specified by the administrator. Although it is called *anonymous*, most of the Internet archive servers use an enhanced FTP server which asks for the email address of the user, and keeps track of logins and transfers by Internet address.

Anonymous FTP is most useful for users who have direct Internet access, and are familiar with the FTP utility.

Advantages:

- Directory navigation commands are similar to those of DOS or UNIX.
- Binary files are handled easily.
- Large files are handled easily.

Disadvantages:

- The reader needs full Internet access.
- Downloads of large files may be interrupted, and can't be resumed.
- Each ftp session requires that state be maintained by the FTP server, therefore many FTP sessions create a large load for the server.
- On UNIX systems, there are no mechanisms or strong conventions to indicate file formats, therefore a reader has to download a file to find out what file format it is.
- Compressed files may not be in a format the reader can decompress.

Solutions:

- Establish file format conventions on the server.
- Establish directory content conventions
- Use an FTP server which includes indexing, packaging, and compresssion extensions.

The relevant specification for FTP is Internet RFC 959.

URL: gopher://ds.internic.net/00/rfc/rfc959.txt

Gopher is an experimental distributed document delivery service developed at the University of Minnesota. It allows users to browse through directories, view files, and download them via Gopher or email. Gopher, like anonymous FTP, does not require an account on the remote system for each user.

Gopher access can be provided to dial-in users as well as those who are connecting via the Internet. This makes Gopher a good replacement for a read-only BBS.

The user starts the Gopher client application on the local system, and specifies that name or address of the remote system. The Gopher client connects to the Gopher server and begins a session. The user sees the remote system as a series of menus and menu items. Each menu item can be a file, a link to another menu, a telnet session,

When the author puts a file on the server, the author specifies the type of data the file contains. The Gopher client can try to present the data in the most appropriate manner. For example, if the file was PostScript, the Gopher client could ask the user if they wanted view it using a PostScriptpreviewer.

Gopher allows the author to create long mnemonic names for documents and to create the equivalent of a table of contents for that directory. It also supports freeWAIS indices, so that the contents of the files can be indexed and searched rapidly by the users.

The Gopher clients provide graphical user interfaces, and allow the user to create bookmarks that record the location of files for later reference. There are Gopher clients for character-cell terminals, DOS, Windows, Macintosh, and X.

Advantages:

- MIME types can be used to describe file formats.
- Documents can be browsed, saved, and printed without leaving the Gopher client.
- Non-ASCII documents or files can be downloaded.
- External programs for viewing non-ASCII files can be defined.
- Any VT100-compatible terminal can use Gopher.
- Graphical clients are available for Microsoft Windows, Apple Macintosh, and X.

Disadvantages:

- Only the Gopher protocol is supported by most Gopher clients.

- Eighty column ASCII text is the best-supported format.
- Menus are the only format.
- Documents cannot contain links.

Z39.50 is the ANSI/NISO Z39.50-1992 ISO Search and Retrieval application-level protocol.

URL: file://ftp.loc.gov/pub/z3950

The Center for Networked Information Discovery and Retrieval (CNIDR) is developing sample implementations of Z39.50 version 3 server and client program for TCP/IP, which will be known as ZDist. The current sample clients and servers, which are based on Z39.50-1988, are known as freeWAIS.

The relevant ISO standard is ISO 10163.

The primary disavantage of Z39.50 is that the current free implementations are rather difficult to install and configure, and that the protocol is not yet in wide use.

WWW World Wide Web began as an experimental hypertext system developed by researchers at the *European Laboratory for Particle Physics* AKA *Conseil Europeen pour la Recherche Nucleaire* (CERN) to support High Energy Physics research activities. The researchers invented a new protocol called HTTP and a document markup called HTML. There is a draft RFC for the HTTP protocol.

URL: http://www.w3.org/hypertext/WWW/Protocols/

Like Gopher, WWW clients which employ HTTP and HTML allow users to browse through documents, view them, and save them. Unlike Gopher, WWW clients present the information as a document with embedded links, rather than as a menu. HTML documents can include inline graphics as well as text. It also supports forms via icons, buttons, and input fields.

Since the "World Wide Web is the universe of network-accessible information," which includes all information-sharing network protocols, the term WWW includes FTP and Gopher and NNTP, even though it is HTTP which catalysed its development.

Advantages:

- Line-mode, full-screen, and graphical clients are available.
- Tools exist to convert legacy formats to HTML.
- Most HTTP clients also talk a subset of other WWW protocols like Gopher, NNTP, and FTP.

Disadvantages:

- Documents with large unnecessary pictures waste network bandwidth and disk space.
- The protocol is still evolving rapidly.

