NIST Special Publication 500-196

Computer Systems Technology

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology

NIST

Guidelines for the Evaluation of File Transfer, Access and Management Implementations

Paul Markovitz
Steve Trus
Curtis Royster

NIST Special Publication 500-196

Guidelines for the Evaluation of File Transfer, Access and Management Implementations

Paul Markovitz
Steve Trus
Curtis Royster

Systems and Network Architecture Division
Computer Systems Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

This Report Was Prepared Under Funding Provided by the U.S. Army Information Systems Command, the U.S. Air Force Communications Command, and the Internal Revenue Service.

October 1991

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

Reports on Computer Systems Technology

The National Institute of Standards and Technology (NIST) has a unique responsibility for computer systems technology within the Federal government. NIST's Computer Systems Laboratory (CSL) develops standards and guidelines, provides technical assistance, and conducts research for computers and related telecommunications systems to achieve more effective utilization of Federal information technology resources. CSL's responsibilities include development of technical, management, physical, and administrative standards and guidelines for the cost-effective security and privacy of sensitive unclassified information processed in Federal computers. CSL assists agencies in developing security plans and in improving computer security awareness training. This Special Publication 500 series reports CSL research and guidelines to Federal agencies as well as to organizations in industry, government, and academia.

National Institute of Standards and Technology Special Publication 500-196
Natl. Inst. Stand. Technol. Spec. Publ. 500-196, 107 pages (Oct. 1991)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1991

For sale by the U.S. Government Printing Office
Superintendent of Documents, Mail Stop: SSOP, Washington, DC 20402-9328
ISBN 0-16-035984-8

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

In August 1990, the Government Open Systems Interconnection Profile (GOSIP, FIPS 146) mandated that Federal agencies requiring to transfer, access, and manage files procure products conforming to the File Transfer, Access and Management (FTAM) International Standard (ISO 8571). Version 2 of GOSIP (FIPS 146-1) was promulgated in April 1991. Version 2 of GOSIP supersedes Version 1 of GOSIP and should be used as the sole GOSIP reference. Version 2 of GOSIP does not change the procurement mandates specified in Version 1. The GOSIP Users' Guide (NIST Special Publication 500-163) advances the goals of the GOSIP by providing government technical, managerial, and procurement personnel with the information they need to acquire and use GOSIP compliant products. This document, the Guidelines for the Evaluation of File Transfer, Access and Management Implementations, also advances the goals of the GOSIP by providing guidelines for evaluating FTAM implementations. These guidelines can assist a user in determining which implementation, among several candidates, will best meet the functional and performance requirements of that user. This document is one in a series of evaluation guidelines documents for Open Systems Interconnection (OSI) applications. It has been preceded by evaluation guidelines for Message Handling Systems (MHS) implementations. Evaluation guidelines for other OSI applications, such as Virtual Terminal (VT) and Directory Services, will follow.

The philosophy of the FTAM Evaluation Guidelines is explained in an analogy. If Samantha Smith selects a new car based on a "gut" feeling such as how the car looks or how much fun it is to drive, rather than on concrete facts, she may later find that she did not purchase the "best" car for her needs, and may be disappointed with her purchase. Likewise, if Michele Michaels selects an FTAM implementation based on a "gut" feeling such as the implementation's initial appearance, rather than on concrete facts, she may later find that she did not purchase the "best" FTAM implementation for her needs, and may be disappointed with her procurement.

A more logical approach to the problem of purchasing a car which best meets the users' needs is to: (1) Determine the type of car to be purchased, e.g., sedan, sports, van, etc. The type of car can be determined by examining the purposes for which the car will be used, e.g., to drive one person to work, to drive an entire family to various places, to carry packages home from the store, etc. (2) Make a list of cars which are candidates to be purchased. Initially, the list may contain all cars which the user would consider purchasing. After the user has determined any restrictive factors, such as price range and specific manufacturers which the user favors, the list will be narrowed to include only cars which meet the user's restrictive factors. (3) Create one list of functional characteristics of cars, and another list of performance measurements of cars. The user may obtain the information for the lists from product information provided by the manufacturer, magazines which evaluate cars, etc. The functional list would include concrete features such as the number of passengers that can ride in the car, how many cylinders the engine has, and the capacity of the gas tank. The performance list would include measurable features such as how fast the car accelerates from 0 to 60 miles per hour and how many miles per gallon of gas the car gets. (4) Create a list containing any functional characteristics and performance measurements which are required by the user. Cars which do not meet these requirements should not be further evaluated. For example, the user may require the car to get at least 25 miles per gallon of gas. Cars that do not meet this requirement are unacceptable to the user and will no longer be considered. (5) Assign weights to each of the functional and performance features to indicate their importance. For example, the user may consider the feature of how fast the car accelerates from 0 to 60 miles per hour to be of little importance, and therefore assign it a small weight. In contrast, the user may consider the feature of how many miles per gallon of gas the car gets to be very important, and therefore, assign it a large weight. (6) Score each of the cars by summing the availability of each functional feature times its weight and the measurement of each performance feature times its weight, for all of the features. The score for each car reflects how well it meets the requirements of the user. The car with the highest score is likely to be the "best" car for that user. Note that these ratings are not absolute ratings; another user might rate the same set of cars differently based on a different set of needs.

This document provides guidelines for evaluating FTAM implementations, using the approach defined for evaluating cars. Details are provided to guide the user through each step of the FTAM implementation evaluation process.

1.1. Methodology

This document contains: (1) guidelines for evaluating the functional specifications of FTAM implementations, (2) guidelines for measuring the performance of FTAM implementations, and (3) guidelines for matching the functional and performance specifications of an FTAM implementation to the functional and performance requirements of the user.

The evaluation guidelines are composed of the following components:

(1) An FTAM Configuration. The evaluation document provides guidelines for assisting the user in determining the most appropriate FTAM configuration.

(2) A list of candidate FTAM implementations. The user creates a list of the FTAM implementations which are candidates for procurement. The evaluation document provides guidelines for creating this list.

(3) A set of functions. The guidelines provide a set of the functions which may be available in an FTAM implementation. The user should become familiar with these functions, noting which ones are important to that user.

(4) A set of performance measurements. The guidelines provide a set of performance measurements which may be derived for an FTAM implementation. The user should become familiar with these performance measurements, noting which ones are important to that user.

(5) A set of user requirements. The user determines the user's set of functional and performance requirements. The evaluation document provides guidelines for determining this set.

(6) A rating formula. The guidelines provide formulas to calculate a functional, performance, and overall rating of each of the implementations being considered. The user should become familiar with these formulas.

1.2. Scope

These evaluation guidelines apply to implementations which have been produced according to the International Standard for File Transfer, Access and Management (ISO 8571), FTAM Phase 2 contained in Version 3 of the OSI Implementors Workshop (OIW) Stable

Implementors Agreements for Open Systems Interconnection Protocols, December 1989, and Version 1 of GOSIP.

1.3. Overview

The contents of this document are organized as follows: Section 1 contains an introduction to the document. Section 2 presents an FTAM tutorial. Section 3 specifies the procedure for evaluating FTAM implementations. The evaluation section provides guidelines for determining the FTAM configuration, creating a list of candidate FTAM implementations, functional evaluation guidelines, performance evaluation guidelines, guidelines for rating FTAM implementations based on their functional and performance rating, an example rating, and miscellaneous guidelines that do not pertain to functionality or performance. Section 4 summarizes the conclusions derived from the project. Appendix A reviews the FTAM laboratory used in this project. Appendix B contains a listing of the FTAM functions described in these guidelines, presented in a tabular form. Appendix C contains a listing of the performance experiments described in these guidelines, presented in a tabular form. Appendix D defines the abbreviations used in this document, and Appendix E provides a glossary of FTAM terms. Following the Appendices is a list of References.

1.4. Acknowledgments

The National Institute of Standards and Technology (NIST) wishes to acknowledge and thank the vendors who provided FTAM implementations to assist this project (Data General, Digital, Hewlett Packard, Novell, Retix, Wollongong, and 3Com). These implementations facilitated the development of this document. A diagram of these implementations, as configured in our FTAM Laboratory, is presented in Appendix A.

NIST also wishes to thank Network General and Novell for providing Protocol Analyzers, 3Com for providing OSI lower layer routers, Interactive Systems for providing UNIX operating systems, COMSAT for providing input to the FTAM performance experiments, and John Dempsey of Unisys for assisting with the FTAM tutorial.[1]

[1] Certain commercial products are identified in this report. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment identified is necessarily the best available for the purpose.

2. FTAM Tutorial

File Transfer, Access and Management (FTAM) is an OSI application that enables users to transfer, access, and manage files using direct peer-to-peer communications. File transfer enables a user to send a file to or receive a file from a remote system. File access allows a user to select a specific record in a file on a remote system, and file management allows a user to create and delete files, and to read and change the attributes of a file.

To further explain the concepts of file transfer, access, and management this tutorial is divided into the following sections: Section 2.1 describes FTAM roles. Section 2.2 reviews the FTAM regimes. Section 2.3 describes FTAM filestores. Section 2.4 describes FTAM File Access Data Units (FADUs), and sections 2.4 through 2.6 present a functional view of the FTAM service discussing FTAM actions, document types, and profiles respectively.

2.1. Roles

An FTAM implementation must act in at least one of the following roles: initiator-sender, initiator-receiver, responder-sender, and responder-receiver. The role of an initiator is to submit requests for FTAM services. The role of a responder is to passively answer the requests submitted by an initiator. The role of a sender is to transmit data, and the role of a receiver is to receive the data transmitted by a sender. The FTAM role configurations can best be described with an example. If FTAM implementation "A" requests to send a file to FTAM implementation "B," "A" acts as an initiator-sender while "B" acts as a responder-receiver. If implementation "C" requests to get a file from implementation "D," "C" acts as an initiator-receiver while "D" acts as a responder-sender. Figure 1 illustrates FTAM roles.

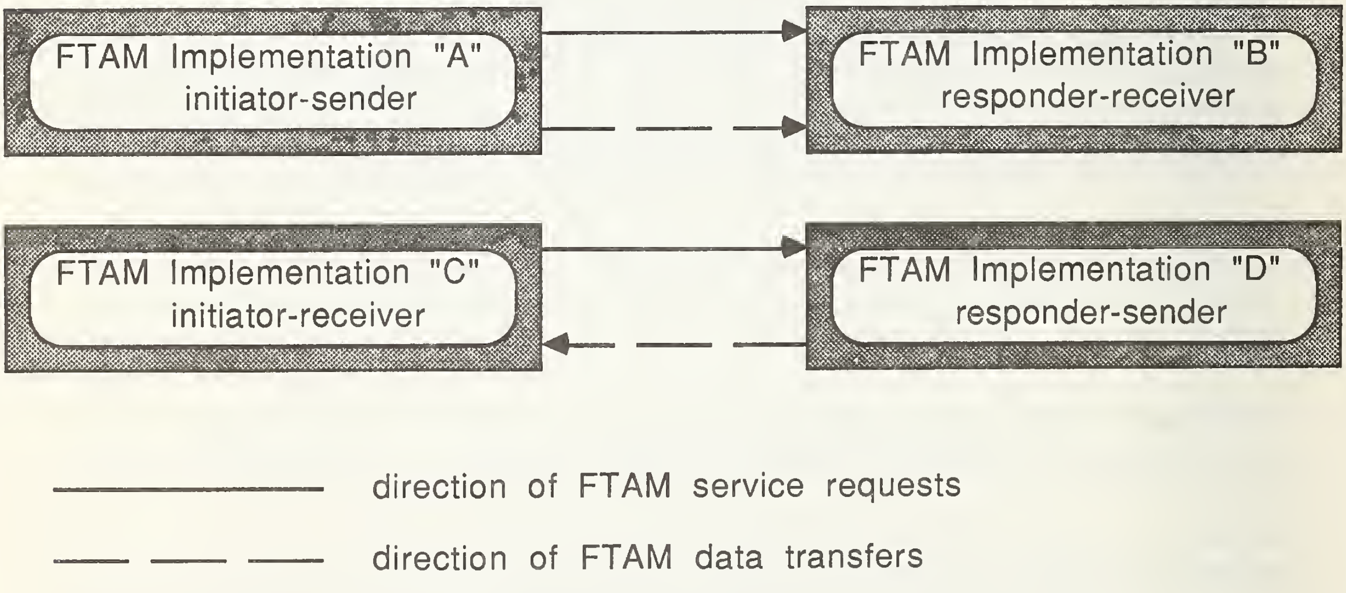


Figure 1. FTAM Roles.

2.2. Regimes

The FTAM service is provided in a series of stages, called regimes. A regime is a period during which specific actions are permitted. Figure 2 depicts the FTAM regimes.

The first regime in the FTAM service is called the FTAM regime. This regime establishes and releases the application association. The F-INITIALIZE service primitive is used to establish the regime. The F-TERMINATE service primitive releases the regime gracefully, and the F-ABORT service primitive releases the regime abruptly. During the establishment of the FTAM regime the initiator and responder negotiate various functions. For example, the initiator and responder may agree that during the association only files of a specific document type can be transferred. FTAM document types are discussed in section 2.6.

Once the FTAM association is established, filestore management may be invoked. This is where file directory services will be available in the near future. File directory services will enable a user to establish a current working directory and to change working directories.

The second regime is the file selection regime. Here, an initiator selects an existing file (F-SELECT), or specifies a file to be created, and selects the newly created file (F-CREATE). This regime is terminated by either deselecting (F-DESELECT) or deleting (F-DELETE) the selected file.

Once the file selection regime has been entered, file attributes can be queried or changed. The F-READ-ATTRIB service primitive is used to query file attributes (e.g., read a filesize). The F-CHANGE-ATTRIB service primitive is used to change file attributes (e.g., change a filename).

The third regime is the file open regime. In this regime, file contents may be accessed by the initiator. Actions on specific records within a file may be requested. For example, a record may be located (F-LOCATE) and/or removed (F-ERASE). The open regime is established with the F-OPEN service primitive and terminated with the F-CLOSE service primitive.

The final regime is the data transfer regime. This regime is established by the F-READ service primitive, if a file is to be received, or the F-WRITE service primitive, if a file is to be sent. Other data transfer service primitives include F-DATA, which is used to carry the data, F-DATA-END, which terminates the transfer of data, F-TRANSFER-END, which terminates the data transfer regime gracefully, and F-CANCEL, which terminates the data transfer regime abruptly.

2.3. Filestores

A real filestore is an organized collection of files residing on a computer system. From a user's point of view, the file system that exists on the same hardware as the user's computer system is considered the local filestore. File systems associated with other computer systems are considered remote filestores.

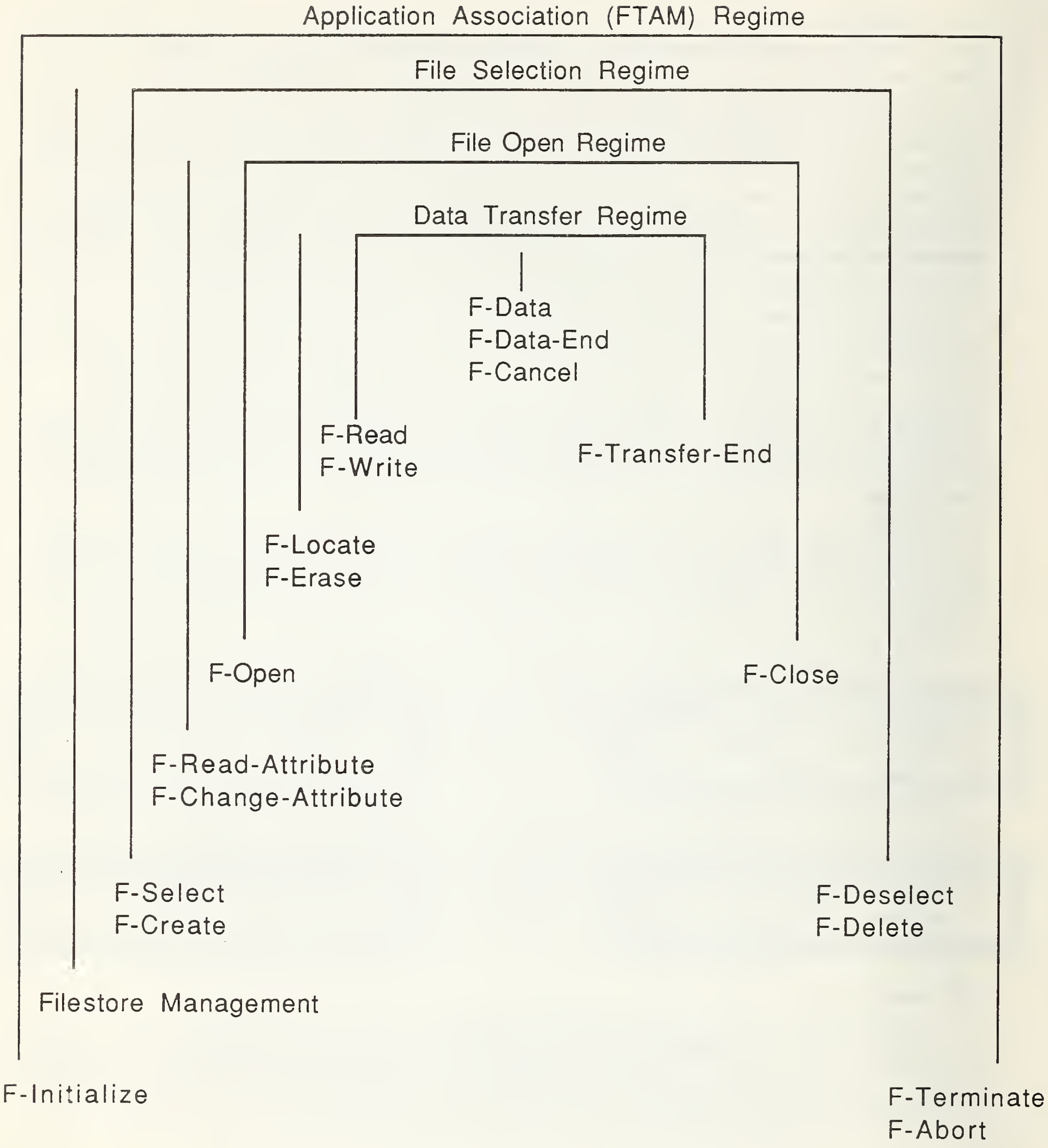


Figure 2. FTAM Regimes.

The ways in which filestores are implemented vary considerably between computer systems. Different systems have a wide range of methods for describing the storage of data and the means by which it may be accessed. The FTAM standard provides a common model for describing files and their attributes. This model is called the "virtual filestore". The virtual filestore allows files, which may be specified differently by different computer systems, to be mapped to a model which is understood by all FTAM implementations. The structure of this common model is discussed in the next section.

Any reference to a filestore in these guidelines, unless qualified with the word "virtual," denotes a real filestore.

2.4. File Access Data Units

A file is a named collection of information and attributes. The contents of a file can be generalized as a set of data units. Every file contains zero, one, or more data units. File data units are structured hierarchically, and thus can be represented by a tree. Figure 3 illustrates this data unit tree, which is called the FTAM file access structure. At each level in the file access structure are nodes. Each node may have an associated data unit. The nodes represent locations in a file, and the data units represent blocks or records of data. The FTAM file access structure provides a general, multi-leveled model for representing the contents of any file.

The file access structure is subdivided into trees called File Access Data Units (FADUs). Since an FADU represents a tree, an FADU may identify a larger or smaller portion of a file. For example, the root node FADU represents the entire file (i.e., all nodes and all data units in the tree) while an FADU at a deeper level may represent one line of data (e.g., one node and one data unit in the tree). File operations, such as locate and access, are performed on FADUs.

Examples can be used to further explain the concept of an FADU. Two of the many file types supported by FTAM are: unstructured and sequential. An unstructured file comprises one node (i.e., the root node) with an associated Data Unit. Figure 4 depicts the file access structure for an unstructured file. This file type contains one FADU with the Data Unit representing all the data in the file. A user can only act on this file in its entirety (e.g., send the file); specific portions of the file may not be accessed.

In contrast, the sequential file contains a root node with no associated data unit. The root node is used as a pointer into the file. Subordinate to the root node is a set of nodes, each with an associated Data Unit. The Data Units corresponding to the subordinate nodes represent a record or block (e.g., one line) of text within the file. Figure 5 depicts the file access structure for a sequential file. This file type contains multiple FADUs. The entire file is represented by one FADU. Each subordinate node is also represented by an FADU. Thus, a user can act on the file in its entirety, or can access specific records within the file.