Security The current generation of Internet protocols have little support for such security concerns as confidentiality and authentication. It is relatively easy for one system to masquerade as another, or for one system to eavesdrop on the network traffic produced by other systems on the same network segment. Many people are interested in using WWW protocols, particularly HTTP, to conduct business over the Internet. To do this in a secure manner, at the minimum, you need some of the following services:

- A way to authenticate the server to the user, and the user to the server.
- A way to transmit confidential information between the two parties.
- A way to provide non-repudiability for transactions, i.e., a way to prove that a transaction has taken place.

There have been several proposals to add security to HTTP. Secure HTTP (S-HTTP)

URL: http://www.eit.com/projects/s-http,

from Enterprise Integration Technologies, uses security mechanisms like those used to provide secure email. Secure Sockets Layer

URL: http://home.netscape.com/info/security-doc.html,

from Netscape Communications, Inc., would allow any TCP/IP client to communicate securely with its server. In each case, the security is provided at the application layer of the protocol, rather than the lower transport layers. Terisa Systems

URL: http://www.terisa.com/

is working on a toolkit that combines S-HTTP *and* SSL.

2 Services

These freely available software components are used by the platform in addition to the services provided by the Distributed System Platform Profile. Unless otherwise noted, the software is written in C.

Using freely available software, based on open systems standards, is required for practical purposes. It would be difficult to prototype and experiment with these servers if they had to be purchased and licensed. In any case, most of the servers are not yet available commercially.

2.1 Electronic Mail

Historically in the UNIX environment, electronic mail was supported via UUCP, which is a point-to-point dial-up protocol. As Internet access becomes more common, use of the *Simple Mail Transfer Protocol* (SMTP) over TCP/IP has become the protocol of choice. Most UNIX systems come with a bundled SMTP-capable Mail Transfer Agent (MTA), such as Sendmail, MMDF, Smail, or PP. Any of these are suitable for an InfoServer system. The relevant specification for SMTP is Internet RFC 821.

URL: gopher://ds.internic.net/00/rfc/rfc821.txt

Users need Mail User Agents (MUA). These are used to read and write email messages, and to maintain private mailing lists. UNIX mail has been limited to 7-bit ASCII messages. Programs like **uuencode** have been used to translate binary files to a mailable ASCII format.

To compensate for the shortcomings of SMTP, and the older message format standard, the Multipurpose Internet Mail Extension (MIME) draft standard (RFC 1521) was written:

> This document redefines the format of message bodies to allow multi-part textual and non-textual message bodies to be represented and exchanged without loss of information.
>
> ...
>
> In particular, this document is designed to provide facilities to include multiple objects in a single message, to represent body text in character sets other than US-ASCII, to represent formatted multi- font text messages, to represent non-textual material such as images and audio fragments, and generally to facilitate later extensions defining new types of Internet mail for use by cooperating mail agents.

The MIME system for describing file types has been adopted by the Gopher and WWW developers as well.

In addition the the MTA, the InfoServer system needs software which can handle mailing lists more conveniently than the MTA's. It also needs to be able to respond to requests for archives of mailing list traffic, or requests for text and binary files which are made available to the general public. One popular package for UNIX is **ListProcessor**, written by Anastasios Kotsikonas:

> This is a system that implements various mailing lists with one list server. It is automated, and obliterates the need for user intervention and maintenance of multiple aliases of the form "list, list-owner, list-request", etc. There is support provided for public and private hierarchical archives, moderated and non-moderated lists, peer lists, peer servers, private lists, address aliasing, news connections and gateways, mail queueing, digests, list ownership, owner preferences, crash recovery, batch processing, configurable headers, regular expressions, archive searching, and live user connections via TCP/IP.

Another is Majordomo, written by Brent Chapman, which is implemented in Perl.

2.2 Anonymous FTP

The FTP daemon which is bundled with most Internet servers can be used by users with regular account to put and get files from the server system. If an account called `ftp` is created, and its home directory is set-up appropriately, other users on the Internet who do not have accounts on that system can potentially put and get files from that home directory,

or one of its subdirectories, depending on the directory permissions. The FTP user logs in as `ftp`, and usually gives their email address as a password. From then on, they are restricted to the special FTP directory tree.

Many sites use an enhanced FTP server from Washington University. This server allows the administrator to specify different classes of users, such as local or remote, and can be configured to limit the number of users by day of the week or time of day. Password-protected directories can be created to allow selected users to access documents. If a file called `.message` is created in a directory, the contents of that file will be displayed when the user changes to that directory. If a file called `README` is created in a directory, the user will be informed of the last date that file was updated.

2.3 Gopher

The Gopher server listens for requests from Gopher clients. Each request is an atomic transaction. The Gopher server does not have to keep track of the current directory or file type, and doesn't need to keep track of who's logged in and who's logged out. This stateless protocol allows a Gopher server to handle more transactions with less overhead than an FTP server.