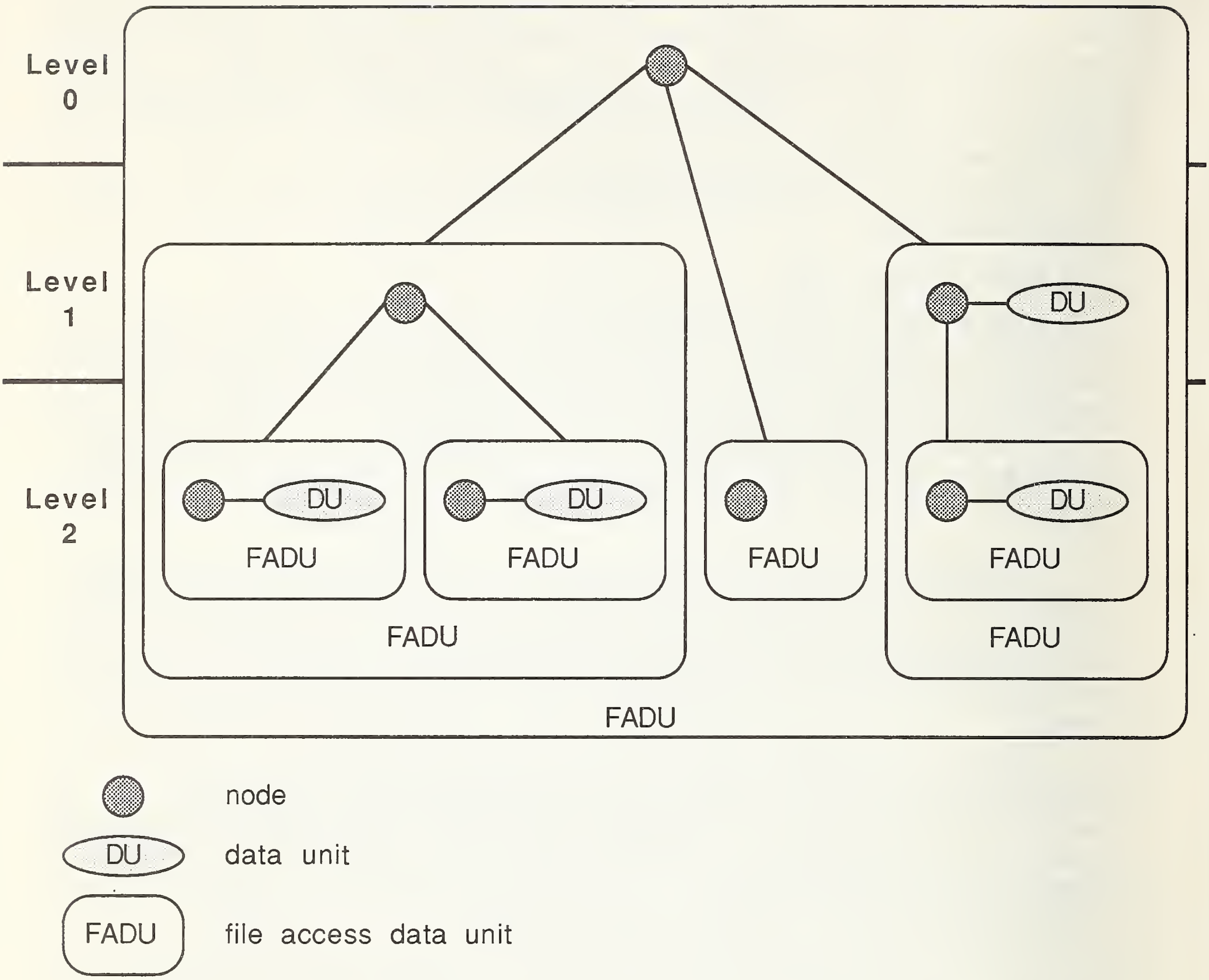


Figure 3. File Access Structure.

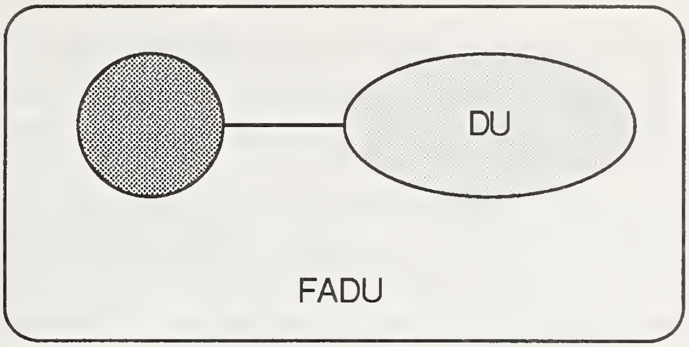


Figure 4. File Access Structure for Unstructured File.

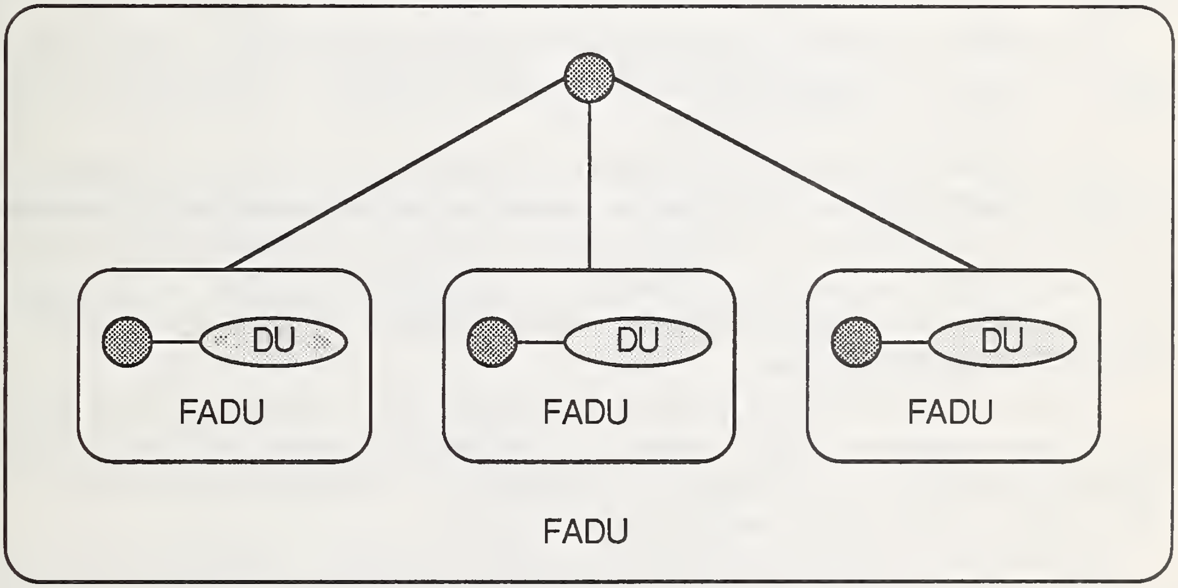


Figure 5. File Access Structure for Sequential File.

2.5. Actions

FTAM supports two types of actions: complete file actions and file access actions. Complete file actions operate on a file in its entirety, and include: selecting a file, deselecting a file, creating a file, deleting a file, opening a file, closing a file, reading file attributes and changing file attributes.

FTAM file access actions operate on FADUs. Supported actions include: locating an FADU (locate), reading an FADU (read), erasing an FADU (erase), creating a new FADU and inserting it into the file (insert), replacing the contents of an existing data unit or FADU (replace), and adding data to the end of a data unit (extend).

2.6. Document Types

The file access structure of FTAM, described in section 2.4, can represent a wide range of file structures. Since implementing the entire file access structure may introduce unnecessary or unwanted overhead for an FTAM implementation, standard subsets of the complete structure are defined in FTAM. These subsets, identified as document types, comprise the FTAM virtual filestore.

GOSIP version 1.0 supports the following document types: FTAM-1, FTAM-2, FTAM-3, NBS-6, NBS-7, NBS-8, and NBS-9. A brief description of these document types is provided below.

The FTAM-1 document type identifies an unstructured text file. A document of type FTAM-1 consists of a single character string with no delimiters. An example of an FTAM-1 file is a text file on a UNIX file system.

The FTAM-2 document type identifies a sequential text file. A document of type FTAM-2 consists of one or more character strings in a sequence separated by delimiters. An example of an FTAM-2 file is a variable length record file on a Digital VMS file system.

The FTAM-3 document type identifies an unstructured binary file. A document of type FTAM-3 consists of a single binary string with no delimiters. An example of an FTAM-3 file is a UNIX executable program.

The NBS-6 document type identifies a sequential file. A document of type NBS-6 consists of data records of various types (e.g., integers, bit strings, IA5 strings, and others) separated by delimiters. An example of an NBS-6 file is a file containing an array of records (e.g., inventory information) stored on magnetic tape.

The NBS-7 document type identifies a random access file. A document of type NBS-7 consists of a series of records, each of which can be accessed independently of the others, and the order of access is not pre-defined. An example of an NBS-7 file is a file containing an array of records (e.g., inventory information) stored on a disk.

The NBS-8 document type identifies an indexed file. A document of type NBS-8 consists of a sequence of records, such that each record is associated with a defined key which may or may not be part of the actual data record. An example of an NBS-8 file is a relational database file.

The NBS-9 document type identifies a file containing a file directory listing. A document of type NBS-9 consists of a series of file directory entries, each of which is a set of file attributes. An example of an NBS-9 file is a file containing a list of filenames, filesizes, and file creation dates; the list references files that reside in a specific file directory.

2.7. Profiles

The OIW Agreements (Phase 2) define FTAM implementation profiles for the specific functions of file transfer, file access, and file management. The implementation profiles defined in the OIW Agreements are:

Simple File Transfer (T1)

Positional File Transfer (T2)

Full File Transfer (T3)

Simple File Access (A1)

Full File Access (A2)

Management (M1)

The implementation profiles are expressed in terms of document types and FTAM actions. Definitions for the implementation profiles are presented in sections 3.3.1.1 and 3.3.1.2.

3. Evaluation Guidelines

This section details the evaluation guidelines for FTAM implementations, and contains the following sections: Section 3.1 assists users in determining their FTAM configuration. Section 3.2 provides suggestions for selecting the FTAM implementations which are procurement candidates. Section 3.3 recommends a functional evaluation procedure for the candidate FTAM implementations. Section 3.4 recommends a performance evaluation procedure for the candidate FTAM implementations. Section 3.5 recommends a procedure for rating the candidate FTAM implementations based on the functional and performance evaluations. Section 3.6 provides an example rating using the previously described guidelines, and section 3.7 describes other factors to consider when evaluating the candidate FTAM implementations.

3.1. FTAM Configurations

This section assists users in determining their FTAM configuration. The FTAM configuration is useful in evaluating the functionality of FTAM implementations because it provides input to the user's functional requirements. The configuration is also important in evaluating the performance of FTAM implementations because performance should be measured on an FTAM configuration which matches the user's configuration.

An FTAM configuration consists of two parts. The first part identifies the role of the FTAM implementation. An FTAM implementation may act in any of the following roles, or in any combination of the following roles: initiator-sender, initiator-receiver, responder-sender, or responder-receiver. An implementation will generally support both initiator roles (initiator only implementation), both responder roles (responder only implementation), or all four roles (initiator and responder implementation). For practical purposes these guidelines will describe FTAM role configurations in terms of initiator and responder roles as opposed to the four roles listed above. The second part of the FTAM configuration identifies the network type used by the implementation.

An FTAM implementation may be configured as an initiator, responder, or both. Users should examine the FTAM roles presented in this section to determine the most appropriate configuration. The following is a description of possible FTAM roles:

(1) Initiator only. (See fig. 6.) An FTAM implementation configured as an initiator can initiate FTAM requests to other FTAM implementations. For example, a user residing on an initiator only configuration can initiate a file transfer with a remote filestore. This is sometimes called an FTAM client.

(2) Responder only. (See fig. 7.) An FTAM implementation configured as a responder can respond to FTAM requests initiated by other FTAM implementations. For example, a remote user (i.e., a user not residing on the FTAM responder) can initiate a file transfer with an FTAM implementation configured as a responder only. This is sometimes called an FTAM server.

(3) Initiator and Responder. (See fig. 8.) An FTAM implementation configured as both an initiator and responder can initiate requests to remote FTAM implementations, and can respond to requests initiated by remote FTAM implementations.

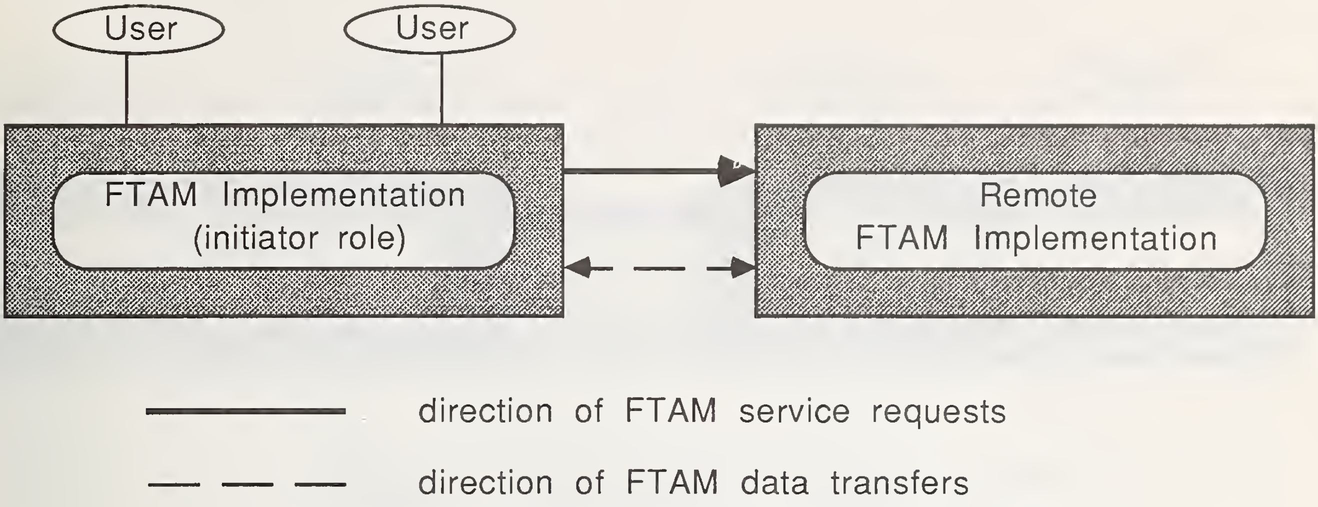


Figure 6. FTAM Role Configuration 1.

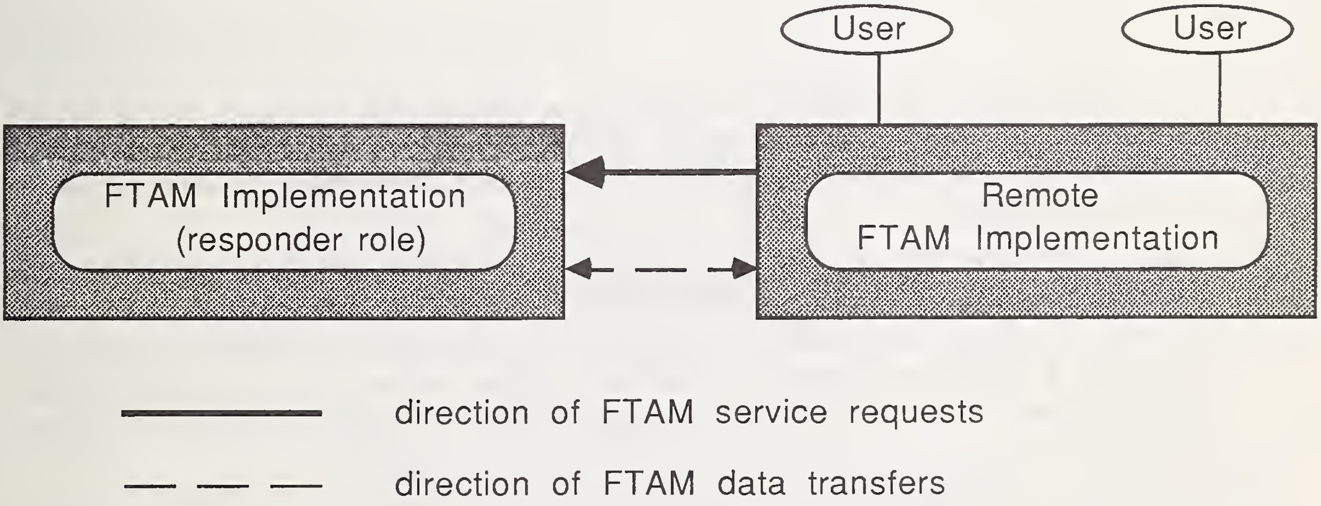


Figure 7. FTAM Role Configuration 2.

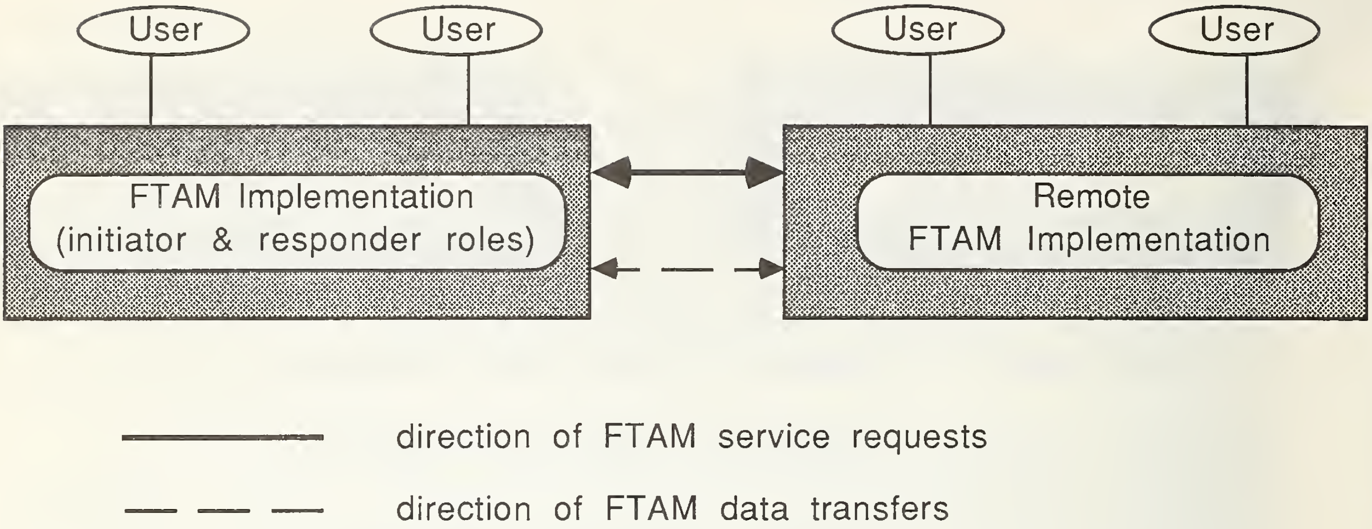

User
User
User
User
FTAM Implementation
(initiator & responder roles)
Remote
FTAM Implementation
direction of FTAM service requests
direction of FTAM data transfers

Figure 8. FTAM Role Configuration 3.

The second part of the FTAM configuration is the network type used by the FTAM implementation. An FTAM implementation may be connected to a Wide Area Network (WAN) and/or a Local Area Network (LAN). The following examples describe the WAN and LAN configurations:

(1) Wide Area Network connection. (See fig. 9.) This configuration allows an FTAM implementation connected to a WAN to communicate with other FTAM implementations. For example, a company may have a computer system with an FTAM implementation connected to a Wide Area Network. Employees of this company may transfer files with other companies having FTAM implementations that can access the same Wide Area Network.

(2) Local Area Network connection. (See fig. 10.) This configuration allows an FTAM implementation connected by a LAN to communicate with other FTAM implementations. For example, a college campus may have computer systems with FTAM implementations located in different buildings on the campus; the computer systems are interconnected by a Local Area Network. Students may transfer files between the computer systems that reside in the different campus buildings.

A user's FTAM configuration is the combination of the FTAM role and FTAM network configurations. Figure 11 provides an example FTAM configuration using the FTAM Role configuration pictured in figure 8 and the FTAM network configurations pictured in figures 9 and 10.

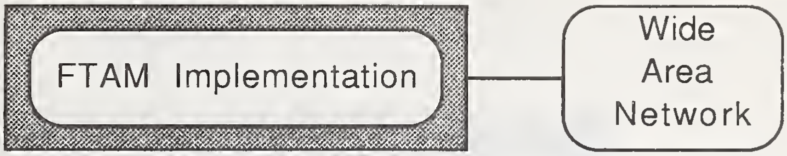


Figure 9. FTAM Network Configuration 1.

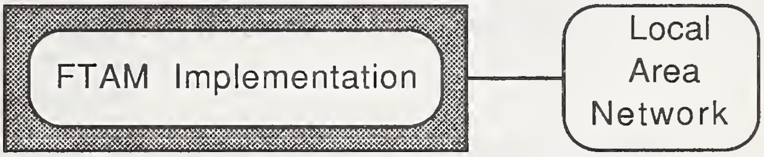


Figure 10. FTAM Network Configuration 2.