There are several Gopher servers available. The UMinn server is the most widely used.

When the UMinn server first opens a directory, it reads the directory contents, parses any `.names` or `.Links` files, and caches the information in `.cache` files in each directory. Thereafter, the server uses the `.cache` information without re-reading the directory. To keep the caches and the directory contents in sync, the caches are updated at pre-determined intervals.

The John Frank's GN server works slightly differently. The author or adminstrator creates a file, usually called `menu` in each directory. This file corresponds to the `.names` file used by the UMinn server. After creating the `menu` file in each directory to be served, the administrator runs the `mkcache` utility which parses the `menu` files and creates the cache files. The advantage of this approach is that an administrator can hide files that the FTP server might need, but that the Gopher user needn't see. The disadvantage is that new files don't become visible until the `mkcache` program is re-run. The GN server also supports the HTTP protocol, so that a site whose main method of access is Gopher doesn't have to run another server for HTML documents.

Each of the servers can be used with per-directory access control files, so that access to the can be limited to a specified Internet domain, e.g., `ncsl.nist.gov`.

The relevant specification for Gopher is RFC 1436.

URL: gopher://ds.internic.net/00/rfc/rfc1436.txt

2.4 Z39.50

The most widely used Z39.50-based software is freeWAIS (see A.5). The freeWAIS software includes source for a server, a command-line client and an X client, and full-text indexing. The most recent version allows the administrator to specify other indexing engines.

CNIDR will release ZDist, based on Z39.50-1992, in the near future.

Another package which makes a useful companion to freeWAIS is **Essence**. Essence tries to do some semantic interpretation of the files it encounters, rather than blindingly indexing all text. It will extract the files contained in archive files and analyze their contents. It is uses external programs to parse the files, which makes it easy to extend or modify the parsing.

Essence is now part of the Harvest Information Discovery and Access System.

URL: http://harvest.cs.colorado.edu/

> Harvest is an integrated set of tools to gather, extract, organize, search, cache, and replicate relevant information across the Internet. With modest effort users can tailor Harvest to digest information in many different formats, and offer custom search services on the Internet.

2.5 World Wide Web

The WWW server is a stateless, transaction-oriented server like Gopher.

CERN and the National Center for Supercomputing Applications (NCSA) have written two of the most popular servers. They have similar features, including forms, clickable images, authorization, automatic creation of hypertext views of directory trees, and server scripts.

There is also a variety of clients. CERN distributes a line-mode client suitable for hard-copy terminals. University of Kansas has written a screen-oriented client called **Lynx**, which is suitable for ANSI terminals. NCSA has created several popular WWW clients for X/Motif, Microsoft Windows, and Apple Macintosh known collectively as the Mosaic browsers (see A.7). Netscape sells the popular Netscape browsers, and has aggressively added support for proposed HTML features.

Most of the clients have the ability to run an external program to view file types which can not be displayed natively within the client. For example, a sound file can be sent to a sound-player program, or a PostScriptfile can be displayed with a PostScriptpreviewer like **GhostScript**.

WWW uses an ASCII markup format called the HyperText Markup Language (HTML). HTML is used to indicate the logical structure and representation of the document. For example, `<H1></H1>` is used to mark a first level heading. The HTML format was inspired by SGML, the ISO standard *Standard Generalized Markup Language*, and there are several versions of SGML Document Type Defitinions (DTDs) for HTML, but currently most HTML editors and browsers cannot be considered conformant SGML applications.

The HTML markup can be added to ASCII documents by using an ASCII editor, or documents existing in another format, such as LaTeX or WordPerfect, can be converted to HTML.

The links in an HTML document can be to other parts of the same document, to other documents on the same system, or to other documents on other systems.

There is an ISO standard which supports hypertext called **HyTime: ISO 10744 Hypermedia/Time-based Structuring Language** but there are relatively few implementations. HyTime is an SGML application. A special interest group is said to be working on a freely distributable implementation. In the future, it is possible that HTML will become a proper subset of HyTime. The relevant standard for SGML is ISO/IEC 8879:1986, Information Processing Text and Office Systems Standard Generalized Markup Language. There is a draft RFC for the HTML definition.