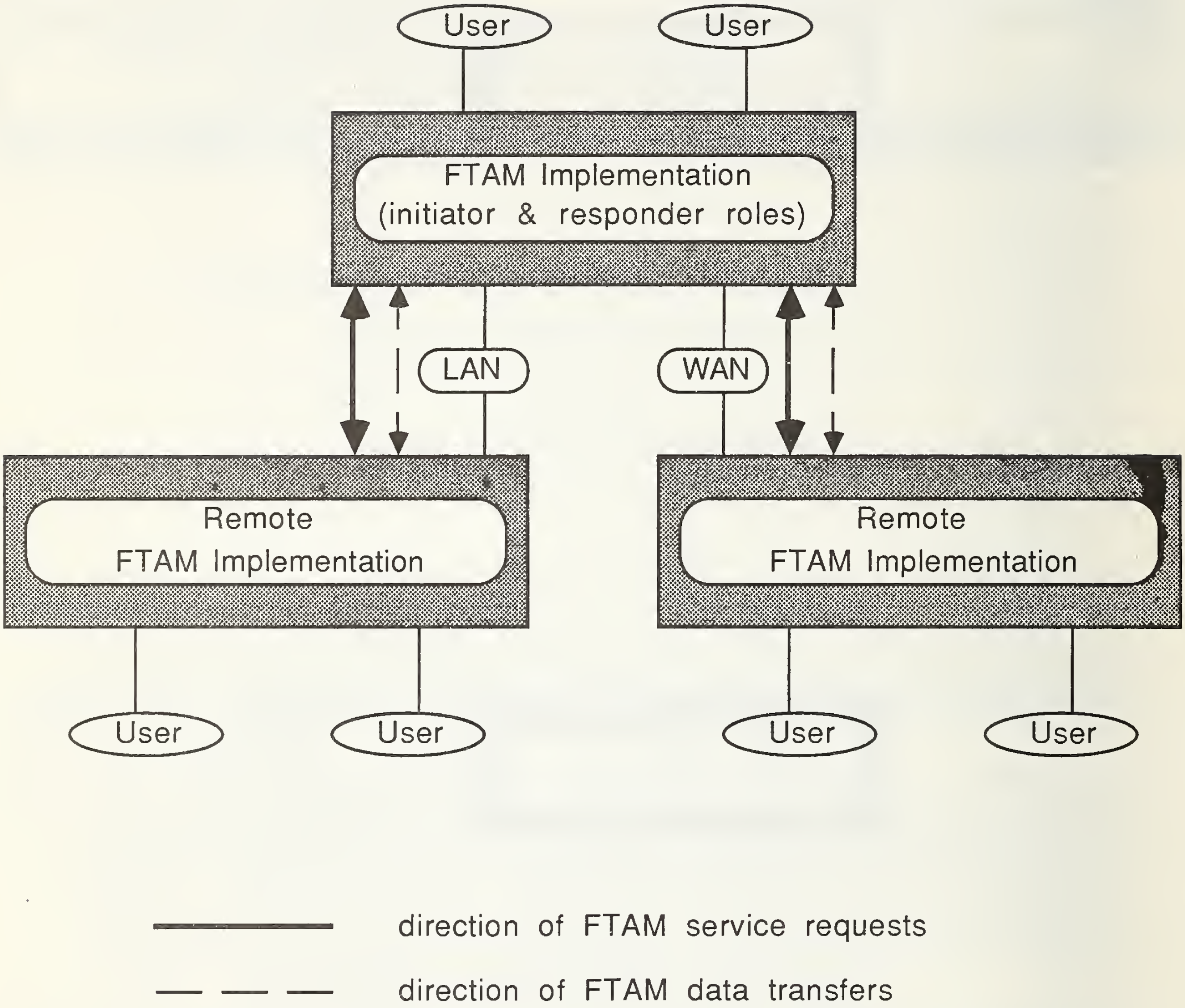


Figure 11. Example FTAM Configuration.

3.2. Candidate Implementations

This section recommends a two-step procedure for creating a list of FTAM implementations, which are procurement candidates. First, the user creates a list of available FTAM implementations. The user may find available FTAM implementations by contacting computer and communications vendors, checking Commerce Business Daily for product offerings, perusing computer and communications periodicals for product advertisements and product announcements, consulting the list of GOSIP conformance tested products, and by attending computer and communications trade shows. Second, the user defines any restrictions which apply to the candidate FTAM implementations. These restrictions may include specifying the hardware and operating system on which the candidate FTAM implementations must run, and specifying a price range for the candidate FTAM implementations. For example, if a user requires that candidate FTAM implementations run on an IBM PC compatible system, only FTAM implementations running on an IBM PC compatible system would be placed on the list of candidate implementations. After the user has determined the restrictions, the list of candidate implementations will comprise FTAM implementations which comply with all the user's restrictions. Once the list is created, product literature, users manuals, technical specifications, and other available information should be requested from each of the vendors for the candidate FTAM implementations. This information will provide input to the evaluation procedure.

3.3. Functional Evaluation Guidelines

This section recommends a procedure to evaluate the functionality of candidate FTAM implementations. It is divided into three sections. Section 3.3.1 describes functionality potentially available in FTAM implementations. Section 3.3.2 recommends a procedure for eliminating candidate FTAM implementations which do not meet mandatory functional requirements of the user. Section 3.3.3 recommends a procedure for determining which remaining candidate FTAM implementation best meets the functional requirements of the user.

3.3.1. Functional Categories

This section describes functionality potentially available in an FTAM implementation. The section is organized by grouping FTAM functions into categories of related functionality. This collection of categories provides a representative sampling of the functionality currently available in FTAM implementations. The functions comprising the categories were derived from the following sources: (1) the International Standard for File Transfer, Access and Management (ISO 8571); (2) Version 3 of the OIW Stable Implementors Agreements for Open Systems Interconnection Protocols, December 1989; and (3) practical experience with, and review of documentation from, FTAM implementations (See app. A) in the NIST Network Applications Laboratory. The minimum functional requirements for an FTAM application are defined by GOSIP. The user should carefully study each category described to become familiar with the functionality potentially available in candidate FTAM implementations. Although this list is extensive, it is not possible to include every function that is important to every user. The user may insert, in any category, any appropriate functions which are not included in that category, but are important to that user.

Certain FTAM functionality must be present in all FTAM implementations that conform to GOSIP. These mandatory functions should not be rated; they are included in this document for informative purposes only. Mandatory FTAM functions are described in section

3.3.1.1.

Many functions described in the FTAM standard are optional. For this reason functional evaluation and rating is very important. Functional evaluation is also important because the robustness of the FTAM service may limit interoperability in some cases. For example, interoperability problems may arise between FTAM implementations supporting different implementation profiles. Functions which are described within the FTAM standard, but not mandated for minimum GOSIP conformance are presented in section 3.3.1.2.

Optional FTAM functionality also includes the provision of non-standard functions. An FTAM implementation may provide a variety of functions not described in the FTAM standard and OIW Agreements. An introduction to these non-standard functional categories is presented here.

The first non-standard functional category is FTAM interfaces. An FTAM implementation may provide a user interface and a programmatic interface. The user interface, which can be window-driven or command-driven, determines an implementation's ease of use. The programmatic interface enables a user to develop an application employing the file transfer, access, and management services provided by an implementation. FTAM interfaces are described in section 3.3.1.3.

The FTAM standard defines functionality for transferring and deleting files. An implementation may, however, provide functions additional to those defined in the standard. Additional file transfer functions are described in section 3.3.1.4. Additional file deletion functions are described in section 3.3.1.5.

An FTAM implementation may allow a user to view files. When a file is viewed the content of the file is displayed to the user; however, a copy of the file is not created on the user's filestore. Functions relating to viewing files are described in section 3.3.1.6.

If an FTAM implementation supports the NBS-9 document type, a user can request a list of files that reside in a file directory. File directory functions are described in section 3.3.1.7.

An FTAM implementation may provide a user with a default configuration database and a filestore database. The default configuration database allows a user to customize aspects of FTAM associations. The filestore database allows a user to save information germane to remote filestores. FTAM databases are described in section 3.3.1.8. An FTAM implementation may also provide an on-line help facility and an operating system interface. On-line help facility functions are described in section 3.3.1.9. System interface functions, which include access to operating system commands and printing resources, are described in section 3.3.1.10.

The functions described thus far are beneficial to a user of the FTAM service. Additional FTAM functionality may be provided to assist with administrating an FTAM implementation.

Administrative functions can be divided into three categories: administration, debugging, and access control. FTAM administration functions relate to the installation, configuration, and maintenance of the implementation. They are described in section 3.3.1.11. Debugging functions are useful for isolating and resolving problems. They are described in section 3.3.1.12. Access control functions are used to limited access to the FTAM implementation. They are described in section 3.3.1.13.

Functions pertaining to the FTAM implementation describe OSI application layer functionality. Additional functionality may be present in other OSI layers. These functions are described in section 3.3.1.14.

The remaining categories in this document include testing requirements, hardware requirements, software requirements, documentation, and pragmatic constraints. Conformance and interoperability testing requirements are described in section 3.3.1.15. An FTAM implementation may require certain hardware and software for operation. Hardware requirements are described in section 3.3.1.16, and software requirements are described in section 3.3.1.17. An FTAM vendor may provide documentation detailing the use and administration of the implementation. Documentation is described in section 3.3.1.18. Finally, pragmatic constraints may be placed on an FTAM implementation. These pragmatic constraints, which include simultaneous usage constraints, filestore constraints, and filename constraints, are described in section 3.3.1.19.

3.3.1.1. Mandatory FTAM Functions

To conform to the FTAM standard and OIW Agreements (Phase 2), an FTAM implementation must support the transfer (either sending or receiving) of unstructured binary files in their entirety (i.e., FTAM-3 files). GOSIP extends this minimal service to include simple file transfer and management functions. Specifically, all GOSIP-conformant systems must support at least the simple file transfer implementation profile (T1) and the management implementation profile (M1). These profiles are described in this section. Such a system is defined in GOSIP as a limited-purpose FTAM system.

Since functionality required for GOSIP conformance is a superset of the functionality required for conformance to the OIW Agreements, some functions in this section may be labeled as optional in the Agreements. Functions mandated by GOSIP, but not mandated by the OIW Agreements, are so noted. None of the functions in this section should be rated by the user.

The *attribute groups function* allows the negotiation of the file attribute groups available during the application association. File attributes are divided into three groups: kernel, storage, and security. The kernel attribute group is the only attribute group for which support by an FTAM implementation is required. The kernel attribute group contains the filename, permitted actions, and contents type attributes. These attributes are described below. The storage and security attribute groups are optional, and are described in section 3.3.1.2.

(1) The filename attribute is a name that identifies a file. Every file in a filestore has a filename. The value for the filename is set when the file is created. A minimum filename length of eight characters must be supported.

(2) The permitted actions attribute indicates the range of actions that can be performed on a file. Actions permitted on a file include: read, insert, replace, extend, erase, read attribute, change attribute, and delete file. The read permitted action allows an FADU to be located and read. The insert permitted action allows a new FADU to be created and inserted into the file. The replace permitted action allows the contents of an existing FADU to be replaced. The extend permitted action allows data to be added to an existing FADU. The erase permitted action allows an FADU to be erased. The read attribute permitted action allows interrogation of the values of requested attributes. The change

attribute permitted action allows the changing of existing file attributes, and the delete file attribute allows a file to be deleted. For a specific file all, some, or none of the permitted actions may be present. For example, a system password file may have no associated permitted actions, and thus, cannot be accessed via FTAM. The value for the permitted actions is set when the file is created.

(3) The contents type attribute indicates the abstract data types of the contents of a file and the structuring information which is necessary if the complete file structure and semantics are to be maintained during the transfer of the file. The value for the contents type is a document type name (e.g., FTAM-3), which is set when the file is created.

The *called application title function* allows an initiator to identify the name of the responder's filestore. The called application title is commonly supplied to the initiating FTAM implementation by the initiating FTAM user.

The *configuration role function* allows an FTAM implementation to act in a specific role during the execution of an FTAM command (e.g., receiving a file). An implementation must support at least one of the following role configurations: initiator-receiver, initiator-sender, responder-receiver, responder-sender.

The *contents type list function* allows the negotiation of document types available during the application association. Support for the FTAM-3 document type is mandated by the OIW Agreements. GOSIP also requires support for the FTAM-1 document type for limited purpose systems.

Implementation profiles allow FTAM applications to support sets of FTAM functions, as defined in the OIW Agreements. GOSIP requires all FTAM implementations to support the simple file transfer profile (T1) and the management profile (M1).

The simple file transfer profile (T1) includes support of document types FTAM-1, FTAM-3, and optionally NBS-9. It also supports:

(a) reading a complete file and/or

(b) writing (replace, extend) to a file.

The management profile (M1) includes the following services: creating a file, deleting a file, reading attributes of a file, and changing attributes of a file.

The *override function* allows an initiator, when requesting to create a file, to define the action to be taken if a file with the same name already exists. GOSIP requires support for the following override values:

(a) fail the create request if the file already exists;

(b) select the file if it already exists;

(c) delete the file if it already exists and create a new file using the attributes requested by the initiator.

The *requested access function* allows an initiator to indicate the basis on which a file is being selected or recovered. Recovery is described in section 3.3.1.2. Support for the read requested access value and/or the replace requested access value is mandated in the OIW Agreements. The read requested access value is used when receiving a file. The replace requested access value is used when sending a file. GOSIP also mandates the following requested access values: read file attributes, change file attributes, and delete file.

3.3.1.2. Optional FTAM Functions

The FTAM standard and OIW Agreements contain certain functionality which an FTAM implementation is not required to support to claim minimum GOSIP compliance. These optional FTAM functions are presented in this section. The functions in this section, as well as all functions described in the remaining sections of 3.3.1, should be taken into account when rating an implementation.

As described in section 3.3.1.1, GOSIP defines a limited-purpose FTAM system. For users requiring more than the minimal functionality specified in the limited-purpose system, GOSIP defines a full-purpose FTAM system. This system is a superset of the limited-purpose system, mandating support for the positional file transfer (T2) and simple file access (A1) profiles in addition to the simple file transfer (T1) and Management (M1) profiles required by the limited-purpose system. A full-purpose FTAM system is able to interoperate with a limited-purpose FTAM system at the intersection of their capabilities. Some functions presented in this section, although defined as optional in the FTAM standard and OIW Agreements, must be supported if an implementation is to conform to the GOSIP-compliant full-purpose system. Functions required for the full-purpose FTAM system are so noted.

The *access passwords function* allows passwords to be associated with the actions specified in the requested access function. This function is available only if the security file attribute group has been negotiated.

The *account function* allows the initiator to identify the account to which costs incurred are to be charged.

The *attribute groups function* allows the negotiation of attribute groups available during the application association. File attributes are divided into three groups: kernel (covered earlier in section 3.3.1.1), storage, and security. Support of the storage and security attribute groups by an FTAM implementation is optional. The attributes comprising the storage group are described below. Unless otherwise noted, support of an attribute within the storage group is also optional.

(1) The file availability attribute indicates whether delay should be expected before the file can be opened. The attribute value is set at file creation. The value may be either "immediate availability" or "deferred availability." If the storage file attribute group is supported, the file availability attribute must also be supported.

(2) The filesize attribute indicates the size of the file. It is altered by the responder whenever the file has been opened for modification or extension and is closed. The attribute value is set to the nominal size in octets of the complete file when the file is closed. If the storage file attribute group is supported, the filesize attribute must also be supported.

(3) The date and time of creation attribute indicates when the file was created. It is set by the responder when the file is created and refers to the local date and time of the responder.

(4) The date and time of last modification attribute indicates when the contents of the file were last modified. It is altered by the responder whenever the file has been opened for

modification or extension and is closed. The date and time attribute value becomes the date and time when the close action was performed.

(5) The date and time of last read access attribute indicates when the contents of the file were last read. It is altered by the responder whenever the file has been opened for reading and is closed. The date and time attribute value becomes the date and time when the close action was performed.

(6) The date and time of last attribute modification attribute indicates when the value of a file attribute was last modified. It is altered by the responder whenever the change attribute action is successfully performed on one or more attributes. The date and time attribute value becomes the date and time when the change attribute action was performed.

(7) The identity of creator attribute identifies the creator of the file. It is set by the responder to identify the current initiator when the file is created.

(8) The identity of last modifier attribute identifies the last modifier of the file. It is altered by the responder whenever the file has been opened for modification or extension and is closed. The identity is set when the file is closed to the current initiator.

(9) The identity of last reader attribute identifies the last reader of the file. It is altered by the responder whenever the file has been opened for reading and is closed. The identity is set when the file is closed to the current initiator.

(10) The identity of last attribute modifier attribute identifies the last modifier of a file attribute. It is altered by the responder whenever the change attribute action is successfully performed on one or more attributes. The attribute is set to identity the current initiator.

(11) The future filesize attribute indicates the nominal size in octets to which the file may grow as a result of modification and extension.

(12) The storage account attribute identifies the accountable authority responsible for accumulated file storage charges. The attribute value is set at file creation.

In addition to the storage attribute group, support of the security file attribute group is optional by an FTAM implementation. The security group contains the access control and legal qualifications attributes. If the security file attribute group is supported, the access control attribute must also be supported; the legal qualifications attribute is optional.

(1) The access control attribute defines conditions under which access to the file is valid. These conditions include: read, insert, replace, extend, erase, read attribute, change attribute, and delete file. Access to the file is allowed if at least one of these conditions is satisfied. The attribute value is set at file creation.

(2) The legal qualifications attribute conveys information about the legal status of the file and its use. The attribute value is set at file creation.

The *charging function* conveys information on the costs attributed to the account during a regime.

The *concurrency control function* indicates access available to the user and access available to other users. Values for concurrency control are set on a per requested access basis (see requested access function in this section), and are as follows:

(a) shared (i.e., the user may perform the operation, so may others);

(b) exclusive (i.e., the user may perform the operation, others may not);

(c) not required (i.e., the user will not perform the operation, others may);

(d) no access (i.e., no one may perform the operation).

The *configuration role function* allows an FTAM implementation to act in a specific role during the execution of an FTAM command (e.g., receiving a file). An FTAM implementation may support any combination of the following role configurations: initiator-receiver, initiator-sender, responder-receiver, responder-sender.

The *contents type list function* allows the negotiation of document types available during the application association. In addition to the document type support required for a limited-purpose system, support for the FTAM-2 document type is required for a GOSIP full-purpose system. Support for the following document types is optional: NBS-6, NBS-7, NBS-8, and NBS-9.

The *create password function* allows an initiator to establish the permission required, if any, to create files in the responder's filestore.

The *diagnostics function* allows a responder to convey detailed information on the failure of a requested action. The information is typically displayed for the initiating user.

The *filestore password function* conveys a password which is used to authenticate the initiator to the responder. Typically, the filestore password is provided by the initiating user in conjunction with the initiator identity to login to the responder's filestore.

Implementation profiles allow FTAM applications to support sets of FTAM functions, which are defined in the OIW Agreements. Support for the positional file transfer (T2) and simple file access (A1) profiles is required for the GOSIP full-purpose system. Support for the full file transfer (T3) and full file access (A2) profiles is optional.

The positional file transfer (T2) profile is a superset of the simple file transfer profile (T1) and includes support of document types FTAM-1, FTAM-2, FTAM-3, and optionally NBS-6, NBS-7, and NBS-9. It also supports:

(a) reading a complete file or a single FADU which is identified by position and/or;

(b) writing (replace, extend, insert) to a file or an FADU.

The simple file access (A1) profile includes support of document types FTAM-1, FTAM-2, FTAM-3, and optionally NBS-6 and NBS-7. It also supports:

(a) reading a complete file or a single FADU which is identified by position and/or;

(b) writing (replace, extend, insert) to a file or an FADU;

(c) locating and erasing within files.

The full file transfer profile (T3) includes support of document types FTAM-1, FTAM-2, FTAM-3, NBS-6, NBS-7, NBS-8, and optionally NBS-9. It also supports:

(a) reading a complete file or a single FADU which is identified by position or a key and/or;

(b) writing (replace, extend, insert) to a file or an FADU.

The full file access (A2) profile includes support of document types FTAM-1, FTAM-2, FTAM-3, NBS-6, NBS-7, and NBS-8. It also supports:

(a) reading a complete file or from a series of FADUs which are identified by position or by key;

(b) writing (replace, extend, insert) to a file or an FADU;

(c) locating and erasing within files.

The *initiator identity function* identifies the calling user. Typically, the initiator identity is provided by the initiating user in conjunction with the filestore password to login to the responder's filestore.

The *override function* allows an initiator, when requesting to create a file, to define the action to be taken if a file with the same name already exists. Support for the following override value by an FTAM implementation is optional: delete the file if it already exists and create a new file using the old file's attributes (i.e., delete the contents of the existing file and select it).

The *recovery function* allows the initiator to recover from a failure after a file has been opened. Recovery can begin before the data transfer, at some negotiated point within the data transfer, or after the data transfer has completed. The recovery may be immediate or deferred on the existing, or on a different, association.

The *requested access function* allows an initiator to indicate the basis on which a file is being selected or recovered. Support of the insert and erase requested access values is optional.

The *restart data transfer function* allows a data transfer to be interrupted and restarted immediately at some negotiation point within the current transfer.

3.3.1.3. FTAM Interfaces

This category begins the description of non-standard FTAM functionality which may be provided by an implementation. The first non-standard functional category to be described is FTAM interfaces. Two interfaces may be provided by an FTAM implementation: a user interface and an application program interface (API).

The FTAM user interface determines the implementation's ease of use. A user interface may be implemented as a program or as an extension to the operating system. If implemented as a program, the user must start the program to access FTAM commands. The term "FTAM command" is used in these guidelines to denote a user-requested FTAM service, such as transferring a file or a part of a file, reading or changing file attributes, or deleting a file. If the FTAM user interface is implemented as an extension to the operating system, specific operating system commands are enhanced to provide FTAM functionality. For example, if "list" is an operating system command that displays file attributes on the local file system, "list/ftam" may be an FTAM command that displays file attributes on a remote filestore.

If implemented as a program, the FTAM user interface may allow a user to execute multiple FTAM commands during one FTAM association. For example, a user may open an association (i.e., login) with a remote FTAM implementation by providing the correct initiator identity and filestore password values. Once the association is established the user can perform various FTAM commands, such as sending and receiving files, before terminating the association. If the FTAM association is not maintained so that the user can execute multiple FTAM commands, the user must provide the initiator identity and filestore password values (i.e., must login) prior to executing each FTAM command.

When a user provides a password (e.g., a responder's filestore password), the value may be concealed (i.e., not displayed on the screen) as a security feature. Nothing may be displayed on the screen while the user enters the password; or some character, such as an asterisk, may be displayed for each character entered by the user.

The FTAM user interface may be window-driven or command-driven. With a window-driven interface, the user may divide the screen into multiple sections, each section responding to input from the keyboard or a mouse. With a command-driven interface, menus may be employed to prompt the user for information. A user may opt not to use menus if they are provided. Although menus may be beneficial to someone unfamiliar with the implementation, it may be faster for a user executing one FTAM command to enter the information at a command line prompt. To enhance speed of use, an implementation may also allow a user to define macros for frequently used FTAM commands.

The FTAM user interface may allow a user to execute FTAM commands in a batch or background mode. If an FTAM command is submitted as a batch command, the user can perform other tasks on the system while the FTAM command executes. Functions relating to the batch command, such as requesting the status of the command, cancelling the command, or resubmitting the command if the command fails, may also be available.

The user interface provided by an FTAM implementation may emulate the user interface provided by File Transfer Protocol (FTP) implementations. FTP is the file transfer service developed by the Department of Defense. The FTP login procedure and relevant FTP command names may be incorporated into the FTAM user interface.

In addition to the FTAM user interface, a user may be provided with an application program interface (API) to the FTAM implementation. An FTAM API enables a user to develop an application employing the file transfer, access, or management services provided by the implementation. The programmatic interface may contain a high level interface and/or a low level interface. The high level interface provides access to high level FTAM functions, such as transferring a file or deleting a file. The low level interface provides access to low level FTAM functions, such as establishing an FTAM association, selecting a file, opening a file, and reading and writing data.

The API provided by the FTAM implementation may in the future be POSIX conformant. POSIX is a Portable Operating System Interface for Computer Environments standard sponsored by the Technical Committee on Operating Systems of the IEEE Computer Society. An application running in a POSIX environment can be ported to other POSIX systems with minimal changes.

3.3.1.4. Transferring Files

The FTAM standard defines functionality for transferring a file to and from a remote filestore; however, an implementation may provide additional functions. An implementation may allow a user to transfer multiple files within a file directory with one command. A user may transfer the files according to various criteria, such as all files created after a specific date and time or all files created by a specific user. If multiple files are to be transferred, an implementation may display the name of each file prior to its transfer, and request that the user confirm the transfer of the file.

Another method by which a user may transfer multiple files is by specifying a filename that contains a wildcard character. Wildcard characters, such as the asterisk, will allow the filename specified by the user to match multiple filenames on a filestore. For example, the filename "CON*.DAT" could match the filenames "CONCERT.DAT" and "CONAN.DAT." The user may be given the option of allowing or disallowing wildcards when specifying filenames.

Receiving multiple files requires that both implementations involved in the file transfers support the NBS-9 (i.e., file directory listing) file type. Since the FTAM standard currently makes no provision for the transfer of multiple files, the user's FTAM implementation must perform a file directory listing to obtain the names and attributes of files in a directory on the responder's filestore. Once the filenames and file attributes are known, the user's FTAM implementation can transfer the files individually. For example, if a user requests to receive all files in a directory on a remote filestore that were created after July 4, 1991, the user's FTAM implementation first performs an NBS-9 file transfer to obtain the names and creation dates of all the files in the directory on the remote filestore. The user's FTAM implementation then parses the creation dates to determine which files were created after July 4, 1991, and transfers these files individually. Support of the NBS-9 file type is not required for sending multiple files because the interaction between an FTAM implementation and its local filestore is determined by local implementation decisions.

An implementation may allow a user to move a file to or from a responder's filestore. The move function is accomplished by transferring the file to or from the responder's filestore, then deleting the original file.

If a file exists with the same name as a file being transferred, functions additional to those defined in the standard may be available to the user. For example, a user may append a file being transferred to the existing file. If the NBS-9 file type is supported by both implementations involved in the transfer, the user may request to transfer a file only if the file to be transferred was created or modified more recently than the existing file.

To assist with transferring files, a user may obtain a listing of all remote filestores that have been configured into the user's FTAM implementation. A user may also transfer a file to or from a different directory on the local filestore. If a file is transferred to or from the local filestore, no data is transferred over a network. Transferring a file locally is a meaningful function only if the FTAM user interface is implemented as a program.

When transferring a file a user may request to simplify the document type. For example, a user may request to receive an FTAM-2 (i.e., sequential text) file as an FTAM-1 (i.e., unstructured text) file. Not all document type simplifications are allowed; the FTAM standard defines rules for simplifying document types.

A user may transfer a file between two remote filestores. For example, a user of FTAM implementation "A" may transfer a file from filestore "B" to filestore "C." This function is accomplished in three steps. The first step comprises transferring the file from filestore "B" to the local filestore (i.e., FTAM implementation A's filestore). The second step comprises transferring the file from the local filestore to filestore "C." Deleting the file from the local filestore is the final step. When transferring a file between two remote filestores, an implementation may allow a user to rename the file. This feature is a means to provide a three-party file transfer with a two-party protocol.

An FTAM implementation may display status information when transferring a file. This information could include filenames, number of files transferred, bytes of data transferred, amount of time required for the transfer, and the transfer rate. FTAM specific information such as the initiator identity value and the responder's filestore name may also be displayed. The information may be presented to the user upon completion of the FTAM command, or may be displayed during the command and updated as the command progresses. A user may be allowed to limit the amount of information displayed (e.g., have only the names of the files being transferred displayed), or have the status information written to a file on the local filestore.

3.3.1.5. Deleting Files

The FTAM standard defines functionality for deleting a file; however, an implementation may provide additional functions. An implementation may allow a user to delete multiple files within a file directory with one command. A user may delete the files according to various criteria, such as all files created after a specific date and time or all files created by a specific user. If multiple files are to be deleted, an implementation may display the name of each file prior to its deletion, and request that the user confirm the deleting of the file.

Deleting multiple files requires that both implementations involved in the deletion support the NBS-9 (i.e., file directory listing) file type. Since the FTAM standard currently makes no provision for the deleting of multiple files, the user's FTAM implementation must perform a file directory listing to obtain the names and attributes of files in a directory on the responder's filestore. Once the filenames and file attributes are known, the user's FTAM implementation can delete the files individually.

A user may delete files that reside on the local filestore. If a local file is deleted, no data is transferred over a network. Deleting a local file is a meaningful function only if the FTAM user interface is implemented as a program.

An FTAM implementation may display status information when deleting files, such as the names of the files deleted and the number of files deleted. The information may be presented to the user upon completion of the FTAM command, or may be displayed during the command and updated as the command progresses. The user may have the status information written to a file on the local filestore.

3.3.1.6. Viewing Files

An FTAM implementation may allow a user to view files on a responder's filestore. When a file is viewed the contents of the file are transferred to the user's FTAM implementation, but instead of being written to a file on the local filestore, are displayed on the screen. A user may be provided with commands for viewing the file, such as scrolling and paging forward and backward.

A user may view files that reside on the local filestore. If a local file is viewed, no data is transferred over a network. Viewing a local file is a meaningful function only if the FTAM user interface is implemented as a program.

3.3.1.7. File Directory Operations

If an FTAM association is established between two FTAM implementations that support the NBS-9 file type, a user can request a listing of the files that reside in a directory on the remote filestore. The manner in which this information is presented to the user can vary. A user may request a long or short listing of the information. A long listing may provide all the file attributes for each file while a short listing may provide only the filename, filesize, and date and time of creation. The user may request that only files matching specific criteria be listed, such as all files created by a specific user. With each file directory listing an implementation may provide summary information, such as the number of files listed. An implementation may also provide formatting control for the information displayed. For example, a user may specify that the filename, filesize, and creation date are to be displayed on one line, and that at most 16 characters are to be displayed for the filename, 6 characters for the filesize, and 10 characters for the creation date. Support of the NBS-9 file type will be a desirable function until filestore management becomes part of the FTAM service.

A user may view attributes of files that reside on the local filestore. If local file attributes are viewed, no data are transferred over a network. Viewing the attributes of a local file is a meaningful function only if the FTAM user interface is implemented as a program.

An FTAM implementation may allow a user to change the current working file directory on a responder's filestore. Although Implementation Agreements have yet to be written for filestore management functions, this function may be provided by proprietary means.

3.3.1.8. Default Databases

An FTAM implementation may provide a user with a default configuration database for customizing all FTAM associations. Information contained in this database may include a default setting for the override parameter (e.g., fail creating a file if a file with the same name exists), the file attributes displayed during a read file attributes FTAM command (e.g., filename, filesize, and date and time of creation), the amount of statistics displayed (e.g., display the filename and filesize when transferring files), and whether wildcards may be used when specifying filenames.

An FTAM implementation may also provide a user with a filestore database. Information contained in this database may include filestore names with associated initiator identity and filestore password values. If a user does not specify an initiator identity or filestore password when logging into a responder's filestore, the values in this database are used. A user may display and/or modify the contents of the databases.

3.3.1.9. On-Line Help Facilities

An FTAM implementation may provide a user with an on-line help facility. Using the help facility, a user can obtain information about the implementation, such as available FTAM commands, how the commands are used, and the parameters and options supported by the commands.

3.3.1.10. System Interface

If the FTAM user interface is implemented as a program, a user may be allowed to stop the program temporarily and access operating system commands. This function is useful, for example, if a user has opened an association with a remote FTAM implementation, but wants to edit a local file before sending it. A user may also be given access to printing resources from within the user interface program. Information such as the content of files, file attributes, and status and debugging information may be printed by the user.

3.3.1.11. Administrative Functions

FTAM administrative tasks begin with the installation of the FTAM implementation. An implementation may provide an on-line training program for installing and configuring the implementation. The installation may be accomplished via an automated procedure which prompts for information. To ensure the installation was performed correctly, an installation verification facility may be provided.

Once installation is complete, the implementation must be started. The starting of the implementation may occur when the system supporting the implementation is started, or it may be started and stopped manually. Also, the implementation may be started as just an initiator or a responder. If the implementation is operating in both initiator and responder roles, one of the roles may be stopped without affecting the other role. For example, an implementation may be stopped from initiating FTAM commands, but can continue to respond to FTAM commands initiated by other FTAM implementations.

Once the FTAM implementation is installed and operating, the two main administrative tasks are maintaining FTAM administration databases and optimizing the implementation. FTAM administration databases are typically used to store information pertaining to remote filestores. This information includes filestore names and underlying OSI layer addressing.

To assist with optimizing the implementation, FTAM statistics relating to implementation usage may be provided. Statistics such as the date and time the implementation was started, total number of files or bytes sent and received, average throughput, current user count, and current FTAM associations may be available. The statistics may be separated to show usage of the implementation in both initiator and responder roles. If the statistics show the FTAM implementation to be overloaded, specific FTAM parameters may be modified to optimize the implementation. These parameters may include the maximum number of simultaneous FTAM commands that can be initiated by local users, the maximum number of simultaneous FTAM commands to which the implementation can respond, the maximum number of simultaneous FTAM commands that the implementation can process as initiator and responder, and the maximum number of simultaneous users.

An FTAM implementation may provide a utility program to assist with performing administrative tasks. The utility program may be used to view and update FTAM database information and optimize FTAM parameters. The utility program may also contain an on-line help facility so that information such as commands recognized by the program, how they are used, and the options and parameters supported by the commands may be viewed.

An FTAM implementation may provide other administrative functions. A user may be able to backup the FTAM implementation, or restore the implementation from a backup. The backup may be restored to a different machine if the original machine is encountering hardware problems. Restoring an implementation is different from installing an implementation in that information registered in FTAM databases need not be re-entered when the implementation is restored.

3.3.1.12. Debug Capabilities

An FTAM implementation may provide functions which assist in resolving problems encountered. An implementation may provide a log file for FTAM activity (e.g., file transfers). Any errors encountered, such as those involving system resources, may also be logged. This information may be displayed on the system console in addition to being written to a file.

To aid in solving specific problems an implementation may provide a utility that monitors and displays all FTAM protocol and data exchanges. For example, a user may be able to view the parameters and values transferred between two FTAM implementations during the establishment of the FTAM regimes. This utility may also perform tracing of the underlying OSI layers.

3.3.1.13. Access Control

An FTAM implementation, acting as a responder, may provide functions which limit the access of initiating FTAM users. Access control may be based on the initiating user's network address or initiator identity value. An FTAM implementation may deny access to specific network addresses or only allow access by specific network addresses. Similarly, access by certain initiator identity values may be denied or limited. An example of limited access is an FTAM responder allowing a specific user to receive files from its filestore, but not allowing that user to send files to its filestore.

An FTAM responder may provide an account that may be used by any initiating user. This FTAM account, sometimes referred to as an anonymous account, is accessed by providing the correct initiator identity value (e.g., ANONYMOUS or GUEST) and any filestore password value. The password is not validated. Since any user can access this account, the account is typically given minimal system privileges as a security precaution. For example, a user establishing an FTAM association using the anonymous FTAM account may only be allowed to transfer files to and from a specific directory on the responder's filestore.

3.3.1.14. Underlying OSI Layers

In addition to FTAM functions, an FTAM implementation may provide additional functionality in the underlying OSI layers. Presentation Layer functionality determines the Presentation Layer implementations with which the FTAM implementation can interoperate. The OIW Agreements specify that only the Kernel Presentation functional unit must be supported. FTAM places definite additional requirements on the functionality of the Presentation and Session layers. Most should not vary with the implementation. Session options are listed in 5.13.1.4 in the OIW Agreements.

For GOSIP end systems, the connection-oriented transport service provided by Transport Class 4 is mandated for government-wide interoperability. It is the required means for providing a reliable end-to-end communications path between end systems.

The Connectionless Network Service (CLNS) is also mandated for government-wide interoperability. Together with Transport Class 4, it provides the means for interconnecting local and wide area subnetworks. The Connection-Oriented Network Service (CONS) is an additional, optional service that may be specified in conjunction with Transport Class 4 for communication among end systems that are directly connected to X.25 wide area networks and Integrated Services Digital Networks (ISDNs). Use of this service can, under certain circumstances, avoid the overhead associated with the CLNS. In addition, Transport Class 0 over the CONS may be required to communicate with systems prevalent outside North America, that are not compliant with GOSIP.

If the FTAM implementation is to be used over an X.25 network, the X.25 software provided with the implementation should conform to the 1984 version of the Consultative Committee on International Telephony and Telegraphy (CCITT) X.25 protocol. It may instead conform to the 1980 X.25 where 1984 services are not available. To assist with setting X.25 parameter values, pre-defined groups of X.25 parameters specific to a network service may be included with the implementation. For example, if the FTAM implementation will be used to transfer files over ACCUNET, the Wide Area Network provided by AT&T, the ACCUNET parameter group would provide all X.25 parameter values needed to establish X.25 ACCUNET connections. The provision of these values decreases installation time and optimizes X.25 performance. X.25 parameters may be modified later to better suit any individual requirements.

Whether FTAM operations occur over a WAN or a LAN, certain statistical information, such as the number of bytes sent and received, the number of packets sent and received, and various error counters may be available. In addition, facilities to optimize performance, troubleshoot interworking problems, and log information within each of the remaining underlying OSI Layers may be provided. A separate log file may exist for each layer.

3.3.1.15. Conformance and Interoperability Testing and Registration

Conformance testing, which verifies that an implementation conforms to the standard, is required by NIST in order for suppliers to claim legitimately to be GOSIP compliant. Interoperability testing verifies that the implementation interoperates with other implementations. Interoperation testing with other implementations is optional, but strongly recommended.

NIST has defined a GOSIP Testing Program to permit Federal agencies to substantiate claims of GOSIP compliance. If a supplier claims GOSIP compliance or conformance for a product, then a buying agency is advised to require the product be tested in accordance with the criteria specified in the GOSIP Conformance and Interoperation Testing and Registration Report. If a product includes a multi-layered GOSIP profile, then all protocols for which GOSIP compliance or conformance is claimed should be tested in accordance with these criteria. Federal agencies requiring claims of GOSIP conformance should consult the Register of Conformance Tested GOSIP products. Agencies that require that interoperability between GOSIP conformant products be documented should consult the data supplied by a service on the register of Interoperability Test and Registration Services. Information such as which tests were performed, with whom, when, as well as the actual test results should be made available to the user. For more information on GOSIP testing, the user should read the "GOSIP Conformance and Interoperation Testing and Registration" document (NISTIR 4594).

An FTAM implementation that conforms to the OIW Agreements may not be GOSIP conformant. This is because the functionality required by GOSIP is a superset of the functionality mandated by the OIW Agreements. Under these circumstances, federal agencies should require contractually that GOSIP functionality be included in the FTAM product by a certain date.

3.3.1.16. Hardware Requirements

Different FTAM implementations may require specific hardware for operation. Requirements pertaining to the CPU, disk space, memory, and external devices (e.g., X.25 interface cards) may need to be met.

3.3.1.17. Software Requirements

Different FTAM implementations may require specific software configurations for operation. For example, an FTAM implementation may consist of multiple software components which need to be installed separately. OSI software not residing in the Application Layer, such as lower layer software, may require installation. In addition, an FTAM implementation may need a specific operating system and version.

3.3.1.18. Documentation

An FTAM implementation may provide a user with a variety of manuals explaining the interworkings of the application. Although each implementation may organize its documentation differently, the following information, regardless of format, may prove useful to the user: installation guide, user's guide, administration guide, troubleshooting guide, and quick reference guide. An installation guide provides information for installing and configuring the FTAM implementation. It may contain sample installations and lists of files which are created on the local file system. A user's guide describes the FTAM commands recognized by the implementation. An administration guide details management and maintenance of the FTAM implementation. A troubleshooting guide describes possible errors and how they are corrected. A quick reference guide, which is useful once the user is familiar with the implementation, provides a quick reference for FTAM commands.

3.3.1.19. Pragmatic Constraints

FTAM pragmatic constraints pertain to the following topics: simultaneous FTAM associations, filestores, and filenames. In a multi-tasking environment, an FTAM implementation may limit the number of simultaneous FTAM associations that may be established. This limitation may occur for the implementation acting in the role of initiator and responder separately. The implementation may also limit the number of users that can use the implementation simultaneously.

The second pragmatic constraint pertains to the number of filestores that can be supported by the FTAM implementation. Most FTAM implementations support one filestore, which meets the requirements of most users.

The final pragmatic constraint pertains to filenames. A filestore may limit the length of a filename. Receiving a file with a name longer than this limit may result in the name of the file on the filestore being truncated. In addition to length, a filestore may limit the characters that can be used in a filename. For example, characters such as quotes or multiple periods may not be allowed in a filename. Also, a filestore may be sensitive to the case of the characters comprising the filename.

3.3.2. Mandatory Functional Requirements

This section recommends a procedure for eliminating candidate FTAM implementations which do not meet the mandatory functional requirements of the user. The user should create a list of any functions which must be included in a candidate FTAM implementation, for that implementation to be acceptable. The user should be careful not to list as mandatory any functions which are instead highly desirable, because this can unnecessarily restrict the list of candidate FTAM implementations. This list may be created by reviewing the functions in each of the functional categories, noting which functions are mandatory for the user. Once the list is created, the user should verify that each of the candidate FTAM implementations contains the mandatory functions. A candidate FTAM implementation which does not contain all of the mandatory functions should be removed from the list of implementations at this point. As an example, if after reviewing the functional categories, the user decides that the candidate FTAM implementations must support connections to both an X.25 Wide Area Network and an 802.3 Local Area Network, then FTAM implementations which do not support connections to both X.25 and 802.3 are unacceptable to the user, and should no longer be evaluated.

3.3.3. Functional Evaluation

This section recommends a procedure for determining which of the candidate FTAM implementations, possessing all required functions, best meets the functional capabilities desired by the user. First, the user must assign weights to each category of functions and to each function within that category based on how important the category of functions and each function within that category is to the user. This procedure is defined in section 3.3.3.1. Second, the user must rate each of the candidate FTAM implementations, based on the weights assigned by the user and what functionality is available in the candidate FTAM implementations. This procedure is defined in section 3.3.3.2.

3.3.3.1. Weighing Functions

This section provides guidelines for assigning weights to each category of functions and to each function within a category based on how important the category of functions and each function within the category is to the user. First, the user should select one of the three suggested options for weighing the functions within each category. The options are:

(1) Determine a weight for each individual function in a category based on how important that function is to the user. As an example, let us assume that a category contains 20 functions and that the user has decided that functions 1-5 are very important, functions 6-10 are moderately important, functions 11-15 are slightly important, and functions 16-20 are not of any importance. Then the user may assign a weight of 3 for functions 1-5, a weight of 2 for functions 6-10, a weight of 1 for functions 11-15, and a weight of 0 for functions 16-20.

(2) Assign each function in a category, which is of interest to the user, a weight of 1, and all other functions in the category a weight of 0. This option assumes that the user is either interested or not interested in a function, and that each of the functions of interest to the user are of equal importance. As an example, let us assume that a category contains 10 functions and that the user is interested only in functions 1-5. Then the user would assign a weight of 1 for functions 1-5, and a weight of 0 for functions 6-10.

(3) Do not assign any weight to the functions in the category.

The user must decide which option to use for evaluating each category, based on the tradeoffs of the three options. An evaluation of a category based on option 1 is the most precise and the most time-consuming of the three options. An evaluation of a category based on option 3 is the least precise and the least time-consuming of the three options. The user should note that since the functions within one category are rated independently of the functions within the other categories, different rating options may be selected for different categories, based on the user's preferences.

Second, the user should balance each of the categories, and determine how important each category is to the user. This step results in the assignment of a weight to each category.

The categories must be balanced because the maximum score which may be received by one category has no relation to the maximum score which may be received by the other categories. For example, one category may be scored as the sum of the number of functions available in the implementation. If there are 10 functions in the category, then the category can receive a maximum score of 10. Another category may be subjectively scored as 1 if the implementation acceptably performs the functions in this category; otherwise it will receive a score of 0. If these two example categories are of equal importance to the user, then the weight of the second category must be 10 times as large as the weight of the first category, in order to balance the categories.

The user must determine relative levels of importance of the categories. The user first assigns a maximum score to each category to reflect its importance to the user. For example, the user may decide that a category which is extremely important to the user can receive a maximum score of 400 points, a category which is very important to the user can receive a maximum score of 300 points, a category which is important to the user can receive a maximum score of 200 points, a category which is less important to the user can receive a maximum score of 100 points, and a category which is not of any importance to the user can only receive a score of 0 points.