3 Installation and Operation

3.1 Platform Requirements

3.1.1 Hardware

A minimum hardware configuration would include the following:

- CPU that supports multiprocessing and memory protection
- Sufficient memory for the operating system, server software, and logged in users using various application software.
- Sufficient disk space for the operating system, information and indices, and any user files.
- Backup device capable of backing up the operating system, users files, server software, and information. Typical choices would be 8mm or 4mm DAT tape drives, or magneto-optical disks.
- Ethernet interface(s)

If the server will also be used as a authoring workstation, the following components are desirable:

- a high-resolution graphics adapter
- color display
- keyboard and mouse

Other hardware which could be acquired as needed:

- Compact Disk drive/jukebox (if information will be served from CD's)
- Magneto-optical disk drive (for near on-line archives)
- High-speed modems (for dial-in connections, SLIP and PPP)
- Scanner (for scanning documents which are not available in machine-readable formats)

3.1.2 Software

The following specifications from the Distributed System Platform Profile are the minimum necessary for an InfoServer.

POSIX is the logical choice for an operating system interface. The system must be multi-user and multi-tasking, to support multiple clients and servers. Although some of the servers mentioned below have been ported to proprietary PC OS's, generally, those platforms can only support a single server process. The relevant POSIX standards are FIPS 151-2

URL: http://nemo.ncsl.nist.gov/fips/151-2.txt

and ISO/IEC 9945-1

ANSI C is becoming the *de facto* standard for implementing system software in the POSIX environment. In the interests of portability, much of the software continues to be written in a subset of ANSI C, generally known as K&R (for Kernighan and Ritchie). However, as more ANSI C-compliant compilers become available, developers are gradually incorporating elements of ANSI C into their software. As a result, having an ANSI C compiler is often necessary to compile the freely available software used to implement a system. The relevant C compiler standard is ISO/IEC 9899:1990

Some computer systems vendors do not bundle an ANSI C-compliant compiler with the operating system. In that case, the developer must buy an ANSI C development environment from a vendor, or obtain the freely available GNU C compiler (see A.1).

TCP/IP is the *de facto* networking standard for UNIX environments. Given that TCP/IP protocol stacks and libraries are usually bundled with the operating system, developers usually don't need do any extra work obtaining it. Source code is available for the Berkeley implementation of TCP/IP. The relevant TCP/IP networking specifications are Internet RFC 791

URL: gopher://ds.internic.net/00/rfc/rfc791.txt

for IP, and Internet RFC 793

URL: gopher://ds.internic.net/00/rfc/rfc793.txt

for TCP.

Berkeley Sockets API which is abstraction on top of TCP/IP, is used by most of the client/server applications. Like TCP/IP, the socket libraries are a *de facto* standard in UNIX environments, and are usually bundled with the operating system, and source code is available. Socket libraries are also available for Apple Macintosh and Microsoft Windows OS's.

The IEEE P1003.12

URL: http://nemo.ncsl.nist.gov/P1003.12/

working group is working on *de jure* standard socket and TLI API's.

3.1.3 Other Standards and Specifications

These standards and specifications are useful for the information provider. Although any data format can be used, choosing a widely supported one makes the data more useful.

XBM The **X BitMap** format is the common standard for representing bitmapped images for the X Window System. Most graphical WWW clients can render XBM images inlined with text in an HTML document. By default, the white areas of the image are usually set to the background color of the client window, making the image appear to be "printed" on the page.

GIF The **G**raphics **I**nterchange **F**ormat is a popular compressed image format developed by CompuServe, Inc. Most graphical WWW clients can render GIF images inlined with text in an HTML document. GIF is best for simple images with 256 colors or less. Recently, UNISYS has indicated that developers of GIF software that use UNISYS's patented LZW compression must pay royalties. As a result, alternatives to GIF are being pursued by the WWW community. One such is the PNG (**P**ortable **N**etwork **G**raphics) format.

URL: http://quest.jpl.nasa.gov/PNG/

JPEG The **J**oint **P**hotographic **E**xperts **G**roup developed a set of lossy compression techniques for continuous tone image data. Images compressed with JPEG techniques are usually much more compact than GIF images. JPEG is most appropriate for scanned or rendered pictures with more than 256 colors. The document is ISO/IEC JTC1 Committee Draft 10918-1 February 1991. Most graphical WWW clients can now render JPEG images inline.

MPEG The **M**otion **P**icture **E**xperts **G**roup developed a set of compression techniques for full motion video. MPEG will probably replace the host of proprietary video data formats. The document is ISO/IEC 90/176, December 1990 11172.

PostScript Adobe's PostScriptis the most popular page description language. Most graphical WWW clients cannot render PostScriptimages inline.

URL: http://www.adobe.com/PS/

SGML is the ISO Standard General Markup Language. With the advent of WWW, interest in SGML, and HTML as an application have increased markedly.

URL: http://www.w3.org/hypertext/WWW/MarkUp/SGML.html

HTML is the evolving WWW hypertext mark-up language. Version 3 is currently on the drawing board. The original HTML was merely similar to an SGML-style markup language, while the later versions have formal SGML Document Type Definitions (DTD's).