The user must then compute a weight for each category. The weight for each category is calculated by dividing the maximum score the user has determined that the category can receive by the maximum score of the functions within the category. As an example, the user may consider the categories of transferring files and deleting files to be extremely important (400 points), the category of FTAM interfaces to be fairly important (300 points), the category of file directory operations to be important (200 points), the category of viewing files to be less important (100 points), the category of default databases to be of no importance (0 points). If the maximum score of the functions in the transferring files category is 10, then it is assigned a weight of 400/10 which equals 40. If the maximum score of the functions in the deleting files category is 20, then it is assigned a weight of 400/20 which equals 20. If the maximum score of the functions in the FTAM interfaces category is 10, then it is assigned a weight of 300/10 which equals 30. If the maximum score of the functions in the file directory operations category is 5, then it is assigned a weight of 200/5 which equals 40. If the maximum score of the functions in the viewing files category is 1, then it is assigned a weight of 100/1 which equals 100. If the maximum score of the functions in the default databases profile category is 10, then it is assigned a weight of 0/10 which equals 0.

3.3.3.2. Functional Evaluation Rating

This section recommends a procedure to rate the candidate FTAM implementations functionally, based on the weights assigned by the user and what functionality is available in the candidate FTAM implementations. In order to determine which of the functions in a category are available in a candidate FTAM implementation, information such as product literature, user's manuals, and technical specifications, should be obtained from the vendors. This functional rating procedure must be performed on all of the categories described in this document, and the procedure must be repeated for each candidate FTAM implementation. The procedure for determining a functional rating for each candidate FTAM implementation follows.

First, the user must score each category of a candidate FTAM implementation. The procedure for scoring a category varies, depending on which of the three previously described weighing options is chosen.

If option 1 is chosen, then the score of the category is the sum of the weight of each function which is present in the implementation. For example, using the weights defined for the 20 functions given in the example of option 1 in section 3.3.3.1, if functions 1, 2, 3, 6, 7, 15 are present in an implementation, then that implementation would receive a score of 3 + 3 + 3 + 2 + 2 + 1 which equals 14.

If option 2 is chosen, then the score of the category is the sum of the number of functions, which are important to that user, and are present in the implementation. For example, using the weights defined for the 10 functions given in the example of option 2 in section 3.3.3.1, if functions 1, 2, 3, 6, 7 are present in an implementation, then that implementation would receive a score of 1 + 1 + 1 + 0 + 0 which equals 3.

If option 3 is chosen, then the user selects one of two recommended subjective scoring methods. First, the user can subjectively score the category based on the user's overall impression of how well that category is represented in the candidate FTAM implementation. As an example, the user may decide to rate the functionality of a category, contained in a candidate FTAM implementation, on a scale of 0-10 where a score of 10 indicates that the candidate FTAM implementation contains all of the functionality in the category that the user requires, a score of 5 indicates that the candidate FTAM implementation contains an average amount of functionality in the category that the user requires, and a score of 0 indicates that the candidate FTAM implementation does not contain any of the functionality in the category that the user requires. Second, the user can subjectively score the category based on the user's overall impression of whether or not the candidate FTAM implementation acceptably performs the functions in this category. If the candidate FTAM implementation acceptably performs the functions in this category it will receive a passing score (or 1); otherwise it will receive a failing score (or 0).

The user must repeat this procedure for scoring categories, on each category. The equations for rating categories, using each of the three options, are specified in table 1.

Table 1. Equations for Rating Categories of Functionality

Option 1: $$C = \sum_{i=1}^{n} (F_i * WF_i)$$

Option 2: $$C = \sum_{i=1}^{n} (F_i)$$

Option 3: $$C = S$$

C = *Score of Category*

F_i = 1 if *Function i is Present in Implementation*, 0 *Otherwise*

WF_i = *Weight of Function i*

S = *Subjective Score of Category*

n = *Number of Functions in Category*

Second, the user determines the total functional rating, for a candidate FTAM implementation, by summing the weight of each category times the score for that category, for each of the categories. For example, if there are categories X, Y, and Z and category X has a weight of 5, category Y has a weight of 3, and category Z has a weight of 1, and an implementation has a score of 25 for category X, a score of 50 for category Y, and a score of 75 for category Z, then the functional rating for the implementation is [(5*25) + (3*50) + (1*75)] which equals 350. The equation for determining the total functional rating is specified in table 2. The candidate FTAM implementation with the highest score is likely to be the "best" implementation, functionally, for that user. Note that these ratings are not absolute ratings; another user might rate the same candidate FTAM implementations differently based on a different set of requirements.

Table 2. Equation for Functionally Rating FTAM Implementations

$$RF = \sum_{i=1}^{m} (C_i * WC_i)$$

RF = *Total Functional Rating*

C_i = *Rating of Category i*

WC_i = *Weight of Category i*

m = *Number of Categories*

3.4. Performance Evaluation Guidelines

This section recommends a procedure to evaluate the performance of candidate FTAM implementations. It is divided into three sections. Section 3.4.1 contains experiments which measure the performance of FTAM implementations. Section 3.4.2 recommends a procedure for eliminating candidate FTAM implementations which do not meet mandatory user performance requirements. Section 3.4.3 recommends a procedure for determining which remaining candidate FTAM implementation best meets the performance requirements of a user.

Time and expense are involved in evaluating the performance of candidate FTAM implementations. Although performance will be an issue for some users, it is recommended that a user first determine if a performance evaluation is required. The following examples may assist a user in determining whether performance is an issue.

Some users may not be concerned with the time required to transfer a file to a remote filestore. For example, users working on a computer system with no back-up media may develop a utility that transfers their files nightly to another computer system. Since the users do not wait for the file transfers to complete, they may not be concerned with the amount of time needed for a file to be transferred. Other users may be more concerned with the amount of time required to transfer a file. For example, an employee in a retail store may transfer inventory orders to the retail store's warehouse. Since the user in this example must ensure that the file transfer completes successfully before servicing other customers, this user would like the inventory orders to be transferred as quickly as possible. Thus, the urgency of the user's file transfer may determine whether performance is an issue.

Overall file transfer usage can also determine whether performance is an issue. An example of minimal file transfer usage is a computer system serving 50 users, and each user transfers an average of one 1024-byte file per day. Performance may not be an issue for this system, in which an average of only fifty 1024-byte files are transferred per day, because any FTAM implementation should be capable of transferring this minimal amount of data without significant delays. An example of extensive file transfer usage is a computer system serving 200 users, and each user transfers an average of 25, one-megabyte files per day. Performance may be an issue for this system, in which an average of 5000 thousand, one-megabyte files are transferred per day, because not all FTAM implementations may be able to transfer this extensive amount of data without significant delays.

3.4.1. Performance Measurements

This section contains a variety of experiments which measure the performance of FTAM implementations. These experiments provide a representative sampling of FTAM performance measurements. The experiments were created with the assistance of users and vendors. FTAM performance measurements that may be important to a user were determined, then the experiments necessary to perform these measurements were created. The user should carefully study each experiment to become familiar with the performance measurement obtained. Since there is cost associated with performing experiments, the user should only select the performance measurements which are essential. Although a variety of experiments are described in this section, it is not possible to include every performance measurement that is important to every user. The user may add any performance experiments which are not included in these guidelines, but are important to that user.

The performance experiments defined in these guidelines can be performed by either the user or the vendor. It is more practical for the vendor to perform these experiments for four reasons: (1) The vendor should have the hardware and software required to run their FTAM implementation in-house or at the user site. Thus, the user avoids procuring or leasing hardware and software to perform the experiments. (2) The vendor should have the expertise available to install the FTAM implementation on the test system, or it may already be installed. Thus, the user avoids the installation and configuration procedure for the FTAM implementation. (3) The vendor should have access to the source code for the FTAM implementation to make any modifications needed to perform the experiments. For example, the vendor may need to add code which displays times relevant to the transfer of a file. Thus, the user avoids the problems associated with acquiring and modifying the source code for the FTAM implementation. (4) The vendor should have the expertise to optimize the performance of the FTAM implementation, providing the best results possible for the implementation. It is important, when comparing performance of different FTAM implementations, to compare the best performance of each FTAM implementation so the comparison is fair. Thus, the user avoids the time required to learn how to optimize the FTAM implementation being measured. Because of these reasons, it is recommended that the user procuring an FTAM implementation request the vendor to perform the required experiments.

The person performing these experiments must follow some general guidelines. The experiments determine various measurements based on the FTAM file transfer function; however, the user may modify any experiment to measure FTAM file access or file management functions. To measure FTAM file access functionality, the user should request that one of the FADU functions - erase, extend, insert, locate, read, or replace - be performed in lieu of transferring an entire file. To measure FTAM file management functionality, the user should request that one of the file management functions - create a file, delete a file, read file attributes, or change file attributes - be performed in an experiment in lieu of transferring an entire file. The user should note that not all FTAM file access functions may be available in an FTAM implementation. An implementation may also combine multiple FTAM functions into one composite FTAM command. For example, an FTAM implementation may allow a user to move a file. The move command is implemented by transferring a file from its source to its destination, then deleting the source file. It is recommended to measure the FTAM functions that comprise the composite command (e.g., transfer then delete a file) rather than measure the performance of composite FTAM commands (e.g., move a file).

A user may want to measure the performance of an FTAM implementation which acts as both an initiator and a responder. In this case the experiments are performed by transferring files between an FTAM initiator and an FTAM responder of the same vendor. The initiator and responder applications must reside on separate hardware.

A user may want to measure the performance of an FTAM initiator or an FTAM responder only. In the case of measuring the performance of an FTAM initiator, the experiments are performed by transferring files between an FTAM initiator of a specific vendor and a reference FTAM responder. A reference FTAM responder is a responder application specified by the user against which the performance of all FTAM initiators is measured.

The procedure for measuring the performance of an FTAM responder is similar to the procedure for measuring the performance of an FTAM initiator. Experiments are performed by transferring files between an FTAM responder of a specific vendor and a reference FTAM initiator. It is beyond the scope of this document to recommend procedures for selecting a reference FTAM initiator or FTAM responder.

The guidelines above define a homogeneous environment for performing the experiments. This environment ensures that only one variable is measured (i.e., only one FTAM implementation), and that the performance measurements are fair to all vendors. The user should note that although a homogeneous environment ensures fairness, experimental performance measurements may vary from performance measurements obtained in a production environment. This is because in a production environment a user may transfer files between FTAM implementations of different vendors.

A basic measurement made in most of the experiments is the File Transfer Time interval. This interval, referred to as FTT, is defined as the ending file transfer time minus the beginning file transfer time. The ending file transfer time is the time the file is deselected (i.e., the file selection regime is terminated). The beginning file transfer time is the time the file is selected (i.e., the file selection regime is established). The measurement of FTT does not include the time required by an FTAM initiator to establish and terminate the FTAM application association. The basis for this decision is discussed below.

Some FTAM applications implement the establishment of an FTAM association as an operation separate from FTAM commands. As a result, a user can perform multiple FTAM commands during one association. An example sequence of events for this type of FTAM implementation could be (1) open an FTAM association, (2) transfer a file, (3) transfer a second file, (4) delete a file, and (5) close the association. Since multiple FTAM commands (e.g., two file transfers and one file delete) can be performed during one association, it is not desirable to include the time required to establish and terminate an FTAM association in FTT.

A graphic representation of the FTT interval is presented in figure 12. Although the experiments in this section are structured to measure FTT, any experiment may be modified to include the measurement of the FTAM application association.

The FTT interval provides a simple measurement for comparing the performance of FTAM implementations. For each experiment which determines an FTT interval, the user may also request the computation of the file transfer throughput. File transfer throughput is defined as the number of bits in a file divided by the FTT interval.

To obtain accurate experiment results, measurements must be taken repeatedly and averaged. This is because there are several factors which may vary the result each time the measurement is performed. These factors include utilization of the CPU by other processes, utilization of the LAN or WAN by other users of the network, retransmissions of packets due to transmission errors, and others. The person conducting the experiments must determine the number of repetitions required to stabilize the results of the experiment. This number may vary from experiment to experiment. In order to determine the number of repetitions for a measurement, the measurement should be taken some reasonable number of times (e.g., 20 times) and the results averaged. The measurement should then be taken an additional number of times (e.g., 10 additional times), and averaged over all times (30 total times), to determine whether the additional measurements affected the results. This process should be repeated, increasing the number of additional measurements, until the results are stable to the satisfaction of the person conducting the experiment.

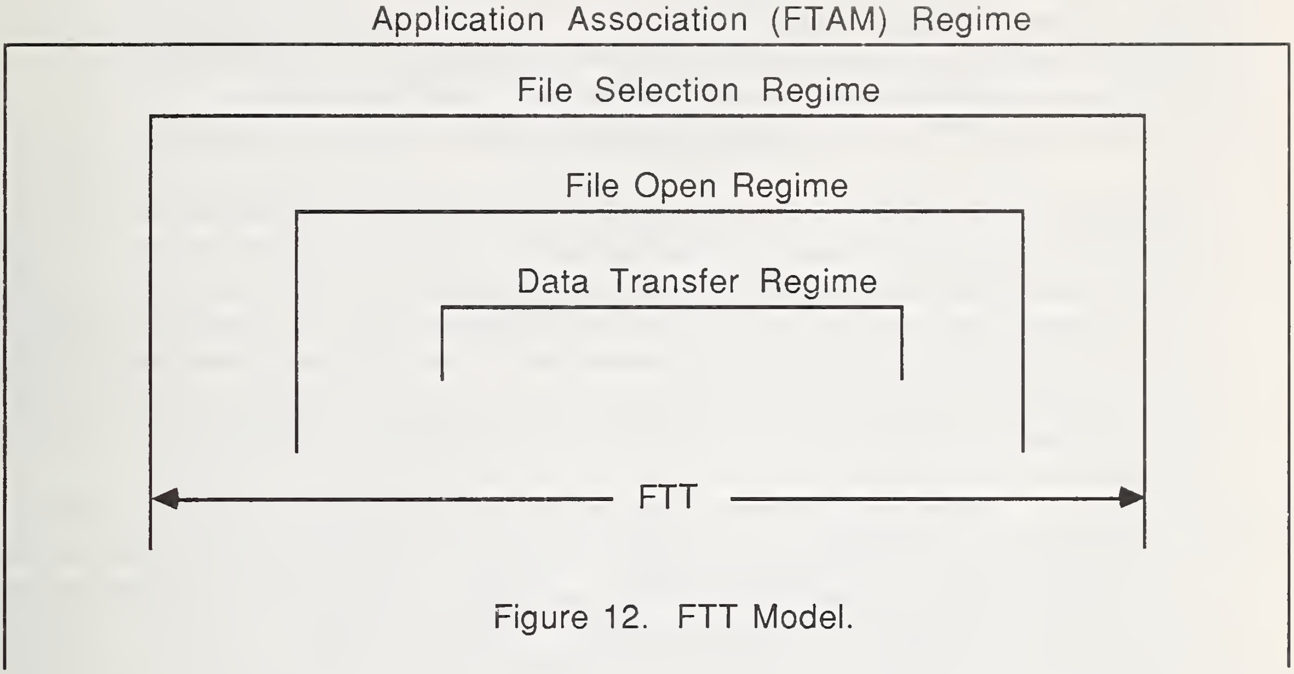


Figure 12. FTT Model.

Several general inputs must be provided by the user in order to perform the experiments. Inputs 1-5 are constant for all the experiments; inputs 6-7 may vary from experiment to experiment. A description of these inputs follows:

(1) The user's FTAM configuration. As described in section 3.1, the user should determine the FTAM configuration to be used. The performance of candidate FTAM implementations should be measured on a configuration which matches the user's FTAM configuration.

(2) The model(s) of the computer(s) and disk to be used in the experiments. Many computer systems are available in a variety of models, where the low-end models have a slower CPU and disk access time than the high-end models. If the user already has the hardware for the FTAM implementation, then the model number should be specified to the person conducting the experiment. If the user is procuring hardware with the FTAM implementation, then the user may provide the vendor with input on the user's

performance requirements. The vendor may then recommend which models of computers and disk are likely to meet the performance measurements required by the user. The user should be careful, when specifying performance requirements to the vendor, not to specify a level of performance which greatly exceeds the user's requirements. This may increase hardware costs unnecessarily. The person conducting the experiments should run the experiments on the model of hardware that will be used by the user.

(3) The network type. The user should provide the vendor with the type of Local Area Network or Wide Area Network, and if relevant, the network speed and network configuration (i.e., public vs. private WAN). If possible, the person conducting the experiments should perform the experiments on a network of the same type, speed, and configuration as the user's network. Otherwise, the user should note that the results of the experiments may differ from the user's actual performance. For example, if the person conducting the experiments performs them on a LAN while the user's configuration contains a private WAN, which may have a significantly slower transmission rate, the user's actual performance may be worse than the performance measured in the experiments.

(4) The experimental environment. Two types of user environments are defined for performing the experiments. The first environment, referred to as the ideal environment, is one in which there are no application processes competing with the FTAM implementation for the resources of the CPU, disk, and network. This environment is useful for measuring the maximum performance of an FTAM implementation. The second environment, referred to as the real environment, is one in which there are application processes competing for the resources of the CPU, disk, and network with the FTAM implementation. The percent of CPU, disk, and network resources utilized by the competing application processes should reflect the utilization of these resources by application processes normally run on the user's system. This environment is useful for measuring the typical performance of an FTAM implementation. On a single user system the only resource for which there may be competition is the network, since multiple processes cannot run in this environment. The experiments may be performed, as requested by the user, in either an ideal or real environment, or both. It is beyond the scope of these guidelines to recommend procedures for performing experiments in a real environment.

(5) The file transfer direction. The user can specify the direction, relative to the initiator and responder, in which files are to be transferred. The experiments can be performed using one of three methods: (1) The initiator receives a file from the responder. (2) The initiator sends a file to the responder. (3) The initiator receives a file from the responder and sends a file to the responder. Some users will be interested in one file transfer direction (i.e., sending or receiving a file) while other users will be interested in both file transfer directions.

(6) The FTAM document type. The user can specify which FTAM document type to use for the experiment. The following document types are referenced in Version 3 of the OIW Agreements: FTAM-1, FTAM-2, FTAM-3, NBS-6, NBS-7, NBS-8, and NBS-9. The user may also request that the person performing the experiments reference the NIST FTAM Interoperability Test suite (NISTIR 4435), and based on the document type specified by the user, obtain specific file contents to be transferred in the experiments.

(7) The filesize. The filesize refers to the total number of bytes in the file. Only one filesize is input to an experiment, and it should represent the average size of a file to be transferred by the user's FTAM implementation. If the user has an existing file transfer application, the average filesize can be calculated by dividing the number of bytes transferred in files in a certain time period, such as a day, by the number of files transferred in that time period. Otherwise, the user may reference the NIST FTAM Interoperability Test suite and obtain a filesize based on the document type requested by the user.

The performance experiments in this section are presented with the following format. Each experiment has five parts: introduction, methodology, user input, example, and summary of results. The introduction contains the purpose of the experiment. It describes what the experiment measures and its applicability to a user. The methodology section contains a diagram followed by instructions for the person conducting the experiment. The diagram shows the file transfer path between the users, the FTAM initiator, and the FTAM responder involved in the experiment; the instructions specify how the experiment is to be performed. The user input section lists the information the user must provide to the person conducting the experiment. Certain user inputs remain constant for each experiment, i.e., the FTAM configuration, the model(s) of the computer(s) to be used, the network type, the experimental environment, and the file transfer direction. These requirements are not repeated in the user input section of each experiment. Following user input is an example for each experiment. Both user input values and experiment results are presented in the example. The input values and results in the examples are simulated. The final section, summary of results, discusses the results presented in the example section. Formulas for calculating results as well as an interpretation of the results are provided.

Six performance experiments are presented in this section. The first four experiments measure FTT. Experiment 1 provides the base measurement; FTT is calculated for an FTAM file. Experiments 2, 3 and 4 each introduce one additional factor that may affect FTT. Experiment 2 measures the impact of varying the filesize. Experiment 3 measures the impact of varying the document type. Experiment 4 measures the impact of estimated file transfer usage. Comparing the measurements of FTT in Experiments 2, 3 and 4 to the measurement of FTT in Experiment 1, the user can determine the effect of the factor measured in the experiment. Except for the filesize input for Experiment 2 and the document type input for Experiment 3, the user must input the same filesize and document type values for each experiment for the comparisons to be fair.

Experiments 5 and 6 measure file transfer capacity and system utilization respectively. Experiment 5 measures the amount of time required by an FTAM implementation to transfer a maximum number of files. Experiment 6 determines the amount of CPU and I/O utilized by an FTAM implementation.

3.4.1.1. Experiment 1

(1) Introduction

Experiment 1 measures FTT for a single file. This measurement represents a baseline file transfer time interval; FTT measurements in other experiments can be compared to the FTT measurement in this experiment to determine the impact of other factors on FTT. Since this experiment measures no other factors, the measurement of FTT in this experiment also represents the optimum file transfer time interval.