URL: http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html

MIME Recently, the draft **M**ultipurpose **I**nternet **M**ail **E**xtensions (MIME) RFC has become popular. This draft Internet standard provides a way to create multi-part messages which can contain data in various formats. MIME provides a standard way of specifying the type of data, and the encoding (if any) used to make it mailable. Gradually, more MUA's have started incorporating support for reading and writing MIME-compliant mail messages. The relevant specifications for MIME are RFC's 1521,

URL: gopher://ds.internic.net/00/rfc/rfc1521.txt

1522,

URL: gopher://ds.internic.net/00/rfc/rfc1522.txt

1523,

URL: gopher://ds.internic.net/00/rfc/rfc1523.txt

and 1524.

URL: gopher://ds.internic.net/00/rfc/rfc1524.txt

Digital Signatures The federal standard is FIPS180-1, the Secure Hash Standard.

URL: http://csrc.ncsl.nist.gov/fips/fip180-1.txt

The most popular U.S. commercial algorithm is MD5,

URL: gopher://ds.internic.net/00/rfc/rfc1321.txt

from RSA Data Security, Inc.

URL: http://www.rsa.com/

3.2 Implementation Decisions

The easiest way to provide information is to configure all the servers to serve information from the same directory tree. The disadvantage is that some of the support files required by one server, which are usually hidden from the users, may be exposed by another server. In general, this is merely unsightly, not insecure.

There are two approaches to managing the files. The first approach is to have a single account (let's call it the Librarian) be in charge of all the files. The Librarian will obtain the files from the authors, make sure they are in the appropriate formats, and put the files in the appropriate places in the directory hierarchy. The advantage of this approach is there

need be only a single interactive user account, and that user is responsible for ensuring that any organizational rules about providing information have been followed. The Librarian is also responsible for seeing that the table of contents and/or index is kept up-to-date. The disadvantage is that Librarian becomes a bottleneck in the process, and if being Librarian is not a full-time or first-priority activity, authors may not be able to make their documents available in a timely manner.

The second approach is to delegate authority by allowing each branch of the directory tree to be owned and managed by the author, or by a subject-area manager. The advantage of this approach is that files can be added and removed without the intervention of the Librarian. This means that, aside from routine maintenance, the Librarian does not have to be a full-time position. The disadvantages are that authors have to learn more about how to create tables of contents, and what are appropriate file formats. And since it is relatively inconvenient to manipulate the files remotely, the authors will need interactive accounts on the server. The presence of multiple interactive accounts increases the chance that the authors will inadvertently degrade the servers' operation, and increases the administrative load on the system manager.

The second approach would be more convenient is automated and secure tools existed to allow authors or subject-area managers to remotely manage their documents without needing accounts on the server. We are investigating some tools which could be modified to support this approach.

3.2.1 Data Formats

Most organizations serve data in a variety of formats, which causes problems for the authors and the readers. Here are some of the common formats with their attendant advantages and disadvantages:

Seven-bit ASCII is the easiest to maintain and download, but all printing and graphical information is lost. It is the most convenient format for providing text that will be inserted into other people's documents. Seven-bit ASCII can be emailed without encoding.

Using a popular wordprocessor for example, WordPerfectdocuments can be saved as Text, but the formatting will look better (including tables) if you select a daisywheel printer, and `Print To File`.

ASCII with markups, such as Troff, TeX, and SGML, is as easy to maintain as plain ASCII, and allows users to easily insert text into other documents. The markup languages usually support simple line graphics. Their chief disadavantage is that potential users must be familiar with the relevant markup languages, and must have access to the appropriate text-processing programs to print out the documents. The HyperText Markup Language (HTML), which is an SGML DTD, has the advantage of being browsable via WWW browsers such as NCSA's Mosaic.

PostScript is appropriate for finished documents whose appearance is very important, and for documents containing graphics. PostScriptcan usually be emailed without encoding, however some wordprocessors create PostScriptfiles that contain lines that are too long to email. One drawback is that potential users must have access to a PostScriptprinter. Another is that the PostScriptoutput for even small documents can be very large.

PDF is an alternative to PostScript. It allows users to read and annotate documents. One drawback is that potential users must have a PDF viewer on their system. PDF documents are created by running `Distill` on an existing PostScriptdocument, or by selecting a PDF printer from the Windows or Macintosh print menu.

WordPerfect, or other wordprocessor formats are appropriate when a group of users has agreed to use a particular wordprocessor, or the original document is only available in that format. Most wordprocessor formats don't use seven-bit ASCII, so they cannot be emailed. WordPerfecthas an ASCII encoding mechanism, but it is little used. Microsoft's Rich Text Format (RTF) is ASCII, but applications which export RTF usually create lines that are too long to be emailed. As a result most wordprocessor formats must be encoded by the provider, and decoded by the users. Some of the common encoders are MIME's `base64`, `UUENCODE`, `XXENCODE`, `binhex` and `btoa`.

Archive formats are appropriate when a document is large, or consists of a number of smaller files. The files are bundled as a package, and may be compressed as a group, or individually within the archive. Some of the most common formats are `tar` with `compress`, `ZIP`, `zoo`, and `ARC`. The primary advantages of these formats are that they make handling a group of files easier, and the compression can make the files much smaller. The primary disadvantages are that they are not emailable without being encoded, and since the archive files are often fairly large, the archive must be split up into multiple email messages.

When an organization is moving their data from a BBS to a open system, the proprietary BBS index or file listing formats will need to be converted to ones usable by Gopher and/or WWW .

3.2.2 Email Lists and File Retrieval

Email lists can be handled using whatever the default MTA provides, or email list maintenance software can be installed. If email file retrieval will be provided, the appropriate software will need to be installed.

3.2.3 FTP

The root of the tree should be owned by the FTP daemon. The data directories are created in the `ftp/pub` directory. All the other servers will use that directory as their root directory. A `.message` file can be created in each directory which will be shown to the FTP user, and

which usually describe the contents of the directory. If there is a `README`, the user will be told how recently that file has been updated and asked to read it.

3.2.4 Gopher

By default, Gopher will serve files in a directory like FTP, that is, it just lists the names of the files. If you are using the `describe` feature of the UMinn Gopherd, you will need to go into each directory and describe each file. Otherwise, you can create a file, usually called `.names`, in each directory which maps a long mnemonic name to the shorter file names usually used by authors. Details on creating the `.names` files can be found in the `Gopherd.1` manual pages.

You can create long file names like: Annual_Summary_of_Web_Server_Statistics, but FTP users find it annoying to type them in.

If you use the GN Gopher server, you can use the supplied utility to create the initial file name databases that GN uses.

3.2.5 freeWAIS

Once the freeWAIS server is configured and installed, the only other step is to choose which parts of the directory hierarchy will be indexed. You can index all or parts of the hierarchy. You can choose to index some file types and not others. There is also support in freeWAIS for indexing the same file in different formats, e.g., text, PostScript, and GIF. Once the indexing strategy is decided upon, a `crontab` script can be used to update the indices at the desired intervals.

3.2.6 World Wide Web

Both the CERN and the NCSA Web servers can serve a directory as if it was an FTP or Gopher directory. If the files in a directory will change frequently, or there's no need for fancy presentation, that is any easy way to provide the service.

To take advantage of all the features of WWW, you will need to create HTML documents that either contain the information to be server, or links to other files containing that information. The documents can be created using an ordinary text editor, or an editor that provide some syntax support like the HTML mode of EMACS, or existing documents can be translated from other formats like LaTeX, Microsoft's RTF, or WordPerfect. There is a variety of free translators available.

`latex2html` does a particulary good job of translating LaTeX documents into HTML. Since the current version of HTML doesn't have support for tables or equations, `latex2html` automatically takes LaTeX tables, equations, and pictures, and creates embedded images. It also gives you control over the nesting depth of the hypertext document it creates, so that you can tell it to create one big HTML file, or multiple nested files, depending on the LaTeX section directives.

3.3 Ongoing Management

- If you have delegated authority to the authors, there will be no additional work for you within the hierarchy, except for resolving user complaints about unreadable files or dangling hypertext links. These complaints can be forwarded to the appropriate author.

- If you have WAIS indices, you will want to re-index the tree periodically. If your files change infrequently, you can add new files to the index without re-indexing the entire tree. If your tree is relatively small, you could re-index it nightly. Or you can search directories for new or changed files, and just re-index that subtree.

- You will probably want to run one of the programs that will summarize usage for each of the protocols, and for the various subtrees. One program called `wusage`

  URL: http://siva.cshl.org/wusage.html

  which summarizes HTTP logs, also draws a line graph summarizing usage, and a pie chart that displays downloads by Internet domain.

- If your access logs are growing rapidly, you may want to extract and compress the logs each month.

- You can solicit user comments or requests for more information by creating a WWW HTML form. Scripts to handle the forms can be written in almost any language, include C, Bourne Shell, Perl, and Tcl. The user fills out the form, and the contents can be mailed to the appropriate person.

  The current version of Mosaic supports adding personal annotations to WWW documents. Future versions of the WWW HTTP clients and servers will support group annotations. This will let readers see others readers' comments on the documents.

3.4 Case History: The System and Software Technology Division's InfoServer

The System and Software Technology Division's first step toward an electronic forum was using **electronic mailing lists** on the Distributed Systems Engineering Group's UNIX server to send information regarding various division programs, e.g., POSIX Conformance Testing, to interested parties.