(2) Methodology

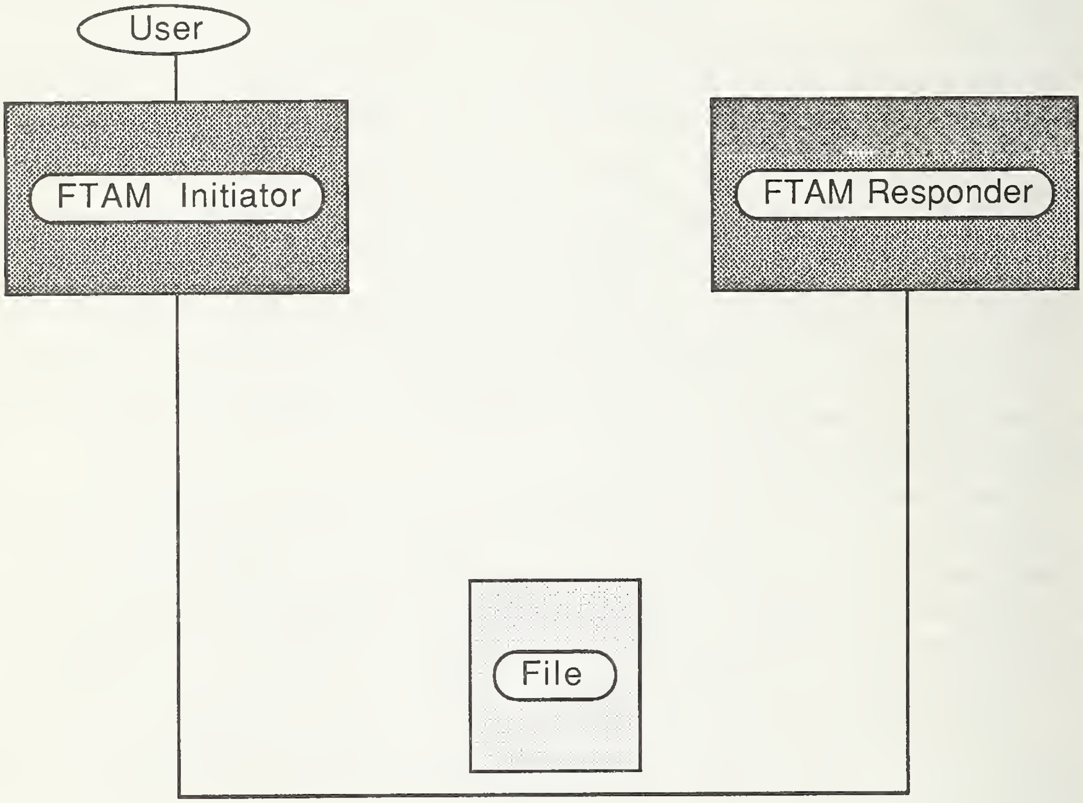


Figure 13. Experiment 1 - File Transfer Path.

This experiment involves one FTAM initiator supporting one user and one FTAM responder. The initiator and the responder must reside on different hardware. FTAM files are transferred serially; the FTAM user should not transfer a file until the previous file has been transferred. This may be accomplished by transferring files according to a predefined time interval. The interval must be long enough to allow the previous file to complete its transfer (i.e., for the previous file to be deselected). All files transferred must be of uniform size and document type. For each file transferred, the person conducting the experiment should log the beginning and ending file transfer times.

(3) User Input

The user must provide the person conducting the experiment the following information:

- the size of the files to be transferred

- the document type of the files to be transferred

(4) Example

For this example, a government weather agency is interested in procuring an FTAM implementation. The agency collects most of its weather data via instruments located in satellites, hot air balloons, and ocean surface buoys. The data from these sources are received at various ground locations and examined to remove noise and spurious observations. The data are then transferred to a central site for detailed analysis. The remaining data collected by the agency originate from weather stations located throughout the country. These stations send information to the central site depicting the weather conditions in their region.

The weather data are currently transferred from the ground locations and weather stations to the central site via File Transfer Protocol (FTP) implementations. Due to the migration to GOSIP the FTP implementations are to be replaced by FTAM implementations. The government agency, in cooperation with a vendor of choice, have determined that each ground location and weather station will use one FTAM initiator to send the weather data. Each initiator will support several users. The central site will support one FTAM responder for receiving the weather data to be analyzed. The central site will be connected to the ground locations and weather stations by a WAN. A diagram of this FTAM configuration is presented in figure 14. Due to the voluminous data transferred to the central site, the government agency wants to test performance across the WAN.

For Experiment 1 the agency requests that the experiment be performed using a document type of FTAM-3 and a filesize of 100,000 bytes. The person conducting the experiment provides the agency the following results:

Table 3. Example Results for Experiment 1

File Number	Beginning File Transfer Time	Ending File Transfer Time	FTT
1	12:00.00 PM	12:00.22 PM	22 sec
2	12:05.00 PM	12.05.29 PM	29 sec
3	12:10.00 PM	12:10.24 PM	24 sec
4	12:15.00 PM	12:15.26 PM	26 sec
5	12:20.00 PM	12:20.34 PM	34 sec
6	12:25.00 PM	12.25.29 PM	29 sec
7	12:30.00 PM	12:30.21 PM	21 sec
8	12:35.00 PM	12:35.27 PM	27 sec
9	12:40.00 PM	12.40.30 PM	30 sec
10	12:45.00 PM	12.45.21 PM	21 sec

The minimum FTT measurement is 21 seconds.
The maximum FTT measurement is 34 seconds.
The average FTT measurement is 26.3 seconds.

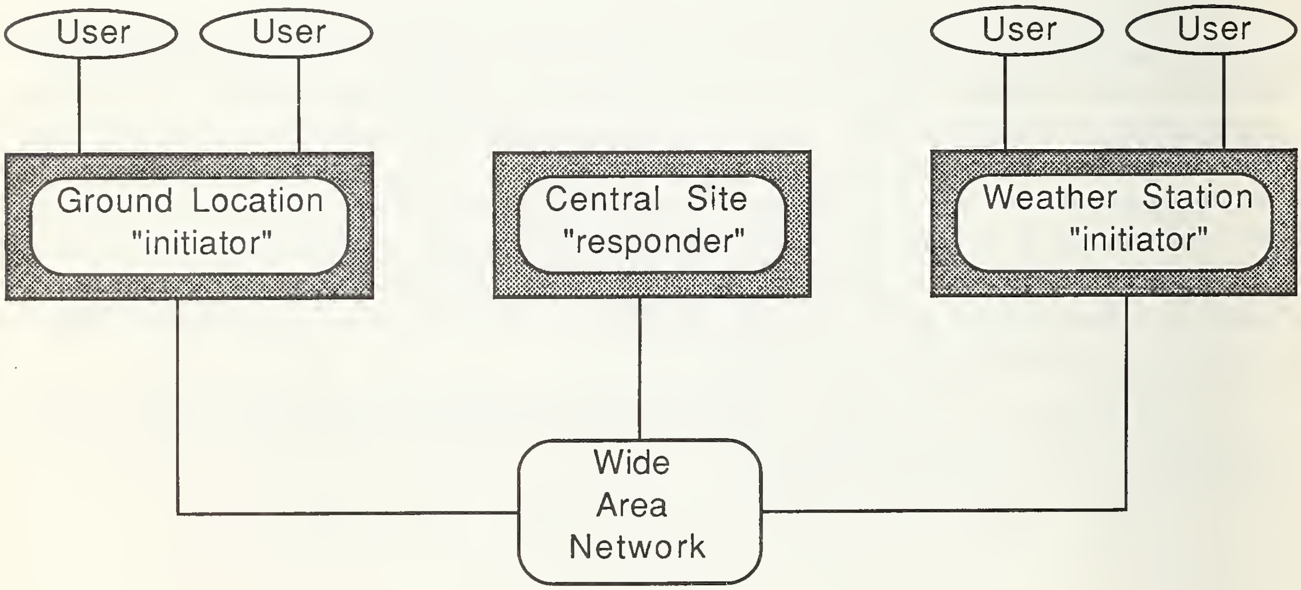


Figure 14. FTAM Configuration Used in Experiment Examples.

(5) Summary of Results

The person conducting the experiment determined that the experiment results were stable with the sending of 10 files. For each file transferred the following information is provided: the file number, the beginning file transfer time, the ending file transfer time, and FTT. FTT, which is the only calculated value, is the ending file transfer time minus the beginning file transfer time.

The minimum, maximum, and average FTT measurements are also provided. The minimum and maximum FTT measurements are the fastest and slowest file transfer times respectively. The average FTT measurement is the sum of all FTT measurements divided by the number of files transferred. See table 4 for the formula. The average FTT measurement for this experiment represents the optimum file transfer time interval for sending a file of the size and document type requested.

The results recorded in this experiment do not represent actual data collected. They are listed for example purposes only. While the results simulate data obtained from one vendor's FTAM implementation, the experiment must be repeated for each vendor's FTAM implementation which is of interest to the user.

Table 4. Equation for Average FTT

$$AFTT = [\sum_{i=1}^{m} (FTT_i)] / m$$

$AFTT = Average\ FTT$

$FTT_i = FTT$ for $File\ i$

$m = Number\ of\ Files\ Transferred$

3.4.1.2. Experiment 2

(1) Introduction

Experiment 2 measures FTT for a single FTAM file. The filesize used in this experiment must be different from the filesize used in Experiment 1. Comparing the results of this experiment to Experiment 1, the user can determine the impact of varying the filesize on the time required to transfer a file.

(2) Methodology

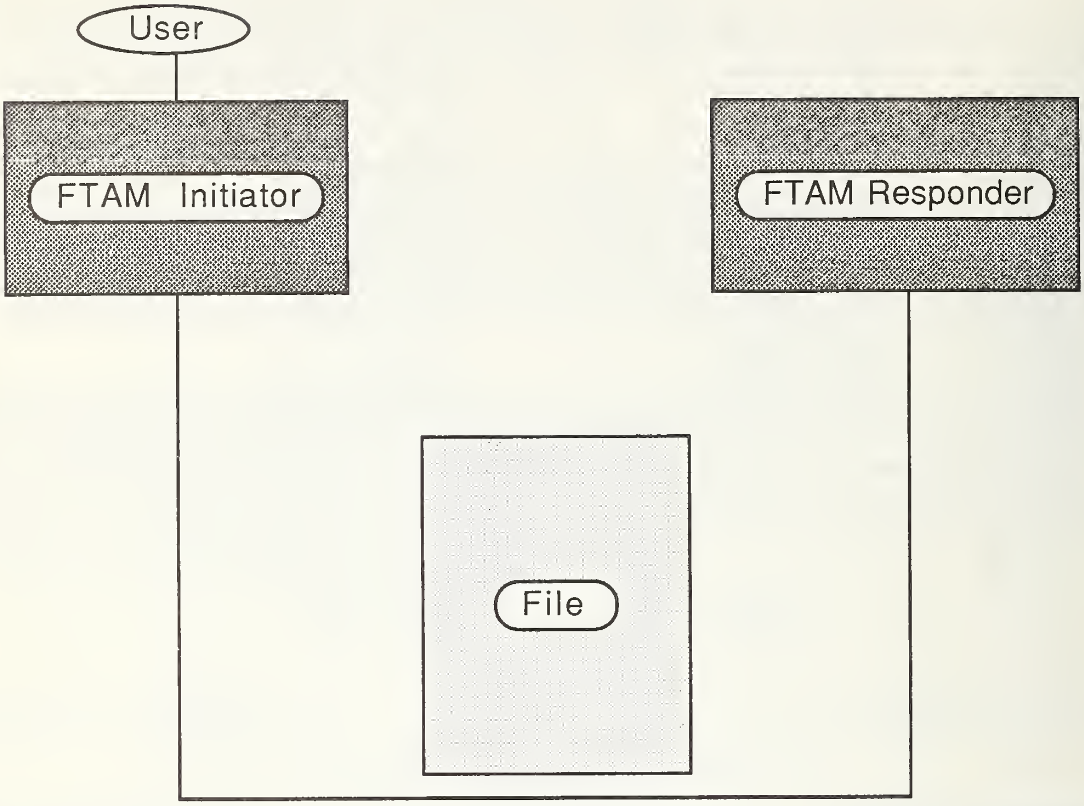


Figure 15. Experiment 2 - File Transfer Path.

This experiment involves one FTAM initiator supporting one user and one FTAM responder. The initiator and the responder must reside on different hardware. FTAM files are transferred serially; the FTAM user should not transfer a file until the previous file has been transferred. This may be accomplished by transferring files according to a predefined time interval. The interval must be long enough to allow the previous file to be transferred (i.e., for the previous file to be deselected). All files transferred must be of uniform size and document type. For each file transferred, the person conducting the experiment should log the beginning and ending file transfer times.

(3) User Input

The user must provide the person conducting the experiment the following information:

- the size of the files to be transferred

- the document type of the files to be transferred

(4) Example

During hurricane season the government weather agency described in Experiment 1 increases the amount of data transmitted from ocean surface buoys. Thus, it is of interest to the agency to determine the time required to transfer a file representing hurricane data. This file is larger than the typical file transferred from a ground location to the central site. For this experiment the agency requests that the experiment be performed using a document type of FTAM-3 and a filesize of 150,000 bytes. The person conducting the experiment provides the agency the following results:

Table 5. Example Results for Experiment 2

File Number	Beginning File Transfer Time	Ending File Transfer Time	FTT
1	12:00.00 PM	12:00.34 PM	34 sec
2	12:05.00 PM	12.05.34 PM	34 sec
3	12:10.00 PM	12:10.39 PM	39 sec
4	12:15.00 PM	12:15.36 PM	36 sec
5	12:20.00 PM	12:20.34 PM	34 sec
6	12:25.00 PM	12.25.39 PM	39 sec
7	12:30.00 PM	12:30.31 PM	31 sec
8	12:35.00 PM	12:35.37 PM	37 sec
9	12:40.00 PM	12.40.40 PM	40 sec
10	12:45.00 PM	12.45.33 PM	33 sec

The minimum FTT measurement is 31 seconds.
The maximum FTT measurement is 40 seconds.
The average FTT measurement is 35.7 seconds.

The difference between the average FTT measurement of Experiment 1 (26.3 s) and the average FTT measurement of Experiment 2 is 9.4 seconds.

(5) Summary of Results

The person conducting the experiment determined that the experiment results were stable with the sending of 10 files. For each file transferred the following information is provided: the file number, the beginning file transfer time, the ending file transfer time, and FTT. FTT, which is the only calculated value, is the ending file transfer time minus the beginning file transfer time.

The minimum, maximum, and average FTT measurements are also provided. The minimum and maximum FTT measurements are the fastest and slowest file transfers respectively. The average FTT measurement is the sum of all FTT measurements divided by the number of files transferred. See table 4 for the formula. The final calculated value is the difference between the average FTT measurement for this experiment and the average FTT measurement for Experiment 1.

This experiment is identical to Experiment 1 except that the size of the file transferred is larger. The user should expect the amount of time required to transfer a file of this size to be greater than that of Experiment 1, because more data must be transferred. Likewise, if the user requested a filesize less than that of Experiment 1, the user should expect the amount of time required to transfer the file to be less. The exact amount will be determined by the size of the file.

The results recorded in this experiment do not represent actual data collected. They are listed for example purposes only. While the results simulate data obtained from one vendor's FTAM implementation, the experiment must be repeated for each vendor's FTAM implementation which is of interest to the user.

3.4.1.3. Experiment 3

(1) Introduction

Experiment 3 measures FTT for a single FTAM file. The document type used in this experiment must be different from the document type used in Experiment 1. Comparing the results of this experiment to Experiment 1, the user can determine the impact of varying the document type on the time required to transfer a file.

(2) Methodology

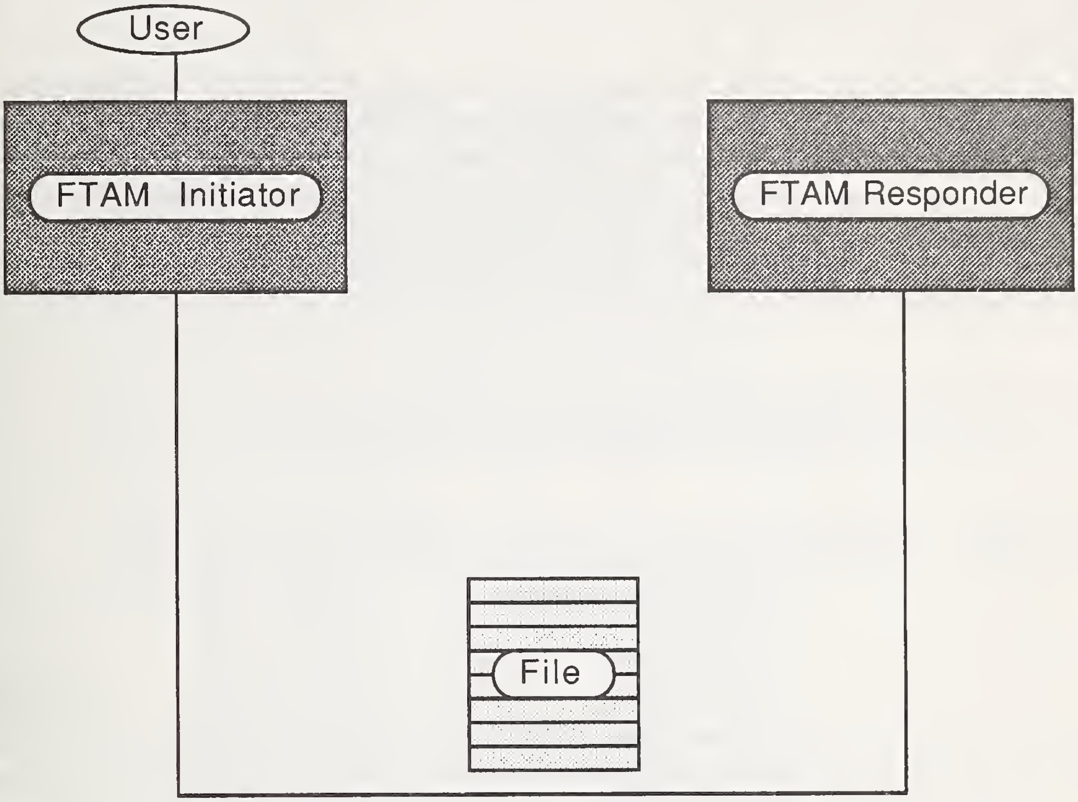


Figure 16. Experiment 3 - File Transfer Path.

This experiment involves one FTAM initiator supporting one user and one FTAM responder. The initiator and the responder must reside on different hardware. FTAM files are transferred serially; the FTAM user should not transfer a file until the previous file has been transferred. This may be accomplished by transferring files according to a predefined time interval. The interval must be long enough to allow the previous file to be transferred (i.e., for the previous file to be deselected). All files transferred must be of uniform size and document type. For each file transferred, the person conducting the experiment should log the beginning file transfer time and the ending file transfer time.

(3) User Input

The user must provide the person conducting the experiment the following information:

- the size of the files to be transferred

- the document type of the files to be transferred

(4) Example

Some weather data received by the government agency described in Experiment 1 originate from weather stations located throughout the country. In contrast to the binary data transferred from ground locations, the weather stations transfer text data. Thus, it is of interest to the agency to determine the time required to transfer a text file. For this experiment the agency requests that the experiment be performed using a document type of FTAM-2 and a filesize of 100,000 bytes. The person conducting the experiment provides the agency the following results:

Table 6. Example Results for Experiment 3

File Number	Beginning File Transfer Time	Ending File Transfer Time	FTT
1	12:00.00 PM	12:00.32 PM	32 sec
2	12:05.00 PM	12.05.29 PM	29 sec
3	12:10.00 PM	12:10.31 PM	31 sec
4	12:15.00 PM	12:15.26 PM	26 sec
5	12:20.00 PM	12:20.35 PM	35 sec
6	12:25.00 PM	12.25.29 PM	29 sec
7	12:30.00 PM	12:30.31 PM	31 sec
8	12:35.00 PM	12:35.27 PM	27 sec
9	12:40.00 PM	12.40.30 PM	30 sec
10	12:45.00 PM	12.45.24 PM	24 sec

The minimum FTT measurement is 24 seconds.
The maximum FTT measurement is 35 seconds.
The average FTT measurement is 29.4 seconds.

The difference between the average FTT measurement of Experiment 1 (26.3 s) and the average FTT measurement of Experiment 3 is 3.1 seconds.

(5) Summary of Results

The person conducting the experiment determined that the experiment results were stable with the sending of 10 files. For each file transferred the following information is provided: the file number, the beginning file transfer time, the ending file transfer time, and FTT. FTT, which is the only calculated value, is the ending file transfer time minus the beginning file transfer time.

The minimum, maximum, and average FTT measurements are also provided. The minimum and maximum FTT measurements are the fastest and slowest file transfers respectively. The average FTT measurement is the sum of all FTT measurements divided by the number of files transferred. See table 4 for the formula. The final calculated value is the difference between the average FTT measurement for this experiment and the average FTT measurement for Experiment 1.

This experiment is identical to Experiment 1 except that the document type used in this experiment (i.e., FTAM-2, sequential text) is different than the document type used in Experiment 1 (i.e., FTAM-3, unstructured binary). The FTAM-2 document type contains structuring and semantic information not contained in the FTAM-3 document type. Thus, the user should expect the amount of time required to transfer a file of type FTAM-2 to be greater than that of type FTAM-3, because the structure and semantics of the file must be transferred in addition to the data contained in the file. The exact amount of additional time required will be determined by the document type of the file being transferred.

The results recorded in this experiment do not represent actual data collected. They are listed for example purposes only. While the results simulate data obtained from one vendor's FTAM implementation, the experiment must be repeated for each vendor's FTAM implementation which is of interest to the user.

3.4.1.4. Experiment 4

(1) Introduction

In a typical FTAM environment, files are transferred randomly by users. A user can create a table of sample file transfer times which estimates usage of the user's FTAM implementation. This experiment measures FTT based on estimated file transfer usage.