The next step was to provide an **email archive server** which could automatically respond to requests for the POSIX Conformance Testing register. This decreased the need to fax updated versions to requestors. Available free software was surveyed to implement this system. A system written in an freely available interpreted language called Perl (see A.1)was selected.

At the time, **anonymous FTP** was ruled out because many of the prospective users didn't have Internet access, and the division server would have been more vulnerable to security

problems. A dial-in BBS was ruled out because of the additional time and effort required to maintain one, the lack of robust free software, and the expense of external telephone lines.

When the OSI Implementors Workshop (OIW) activity was moved from the Systems and Network Architecture Division to the Systems and Software Technology Division, the need for anonymous FTP arose, since the users of the OIW had become accustomed to getting document drafts via FTP.

The first dedicated information server was set up on a underutilized UNIX system. The mail archive server and an enhanced FTP server were installed. Both servers served files from the same directory hierarchy, eliminating the need for duplicate files. When **Gopher** became available, a Gopher server was installed and configured to use the same directory hierarchy. Later, the WWW server was installed and configured to use the same directories. By putting a file in the appropriate directory, it instantly became available via any of the services. The **freeWAIS** software was installed, but the system lacked the disk space for the indices. Eventually, hardware failures on that UNIX system resulted in a search for a new system.

The next system was an INTEL 486-based system running a free, experimental UNIX-like system called **Linux**. Unfortunately, bugs in an early revision of its TCP/IP networking software made the system unstable. Rather than waiting for an indeterminate time for the bugs to be fixed, the division's server systems were re-configured, and a Sun Sparcstation was made available.

The email, Gopher, WWW and freeWAIS server software was moved to the new system, re-configured and re-compiled. The directories containing the information were moved to the new system. The OIW mailing lists were moved, and accounts for information providers were created. The freeWAIS indexing program was run on the contents of the directories to make them searchable.

4 Conclusions

Setting up a distributed information server is still far from easy. Each of the software packages for a particular service stands alone, which makes it easy to install a single service, but difficult to install and coordinate multiple services. One solution would be to choose one or two services, and eliminate the others, since there are large areas of overlap between the services. Another solution would be to design a common information service backend that could be accessed via various protocol frontends.

Recently, commercially-produced HTTP servers and clients have become available, which offer easier installation and maintenance, as well as better security, and sometimes higher performance. They still tend to lack support for indexing and searching, and integrating them with email and FTP is a do-it-yourself proposition.

A Description of Software Sources

Reader's Note: The `.tar` file name extensions means that the file is a TAR archive containing multiple files. The `.Z` extension means that the file has been compressed using the UNIX `compress` utility. Some archive sites may be using the GNU `gzip` compression utility, in which case the file names will end with `.gz`. Sometimes a TAR file which has been compressed with `gzip` has the extension `.tgz`. The locations and version numbers were current the last time this document was edited. However, given the volatilitly of the WWW and the ongoing development of the software, they are probably out-of-date by the time this is published.

A.1 ANSI C Compiler and Miscellaneous Utilities

The GNU C compiler, and other useful GNU utilities are available via anonymous FTP from

URL: file://prep.ai.mit.edu/pub/gnu/

and other archive sites.

The GNU C Compiler is discussed on Usenet.

URL: news:gnu.gcc.announce

gcc-2.6.3.tar.Z The GNU C compiler in compressed TAR format.

find-3.8.tar.Z The GNU find utility. Includes the `locate` utility used by the Squirrel mail archive server.

gzip-1.2.4.tar.Z GNU file compressor.

Perl Perl was written by Larry Wall `<lwall@netlabs.com>`. From the UNIX manual page:

> Perl is an interpreted language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It's also a good language for many system management tasks. The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal). It combines (in the author's opinion, anyway) some of the best features of C, sed, awk, and sh, so people familiar with those languages should have little difficulty with it. (Language historians will also note some vestiges of csh, Pascal, and even BASIC-PLUS.) Expression syntax corresponds quite closely to C expression syntax.

Source for perl is available from ftp.uu.net,

URL: file://ftp.uu.net/pub/languages

and other archive sites.

Perl is discussed on Usenet:

URL: news:comp.lang.perl

A.2 FTP

Washington University Archive FTPD The Washington University Archive enhanced FTP daemon is available via anonymous FTP from Washington University,

URL: file://wuarchive.wustl.edu/packages/wuarchive-ftpd

and other archive sites. The most recent version is wu-ftpd-2.4.tar.Z

Archive administration

URL: news:comp.archives.admin

is discussed on Usenet.