(2) Methodology

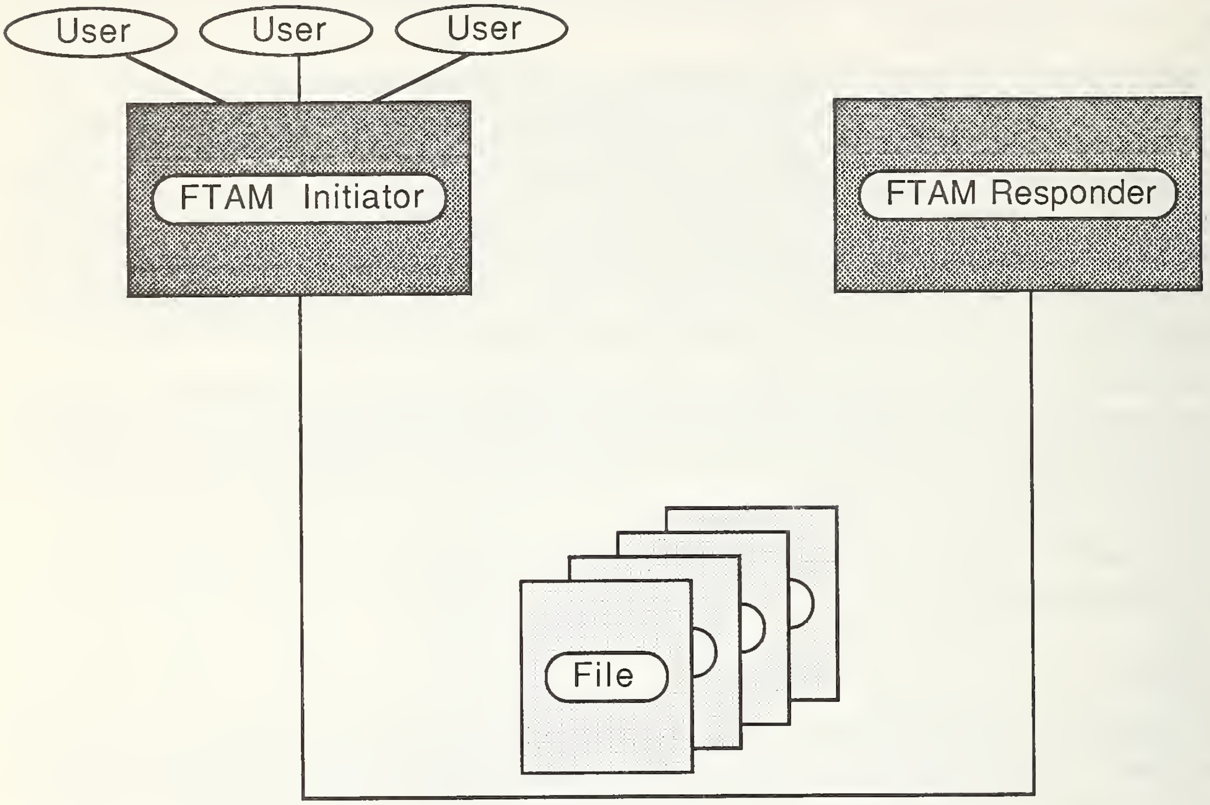


Figure 17. Experiment 4 - File Transfer Path.

This experiment involves one FTAM initiator supporting multiple users and one FTAM responder. The users initiate file transfers based on information in a user defined file transfer timetable. The table should include the first, last, and various intermediate file transfer time entries along with a number of files to be transferred for each entry. If multiple files are to be transferred for a single time entry, the files must be transferred simultaneously. If the users are to send and receive files, the file transfer direction for each file must also be included in the table. All files must be of uniform size and document type. For each file transferred, the person conducting the experiment should log the beginning file transfer time and ending file transfer time.

(3) User Input

The user must provide the person conducting the experiment the following information:

- the size of the files to be transferred

- the document type of the files to be transferred

- the file transfer timetable

(4) Example

The government weather agency described in Experiment 1 had previously monitored the number of files containing binary data transferred from ground locations to the central site. Based on this information the agency has requested that the experiment be performed over the course of 1 hour using a document type of FTAM-3 and a filesize of 100,000 bytes. Files are to be transferred according to the following file transfer timetable.

Table 7. Example File Submission Timetable

Time	Number Of Files
12:00	2
12:02	1
12:05	1
12:15	3
12:20	2
12:30	2
12:33	1
12:37	1
12:45	5
12:59	3

The person conducting the experiment provides the agency the following results:

Table 8. Example Results for Experiment 4

File Number	Beginning File Transfer Time	Ending File Transfer Time	FTT
1	12:00.00 PM	12:00.33 PM	33 sec
2	12:00.00 PM	12.00.35 PM	35 sec
3	12:02.00 PM	12:02.28 PM	28 sec
4	12:05.00 PM	12:05.23 PM	23 sec
5	12:15.00 PM	12:15.39 PM	39 sec
6	12:15.00 PM	12.15.39 PM	39 sec
7	12:15.00 PM	12:15.37 PM	37 sec
8	12:20.00 PM	12:20.32 PM	32 sec
9	12:20.00 PM	12.20.35 PM	35 sec
10	12:30.00 PM	12.30.33 PM	33 sec
11	12:30.00 PM	12:30.34 PM	34 sec
12	12:33.00 PM	12.33.24 PM	24 sec
13	12:37.00 PM	12:37.28 PM	28 sec
14	12:45.00 PM	12:45.44 PM	44 sec
15	12:45.00 PM	12:45.47 PM	47 sec
16	12:45.00 PM	12.45.40 PM	40 sec
17	12:45.00 PM	12:45.39 PM	39 sec
18	12:45.00 PM	12:45.41 PM	41 sec
19	12:59.00 PM	12.59.38 PM	38 sec
20	12:59.00 PM	12.59.35 PM	35 sec
21	12:59.00 PM	12.59.37 PM	37 sec

The minimum FTT measurement is 24 seconds.
The maximum FTT measurement is 47 seconds.
The average FTT measurement is 35.3 seconds.

The difference between the average FTT measurement of Experiment 1 (26.3 s) and the average FTT measurement of Experiment 4 is 9 seconds.

(5) Summary of Results

For each file transferred the following information is provided: the file number, the beginning file transfer time, the ending file transfer time, and FTT. FTT, which is the only calculated value, is the ending file transfer time minus the beginning file transfer time.

The minimum, maximum, and average FTT measurements are also provided. The minimum and maximum FTT measurements are the fastest and slowest file transfers respectively. The average FTT measurement is the sum of all FTT measurements divided by the number of files transferred. See table 4 for the formula. The final calculated value is the difference between the average FTT measurement for this experiment and the average FTT measurement for Experiment 1.

The results of this experiment reflect that FTAM implementations capable of supporting simultaneous FTAM associations can transfer multiple files in less time than implementations not capable of maintaining simultaneous FTAM associations. This is observed with the FTT measurements for files transferred simultaneously. The time required to transfer each of several files initiated simultaneously is not a multiple of the time required to transfer one file (e.g., the time required to transfer two files is not twice the time required to transfer one file). The explanation for this follows: The establishing of an FTAM association, as well as other FTAM services, occurs as a confirmed operation. For example, to establish an FTAM association, an initiator sends an association establishment request to the responder. The responder receives the request, performs some local processing such as the negotiation of permissible document types, then returns a confirmation for the association request. The initiator must wait for the confirmation to return before requesting the next service (e.g., selecting a file). If the FTAM initiator supports multiple simultaneous FTAM associations, the initiator can process other FTAM file transfers while waiting for the confirmation of the association establishment to return.

This experiment is different from Experiment 1 in that FTAM files are transferred based on estimated file transfer usage. The average FTT measurement in this experiment can be compared to that in Experiment 1 to determine the effect of this difference. The user can expect that additional time may be required to transfer a file. This is because the FTAM implementation must transfer files initiated simultaneously by different users. The amount of additional time will be affected by the frequency of file transfers.

The results recorded in this experiment do not represent actual data collected. They are listed for example purposes only. While the results simulate data obtained from one vendor's FTAM implementation, the experiment must be repeated for each vendor's FTAM implementation which is of interest to the user.

3.4.1.5. Experiment 5

(1) Introduction

This experiment measures the file transfer capacity of an FTAM implementation. File transfer capacity is useful for determining the amount of time required to transfer a maximum number of files.

(2) Methodology

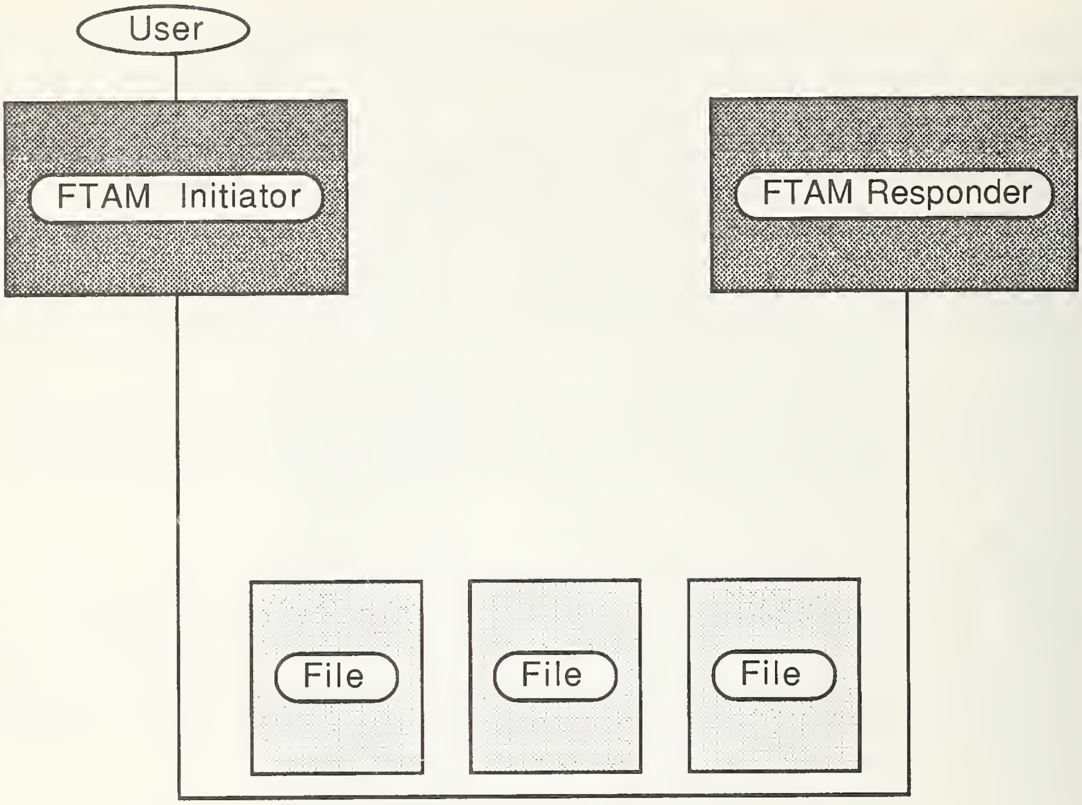


Figure 18. Experiment 5 - File Transfer Path.

For this experiment a user selects a length of time during which heavy usage of the FTAM implementation is expected (e.g., 1 h), then estimates the maximum number of files that would be transferred by the FTAM implementation during that length of time (e.g., 20 files). The user inputs this number of files to the experiment. In addition, the user specifies a file transfer period for the files. The file transfer period represents the time interval between successive file transfers. This period should be a fraction (e.g., 1/2) of the average file transfer time interval for one file, which is calculated by dividing the length of time selected by the user by the number of files. For example, if the user selects a time length of 1 hour, and estimates that the transfer of 20 files represents heavy file transfer usage, then the average file transfer time interval for one file is 3 minutes. The user should take a fraction of this interval (e.g., 1/2), and specify that the files be transferred at 90-second intervals.

The experiment involves one FTAM initiator supporting one user and one FTAM responder. The initiator and responder must reside on different hardware. All files must be transferred according to a user defined file transfer period, and be of uniform size and document type. The person conducting the experiment should log the beginning file transfer time for the first file and the ending file transfer time for the last file.

(3) User Input

The user must provide the person conducting the experiment the following information:

- the size of the files to be transferred

- the document type of the files to be transferred

- the number of files to be transferred

- the file transfer period

(4) Example

The government agency described in Experiment 1 receives periodic weather reports from weather stations around the country. During times of turbulent weather, such as storms, the weather stations send reports at increased frequencies. It is of interest to the agency to determine if this heavy file transfer usage warrants an additional FTAM responder at the central site. Based on a previous analysis, the agency has determined that during any given hour, not more than 120 weather station reports will be transferred to the central site. For this experiment the agency requests the transfer of 120 files using a filesize of 100,000 bytes and a document type of FTAM-2. Files should be transferred approximately every 15 seconds. The person conducting the experiment provides the agency the following results:

Table 9. Example Results for Experiment 5

Number of Files	Beginning File Transfer Time For First File	Ending File Transfer Time For Last File	Total File Transfer Time Interval
120	01:00.00 PM	01:38.57 PM	38 minutes 57 seconds

(5) Summary of Results

For this experiment the following information is provided: the number of files transferred, the beginning file transfer time for the first file transfer, the ending file transfer time for the last file transfer, and total file transfer time interval. The total file transfer time interval, which is the only calculated value, is the ending file transfer time of the last file transfer minus the beginning file transfer time of the first file transfer.

For this experiment the user estimates the maximum number of files that would be transferred during a predetermined length of time. Comparing the total file transfer time interval of this experiment with the length of time selected by the user, the user can determine whether the FTAM implementation can satisfactorily transfer the maximum number of files.

The results recorded in this experiment do not represent actual data collected. They are listed for example purposes only. While the results simulate data obtained from one vendor's FTAM implementation, the experiment must be repeated for each vendor's FTAM implementation which is of interest to the user.

3.4.1.6. Experiment 6

(1) Introduction

This experiment measures the utilization of a system by an FTAM implementation. System utilization is useful for determining the percentage of the CPU and I/O used by an FTAM implementation. This experiment can only be performed on multi-user systems capable of calculating the percentage of CPU and I/O used by a process or processes.

(2) Methodology

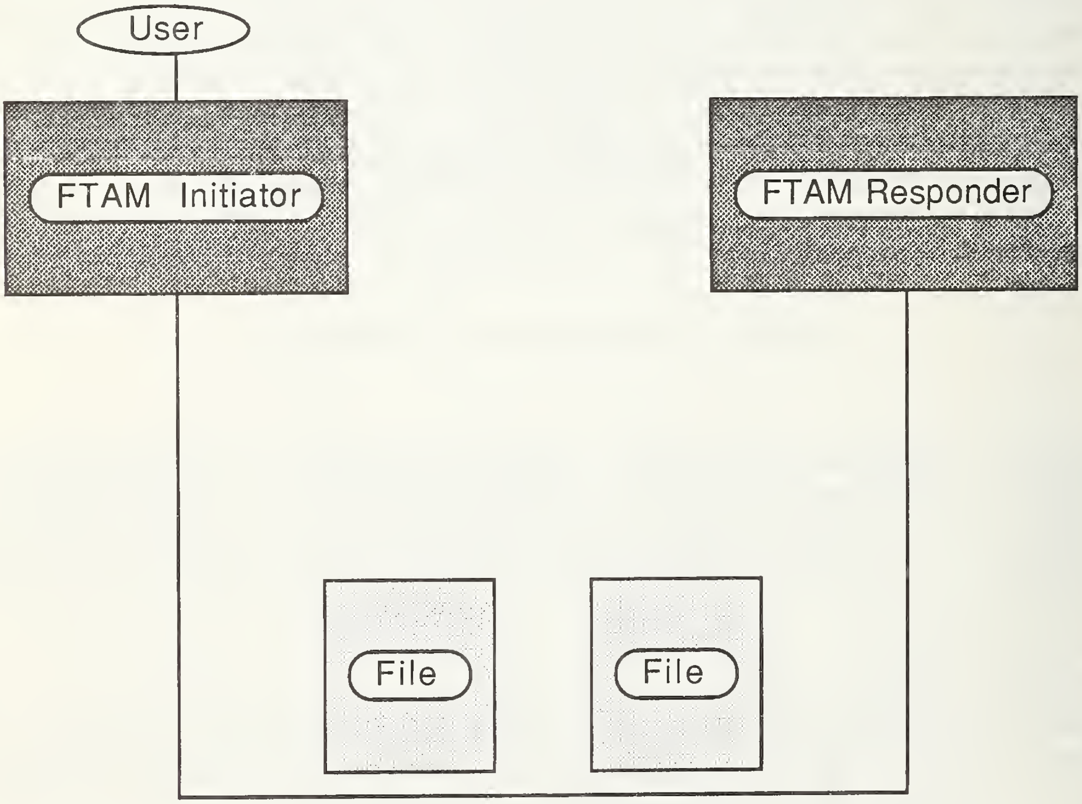


Figure 19. Experiment 6 - File Transfer Path.

For this experiment a user selects an arbitrary period of time (e.g., 1 h), then estimates a typical number of files that would be transferred by the FTAM implementation during that length of time (e.g., 20 files). The user inputs both the length of time and number of files to the experiment. In addition, the user specifies a file transfer period for the files. The file transfer period represents the time interval between successive file transfers. This period should be a fraction (e.g., 1/2) of the average file transfer time interval for one file, which is calculated by dividing the length of time selected by the user by the number of files. For example, if the user selects a time length of 1 hour, and estimates that the transfer of 20 files represents typical file transfer usage, then the average file transfer time interval for one file is 3 minutes. The user should take a fraction of this interval (e.g., 1/2), and specify that the files be transferred at 90-second intervals.

The experiment involves one FTAM initiator supporting one user and one FTAM responder. The initiator and responder must reside on different hardware. All files must be transferred according to a user defined file transfer period, and be of uniform size and document type. The person conducting the experiment should log the beginning file transfer time for the first file transfer and the ending file transfer time for the last file transfer.

(3) User Input

The user must provide the person conducting the experiment the following information:

- the size of the files to be transferred

- the document type of the files to be transferred

- the number of files to be transferred

- the length of time during which the files are to be transferred

- the file transfer period

(4) Example

The government agency described in Experiment 1 receives weather data at a central site. These data are used as input to computational tools which analyze the data in detail. Since the computational tools utilize a large percentage of the computer system's CPU, it is of interest to the agency to measure the CPU and I/O used by the FTAM responder at the central site.

Based on a previous analysis, the agency has determined that during 1 hour, 60 files would typically be transferred from ground locations to the central site. For this experiment the agency requests the transfer of 60 files using a filesize of 100,000 bytes and a document type of FTAM-3. The files are to be transferred for 1 hour at 30-second intervals. The person conducting the experiment provides the agency the following results:

Table 10. Example Results for Experiment 6

Number of Files	Length of Time Allowed for File Transfers	Percentage of CPU Utilization	Percentage of I/O Utilization
60	60 minutes	15 %	25 %

(5) Summary of Results

For this experiment the following information is provided: the number of files transferred, the length of time during which the files are transferred, and the percentages of CPU and I/O utilization. The CPU and I/O utilization percentages are provided by the person conducting the experiment. These percentages can be used to determine the percentage of CPU and I/O remaining for other user applications, which will run simultaneously with the FTAM implementation. The user must also consider that increasing the number of applications will result in increased operating system overhead. This is because the operating system must grant and relinquish CPU and I/O control to the different applications.

The results recorded in this experiment do not represent actual data collected. They are listed for example purposes only. While the results simulate data obtained from one vendor's FTAM implementation, the experiment must be repeated for each vendor's FTAM implementation which is of interest to the user.

3.4.2. Mandatory Performance Requirements

This section recommends a procedure for eliminating candidate FTAM implementations which do not meet the mandatory performance requirements of the user. The user should create a list of performance measurements which must be obtainable by a candidate FTAM implementation, for that implementation to be acceptable. The user should be careful not to list as mandatory any specific performance measurements which are instead highly desirable, because this can unnecessarily restrict the list of candidate FTAM implementations. This list may be created by reviewing the experiments, noting which performance measurements are mandatory for the user. Once the list is created, the user should verify that the candidate FTAM implementations can obtain the performance measurements required by the user. A candidate FTAM implementation which cannot obtain the mandatory performance measurements of the user should be removed from the list of implementations at this point. As an example, if after reviewing the experiments, the user decides that the candidate FTAM implementations must be capable of receiving one hundred 1000-byte files in an hour (Experiment 5), then FTAM implementations which are not capable of receiving one hundred 1000-byte files in an hour are unacceptable to the user, and should no longer be evaluated.

3.4.3. Performance Evaluation

This section recommends a procedure for determining which of the candidate FTAM implementations, obtaining all required performance measurements, best meets the performance requirements of the user. First, the user must assign weights to each experiment based on how important the performance measurement determined by the experiment is to the user.

This procedure is defined in section 3.4.3.1. Second, the user must rate each of the candidate FTAM implementations based on the weights assigned by the user and the performance measurements obtained for the candidate FTAM implementations. This procedure is defined in section 3.4.3.2.

3.4.3.1. Weighing Performance

This section provides guidelines for assigning weights to each experiment based on how important to the user is the performance measurement determined by the experiment. The user should balance each of the experiments and determine how important each experiment is to the user. This step results in the assignment of a weight to each experiment.

The experiments must be balanced because the maximum score which may be received by one experiment has no relation to the maximum score which may be received by the other experiments. For example, one experiment may be scored subjectively on a scale of 0 to 10. Then this experiment can receive a maximum score of 10. Another experiment may be subjectively scored as 0 or 1. Then this experiment can receive a maximum score of 1. If these two example experiments are of equal importance to the user, then the weight of the second experiment must be 10 times as large as the weight of the first experiment, in order to balance the experiments.

The user must determine relative levels of importance of the experiments. The user first assigns a maximum score to each experiment to reflect its importance to the user. For example, the user may decide that an experiment which is very important to the user can receive a maximum score of 100 points, an experiment which is important to the user can receive a maximum score of 50 points, and an experiment which is not of any importance to the user can only receive a score of 0 points.

The user must then compute a weight for each experiment. The weight for each experiment is calculated by dividing the maximum score the user has determined that the experiment can receive by the maximum score of the experiment. As an example the user may consider Experiment 1 to be very important (100 points), Experiment 2 to be important (50 points), and Experiment 3 to be of no importance (0 points). If the maximum score of Experiment 1 is 10, then it is assigned a weight of 100/10 which equals 10. If the maximum score of Experiment 2 is 1, then it is assigned a weight of 50/1 which equals 50. If the maximum score of Experiment 3 is 10, then it is assigned a weight of 0/10 which equals 0.