A.3 Electronic Mailing List Management and File Retrieval

A list of frequently asked questions regarding Mail Archive Servers.

URL: news:news.answers

is posted regularly to Usenet.

Squirrel Mail Server The Squirrel Mail Server was written by Johan Vromans <jv@mh.nl>. It can be obtained via email by sending a mail message to <mail-server@nluug.nl> with the contents:

```
begin
send mail-server
end
```

Its implementation language is Perl.

ListProcessor The ListProcessor was written by Anastasios Kotsikonas

URL: mailto:tasos@cs.bu.edu

It is available via anonymous FTP.

URL: file://cs-ftp.bu.edu/pub/listserv/

Information on a commercial version is also available.

URL: http://listproc.net/.www/listproc.html

Majordomo Majordomo, copyright Great Circle Associates, is available via HTTP

URL: http://www.greatcircle.com/majordomo/

or FTP.

URL: ftp://ftp.greatcircle.com/pub/majordomo/

A.4 Gopher

Gopher2.016.tar.Z The University of Minnesota UNIX Gopher server and client.

Source and executable versions of the University of Minnesota server and clients are available.

URL: gopher://boombox.micro.umn.edu:Gopher

The University asks that commercial entities using the software for internal use only negotiate licensing fees.

There is an experimental Gopher client which uses a 3D scene rendering instead of menus, called GopherVR, which is available via

URL: gopher://boombox.micro.umn.edu/11/Gopher/Unix/GopherVR

and

URL: ftp://boombox.micro.umn.edu/pub/Gopher/Unix/GopherVR

sgopher0.3.tar.Z is a restricted Gopher client which is also available at the University of Minnesota site.

gn-2.22.tar.Z John Franks' GN Gopher server and support utilities. John Franks' GN Gopher server, which supports both the Gopher and WWW HTTP protocols, and has no licensing requirements, is available via anonymous FTP,

URL: file://ftp.acns.nwu.edu/pub/gn

or Gopher.

URL: gopher://hopf.math.nwu.edu:70/

The most recent version as of this writing is `gn-2.22.tar.Z`.

Gopher

URL: news:comp.infosystems.Gopher

is discussed on Usenet.

A.5 Searching

freeWAIS-0.4.tar.Z Source for the freeWAIS clients and servers is available via anonymous FTP.

URL: file://ftp.cnidr.org/pub/NIDR.tools/freewais

The most recent version as of this writing is freeWAIS-0.4.tar.gz

URL: file://ftp.cnidr.org/pub/NIDR.tools/freewais/freeWAIS-0.4.tar.gz

Harvest Source for the Harvest Information Discovery and Access System is available via FTP.

URL: ftp://ftp.cs.colorado.edu/pub/distribs/harvest

A.6 WWW Servers

Source for the CERN server and clients is available via HTTP

URL: http://www.w3.org/pub/src/

Source for the NCSA server is available via anonymous FTP

URL: file://ftp.ncsa.uiuc.edu/Web/ncsa_httpd

WWW

URL: news:comp.infosystems.www.provider

is discussed on Usenet.

WWW*.tar.Z The CERN WWW libraries, server, and line-mode browser.

httpd_1.4.tar.Z The NCSA version of the WWW server. Binaries for selected architectures are also available.

A.7 WWW Clients

Executables for the NCSA Mosaic clients are available via anonymous FTP

URL: file://ftp.ncsa.uiuc.edu/Web/Mosaic-binaries/

The most recent version for X as of this writing is version 2.5

URL: file://ftp.ncsa.uiuc.edu/Mosaic/Unix/

Executables for Microsoft Windows

URL: file://ftp.ncsa.uiuc.edu/Mosaic/Windows/mos20b4.exe

and Apple Macintosh

URL: file://ftp.ncsa.uiuc.edu/Mosaic/Mac/NCSAMosaic200B5.FAT.hqx

are also available.

Mosaic-2.5.tar.Z The NCSA X Window System WWW browser. Requires OSF Motif, libraries. Binaries for selected architectures are also available.

chimera-1.65.tar.Z Chimera is an X11 WWW browser, written by John Kilburg of the University of Nevada, Las Vegas.

URL: mailto:john@cs.unlv.edu

It uses the Athena libraries, rather than Motif or XView.

Its source is available via anonymous FTP

URL: file://ftp.cs.unlv.edu/pub/chimera/chimera-1.65.tar.gz

and it has a home page.

URL: http://www.unlv.edu/chimera/

A.8 HTML Conversion

latex2html.tar.Z (version 5.3) Translates LaTeX files to HTML. Automatically creates section links and inline images for diagrams and equations. Latex2html was written by Nikos Drakos `nikos@cbl.leeds.ac.uk`. It can be obtained via FTP from that site.