3.4.3.2. Performance Evaluation Rating

This section recommends a procedure to rate the performance of the candidate FTAM implementations based on the weights assigned by the user and the performance measurements obtained for the candidate FTAM implementations. The performance measurements for a candidate FTAM implementation are obtained by performing the appropriate experiments. This performance rating procedure must be repeated for each candidate FTAM implementation. The procedure for determining a performance rating for each candidate FTAM implementation follows.

First, the user must score each experiment performed on a candidate FTAM implementation. There are two procedures recommended for scoring an experiment.

(1) The user can subjectively score the experiment based on the user's overall impression of the measurement obtained by the experiment for the candidate FTAM implementation. As an example, the user may decide to rate the experiment, performed on a candidate FTAM implementation, on a scale of 0-10 where a score of 10 indicates that the measurement obtained by the experiment for the candidate FTAM implementation meets all of the user's performance requirements for this experiment, a score of 5 indicates that the measurement obtained by the experiment for the candidate FTAM implementation meets some of the user's performance requirements for this experiment, and a score of 0 indicates that the measurement obtained by the experiment for the candidate FTAM implementation does not meet any of the user's performance requirements for this experiment.

(2) The user can subjectively score the experiment based on the user's overall impression of whether or not the measurement obtained by the candidate FTAM implementation is acceptable. If the measurement obtained from the experiment for the candidate FTAM implementation is acceptable, then it will receive a passing score (or 1); otherwise it will receive a failing score (or 0).

Second, the user determines the total performance rating, for a candidate FTAM implementation, by summing the weight of each experiment times the score for that experiment, for each of the experiments. For example, if there are experiments X, Y, and Z and experiment X has a weight of 5, experiment Y has a weight of 3, and experiment Z has a weight of 1, and an implementation has a score of 25 for experiment X, a score of 50 for experiment Y, and a score of 75 for experiment Z, then the performance rating for the implementation is [(5*25) + (3*50) + (1*75)] which equals 350. The equation for determining the total performance rating is specified in table 11. The candidate FTAM implementation with the highest score is likely to be the "best" implementation, with respect to performance, for that user. Note that these ratings are not absolute ratings; another user might rate the same candidate FTAM implementations differently based on a different set of requirements.

Table 11. Equation for Performance Rating FTAM Implementations

$$RP = \sum_{i=1}^{m} (E_i * WE_i)$$

$RP = Total\ Performance\ Rating$

$E_i = Rating\ of\ Experiment\ i$

$WE_i = Weight\ of\ Experiment\ i$

$m = Number\ of\ Experiments$

3.5. Guidelines for Rating Implementations

This section recommends a procedure for rating the candidate FTAM implementations based on the functional and performance evaluations. To arrive at a total score for each implementation, the user must weigh the functional evaluation totals and the performance evaluation totals, and compute the total score. As an example, let us assume that the user decides that the functional evaluation is twice as important as the performance evaluation. Then the total score for each FTAM implementation is 2*(functional evaluation total score) + (performance evaluation total score). The FTAM implementation which receives the highest total score is probably the best FTAM implementation for the user. The equation for determining the total rating for the candidate FTAM implementations is specified in table 12.

Table 12. Equation for Total Rating FTAM Implementations

$$R = (RF * WF) + (RP * WP)$$

R = *Total Rating*

RF = *Total Functional Rating*

WF = *Weight of Functional Rating*

RP = *Total Performance Rating*

WP = *Weight of Performance Rating*

3.6. Example Evaluation

This section provides an example evaluation of FTAM implementations using the previously described guidelines, and contains the following sections: Section 3.6.1 provides an example functional evaluation of FTAM implementations, section 3.6.2 provides an example performance evaluation of FTAM implementations, and section 3.6.3 provides an example rating of FTAM implementations based on the functional and performance evaluations.

In this example the user has selected four candidate FTAM implementations to evaluate, which are referenced as implementations A, B, C, and D. The user has obtained product literature, users manuals, technical specifications, and other available information from each of the vendors for the candidate FTAM implementations, in order to perform the evaluation.

3.6.1. Example Functional Evaluation

This section provides an example functional evaluation of the candidate FTAM implementations using the previously described guidelines. Appendix B provides, in a tabular format, a complete listing of the functionality potentially available in FTAM implementations, as described in section 3.3.1. This format is useful for performing the functional evaluation. This section contains the following sections: Section 3.6.1.1 provides an example of the procedure for eliminating candidate FTAM implementations which do not meet mandatory functional requirements of the user. Section 3.6.1.2 provides an example of the procedure recommended for determining which candidate FTAM implementation best meets the functional requirements of that user.

3.6.1.1. Example Mandatory Functional Requirements

This section provides an example of the procedure for eliminating candidate FTAM implementations which do not meet the mandatory functional requirements of the user. A list of the functions which must be included in the candidate FTAM implementations, for them to be acceptable, is created by reviewing the functions in each of the functional categories, noting which functions are mandatory for the user. In this example the user has determined that the candidate FTAM implementations must support the Connectionless Oriented Network Service over an 802.3 Local Area Network, the Connectionless Oriented Network Service over an X.25 Wide Area Network, and an FTAM role configuration which includes the initiator-sender, initiator-receiver, responder-sender, and responder-receiver roles. The example mandatory functions are referred to as functions 14.4, 14.5, 2.18. (See app. B for the tabular listing of functions.)

Next, the user verifies that each candidate FTAM implementation contains the mandatory functions. Any candidate FTAM implementation which does not contain all of the mandatory functions is removed from the list of candidate FTAM implementations at this point. Table 13 indicates which of the mandatory functions are contained in the candidate FTAM implementations.

Table 13. Example User Mandatory FTAM Functions

Function Number	Impl A	Impl B	Impl C	Impl D
2.18	Yes	Yes	Yes	No
14.4	Yes	Yes	Yes	Yes
14.5	Yes	Yes	Yes	Yes

Since implementation D does not support function 2.18 which is mandatory for the user, this implementation is removed from the list of candidate FTAM implementations.

3.6.1.2. Example Functional Evaluation

This section provides an example of the procedure for determining which of the candidate FTAM implementations best meets the functional requirements of that user. Four examples are provided to demonstrate the procedure for weighing the functions within a category and scoring a category, using each of the weighing options and their corresponding scoring algorithms. The scores for the remaining categories are provided without the details of how they were derived. One example is provided to demonstrate the procedure for weighing the categories and scoring the implementations. The example rating is derived by the following steps:

(1) The user must weigh each of the functions within a category using one of the options previously recommended. The user must score the category using the equation corresponding to the weighing option selected. This procedure is repeated for each category. An example rating of the functions within a category based on each of the options previously recommended follows.

An example is provided for weighing and scoring the functions within a category based on Option 1. Option 1 recommends determining a weight for each individual function in a category based on how important that function is to the user. The score of the category is the sum of the weight of each function which is present in the implementation. (See table 1 for the equation.) The FTAM Interfaces category is used in this example. Having the user interface implemented as a program (3.1), executing multiple FTAM commands during one association (3.2), and emulating an FTP user interface (3.8) are very important and are assigned a weight of 3. A command-driven interface (3.4) and having menu options for the command-driven interface (3.5) are important and are assigned a weight of 2. Having all passwords concealed (3.3), having user-defined macros (3.6), and executing in a batch or background mode (3.7) are less important and are assigned a weight of 1. All functions pertaining to a programmatic interface (3.9, 3.10, and 3.11) are not of any importance and are assigned a weight of 0. Table 14 contains the functions in the category, their weight for this example, their availability in each of the implementations, and a total score for this category for each implementation.

Table 14. Example Category Rating, Option 1

Function Number	Weight	Impl A	Impl B	Impl C
3.1	3	Yes	Yes	No
3.2	3	Yes	No	No
3.3	1	Yes	Yes	No
3.4	2	Yes	Yes	Yes
3.5	2	Yes	Yes	No
3.6	1	No	No	Yes
3.7	1	No	No	Yes
3.8	3	Yes	No	No
3.9	0	No	Yes	Yes
3.10	0	No	Yes	Yes
3.11	0	No	Yes	Yes
Total	16	14	9	4

In this example, out of a possible 16 points, implementation A scored 14, implementation B scored 9, and implementation C scored 4.

An example is provided for weighing and scoring the functions within a category based on Option 2. Option 2 recommends assigning each function in a category, which is of interest to the user, a weight of 1, and all other functions in the category a weight of 0. This option assumes that the user is either interested or not interested in a function, and that each of the functions of interest to the user are of equal importance. The score of the category is the sum of the number of functions, which are important to that user, and are present, in the implementation. (See table 1 for the equation.) The Transferring Files category is used in this example. The user's requirements (i.e., functions important to the user) for this example are: transfer multiple files with one command (4.1), transfer multiple files conforming to criteria (4.2), move a file (4.5), list known remote filestores (4.7), and view file transfer status information (4.12). The remaining functions are not important to the user. Table 15 contains the functions in the category, their weight for this example, their availability in each of the implementations, and a total score for this category for each implementation.

Table 15. Example Category Rating, Option 2

Function Number	Weight	Impl A	Impl B	Impl C
4.1	1	Yes	Yes	No
4.2	1	Yes	No	No
4.3	0	Yes	Yes	Yes
4.4	0	No	No	Yes
4.5	1	Yes	Yes	Yes
4.6	0	Yes	Yes	Yes
4.7	1	No	Yes	No
4.8	0	Yes	Yes	No
4.9	0	No	No	Yes
4.10	0	Yes	Yes	Yes
4.11	0	Yes	Yes	Yes
4.12	1	Yes	Yes	Yes
4.13	0	No	No	Yes
4.14	0	Yes	Yes	Yes
Total	5	4	4	2

In this example, out of a possible 5 points, implementation A scored 4, implementation B scored 4, and implementation C scored 2.

An example is provided for weighing and scoring the functions within a category based on Option 3. Option 3 recommends not assigning any weight to the functions in the category. The user subjectively scores the category based on the user's overall impression of how well that category is represented in the candidate FTAM implementation. (See table 1 for the equation.) The Deleting Files category is used in this example. The user's requirements for this example are to be able to delete multiple files with one command, delete multiple files conforming to criteria, and view file deletion status information. An implementation which meets all of the user's needs in this category will receive a score of 10, an implementation which meets half of the user's needs in this category will receive a score of 5, and an implementation which meets none of the user's needs in this category will receive a score of 0. Table 16 contains the subjective score for this category for each implementation.

Table 16. Example Category Rating, Option 3

	Max Score	Impl A	Impl B	Impl C
Subjective Score	10	10	7	5

In this example, out of a possible 10 points, implementation A scored 10, implementation

B scored 7, and implementation C scored 5.

An alternate example is provided for weighing and scoring the functions within a category based on Option 3. Option 3 recommends not assigning any weight to the functions in the category. The user subjectively scores the category based on the user's overall impression of whether or not the candidate FTAM implementation acceptably performs the functions in this category. If the candidate FTAM implementation acceptably performs the functions in this category it will receive a passing score (or 1); otherwise it will receive a failing score (or 0). (See table 1 for the equation.) The Viewing Files category is used in this example. The user's requirements for this example are to be able to view remote files and to be provided with viewing commands. Table 17 contains the subjective score for this category for each implementation.

Table 17. Example Category Rating, Option 3

	Max Score	Impl A	Impl B	Impl C
Subjective Score	1	0	1	0

In this example, implementations A and C failed and received a score of 0. Implementation B passed and received a score of 1.

(2) The user assigns a maximum rating to each category of functions, to indicate how important the category of functions is to the user. The user then computes the weight for each category by dividing the maximum rating that the category can receive by the maximum score of the functions within the category. The user determines the total functional rating, for a candidate FTAM implementation, by summing the weight of each category times the score for that category, for each of the categories. (See table 2 for the equation.) The user in this example has decided to assign ratings to the categories so that a category which is extremely important to the user can receive a maximum score of 400 points, a category which is very important to the user can receive a maximum score of 300 points, a category which is important to the user can receive a maximum score of 200 points, a category which is less important to the user can receive a maximum score of 100 points, and a category which is not of any importance to the user can only receive a score of 0 points. The user has decided that the optional FTAM functions, FTAM interfaces, and transferring files categories are extremely important, the deleting files and file directory operations categories are very important, the viewing files, default databases, and administrative functions categories are important, the debug capabilities, access control, underlying OSI layers, certification, on-line help facilities, system interface, and documentation categories are less important, and the hardware requirements, software requirements, and pragmatic constraints categories are not important. The weight for each category is calculated using the previously indicated algorithm. Table 18 contains the FTAM functional categories (Category), the maximum score the user has determined that the categories can receive for this example (Importance), the maximum score of the functions within the categories for this example (Raw Score), the computed

weight of the functional categories for this example (Weight), and the total functional rating for each of the implementations. Note, since category 1 describes mandatory functions, it is not listed in the evaluation.

Table 18. Example FTAM Implementation Functional Rating

Category	Importance	Raw Score	Weight	Impl A	Impl B	Impl C
2	400	10	40	400	320	280
3	400	16	25	350	225	100
4	400	5	80	320	320	160
5	300	10	30	300	210	150
6	200	1	200	0	200	0
7	300	10	30	240	300	150
8	200	10	20	140	180	200
9	100	1	100	100	100	100
10	0	0	0	0	0	0
11	200	10	20	160	160	200
12	100	10	10	60	80	100
13	100	20	5	75	100	25
14	100	10	10	90	50	50
15	100	1	100	100	100	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	100	5	20	100	80	60
19	0	0	0	0	0	0
Total	3000	-	-	2435	2275	1575

Candidate FTAM implementation A received the highest score in this example, and is likely to be the "best" implementation, functionally, for that user. Note that these ratings are not absolute ratings; another user might rate the same candidate FTAM implementations differently based on a different set of requirements.

3.6.2. Example Performance Evaluation

This section provides an example performance evaluation of the candidate FTAM implementations using the previously described guidelines. Appendix C provides, in a tabular format, a complete listing of the measurements potentially available in FTAM implementations, as described in section 3.4.1. This format is useful for performing the performance evaluation. This section contains the following sections: Section 3.6.2.1 provides an example of the procedure for eliminating candidate FTAM implementations which do not meet mandatory performance requirements of the user. Section 3.6.2.2 provides an example of the procedure recommended for determining which candidate FTAM implementation best meets the performance requirements of that user.

3.6.2.1. Example Mandatory Performance Requirements

This section provides an example of the procedure for eliminating candidate FTAM implementations which do not meet the mandatory performance requirements of the user. A list of the measurements which must be obtainable by the candidate FTAM implementations, for them to be acceptable, is created by reviewing each of the experiments, noting which measurements are mandatory for the user. In this example the user has determined that the candidate FTAM implementations must be able to transfer a minimum of 100 files in an hour. The example mandatory measurements are determined by experiment 5. (See app. C for the tabular listing of experiments.)

Next, the user verifies that each candidate FTAM implementation can obtain the mandatory measurements. Any candidate FTAM implementation which cannot obtain all of the mandatory measurements is removed from the list of candidate FTAM implementations at this point. Table 19 indicates the time required by each of the candidate FTAM implementations to transfer 100 files.

Table 19. Example User Mandatory FTAM Measurements

Experiment Number	Impl A	Impl B	Impl C
5	50 minutes 23 seconds	52 minutes 05 seconds	56 minutes 13 seconds

Since all of the implementations can transfer a minimum of 100 files per hour, no implementations are removed from the list of candidate FTAM implementations at this point.

3.6.2.2. Example Performance Evaluation

This section provides an example of the procedure for determining which of the candidate FTAM implementations best meets the performance requirements of that user. Two examples are provided to demonstrate the procedure for scoring the measurements obtained by an experiment, using the two scoring algorithms. The scores for the remaining measurements are provided without the details of how they were derived. One example is provided to demonstrate the procedure for weighing the measurements and scoring the implementations. The example rating is derived by the following steps:

(1) The user must score each experiment performed on a candidate FTAM implementation. An example of each of the two procedures recommended for scoring an experiment follows.

In this first example, the user subjectively scores the experiment based on the user's overall impression of the measurement obtained by the experiment for the candidate FTAM implementation. Experiment 5 is used in this example. The user's requirement for this example is to be able to transfer a minimum of 100 files per hour. A score of 10 indicates that the measurement obtained by the experiment for the implementation meets all of the user's performance requirements for this experiment, a score of 5 indicates that the measurement obtained by the experiment for the implementation meets some of the user's performance requirements for this experiment, and a score of 0 indicates that the measurement obtained by the experiment for the implementation does not meet any of the user's performance requirements for this experiment. Table 20 contains the subjective score for this experiment for each implementation.

Table 20. Example Experiment Rating

	Max Score	Impl A	Impl B	Impl C
Subjective Score	10	10	8	6

In this first example, out of a possible 10 points, implementation A scored 10, implementation B scored 8, and implementation C scored 6.

In this second example, the user subjectively scores the experiment based on the user's overall impression of whether or not the measurement obtained by the candidate FTAM implementation is acceptable. If the measurement obtained from the experiment for the candidate FTAM implementation is acceptable, then it will receive a passing score (or 1); otherwise it will receive a failing score (or 0). Experiment 1 is used in this example. The user's requirement for this example is to be able to transfer a file in less than 1 minute. Table 21 contains the subjective score for this experiment for each implementation.

Table 21. Example Experiment Rating

	Max Score	Impl A	Impl B	Impl C
Subjective Score	1	1	1	1

In this example, implementations A, B, and C passed and received a score of 1.

(2) The user assigns a maximum rating to each experiment, to indicate how important the measurement obtained by the experiment is to the user. The user then computes the weight for each experiment by dividing the maximum rating that the experiment can receive by the maximum score of the measurement obtained by the experiment. The user determines the total performance rating, for a candidate FTAM implementation, by summing the weight of each experiment times the score for that experiment, for each of the experiments. (See table 11 for the equation.) The user in this example has decided to assign ratings to the experiments so that an experiment which is very important to the user can receive a maximum score of 100 points, an experiment which is important to the user can receive a maximum score of 50 points, and an experiment which is not of any importance to the user can only receive a score of 0 points. The user has decided that Experiment 5 is very important, Experiments 1 - 4 are important, and Experiment 6

is not of any importance. The weight for each experiment is calculated using the previously indicated algorithm. Table 22 contains the FTAM experiment number (Experiment), the maximum score the user has determined that the experiment can receive for this example (Importance), the maximum score of the measurement obtained by the experiment for this example (Raw Score), the computed weight of the experiments for this example (Weight), and the total performance rating for each of the implementations.

Table 22. Example FTAM Implementation Performance Rating

Experiment	Importance	Raw Score	Weight	Impl A	Impl B	Impl C
1	50	1	50	50	50	50
2	50	1	50	50	50	50
3	50	1	50	50	50	50
4	50	1	50	50	50	50
5	100	10	10	100	80	60
6	0	0	0	0	0	0
Total	300	-	-	300	280	260

Candidate FTAM implementation A received the highest score in this example, and is likely to be the "best" implementation, for performance, for that user. Note that these ratings are not absolute ratings; another user might rate the same candidate FTAM implementations differently based on a different set of requirements.

3.6.3. Example Rating

This section provides an example of the procedure recommended for rating the candidate FTAM implementations based on the functional and performance evaluations. To arrive at a total score for each implementation, the user must weigh the functional evaluation totals and the performance evaluation totals, and compute the total score. (See table 12 for the equation.) The user in this example has decided the functional evaluation is twice as important as the performance evaluation, and therefore assigns the functional evaluation weight as 2 and the performance evaluation weight as 1. Table 23 contains the functional and performance weights, functional and performance scores for each of the implementations, and the total rating for each of the implementations.

Table 23. Example FTAM Implementation Total Rating

	Weight	Impl A	Impl B	Impl C
Functional	2	2435	2275	1575
Performance	1	300	280	260
Total	-	5170	4830	3410

Candidate FTAM implementation A received the highest score in this example, and is probably the "best" FTAM implementation for the user.

3.7. Other Guidelines

This section describes other factors to consider when evaluating candidate FTAM implementations. The guidelines defined in this section are not as concrete as the ones in the previous sections, and therefore, are not in the functional or performance evaluation sections. They are, however, factors to be considered when evaluating implementations.

This section contains four major topics. The first topic, effectiveness, is relevant to the FTAM implementation. The second two topics, commitment and support, are relevant to the vendor. The final topic is cost.

One factor to consider when evaluating an FTAM implementation is the effectiveness of the functionality provided by the implementation. For example, a user may be provided with a program to assist with installing the implementation; however, the installation procedure may be very difficult and time-consuming despite the installation program. Debugging functions may exist, but may be inadequate to solve problems easily. Finally, the documentation provided with an implementation may not be well organized, or may be difficult to understand.

To appreciate the effectiveness of an FTAM implementation, the user has several options. The user can request a copy of the FTAM documentation from the vendor. By examining the documentation in advance, the user can better determine its adequacy and understandability. The user may also be able to determine how easy the implementation is to install, configure, debug, and use. Another option is for the user to request a demonstration of the FTAM implementation. A demonstration will provide an overall view of the implementation, especially concerning its "user friendliness."

Some evaluation factors relate to the vendor. The user should consider the company's commitment to OSI, and if the personal contacts (i.e., sales and service representatives) are well informed. The user may consider whether the company marketing the product also developed the product. Other noteworthy evaluation factors are the company's ability to service their product, the company's policy regarding product upgrades, and customer service issues. Customer service issues include: software support, whether the support is local or out of town, and maintenance agreements. The user should ask the vendor about the type and extent of customer support that is available.

The final topic concerning the evaluation of an FTAM implementation is the cost of the implementation. This includes hardware costs (e.g., computer systems, LAN cards, WAN cards), software costs, and maintenance costs such as maintenance contracts. The budget of the user will determine the importance of cost as an evaluation factor.

4. Conclusion

As stated in the Introduction, the intent of this document is to advance the goals of the GOSIP by providing guidelines for evaluating FTAM implementations. These guidelines can assist a user in determining which implementation, among several candidates, will best meet the functional and performance requirements of the user.

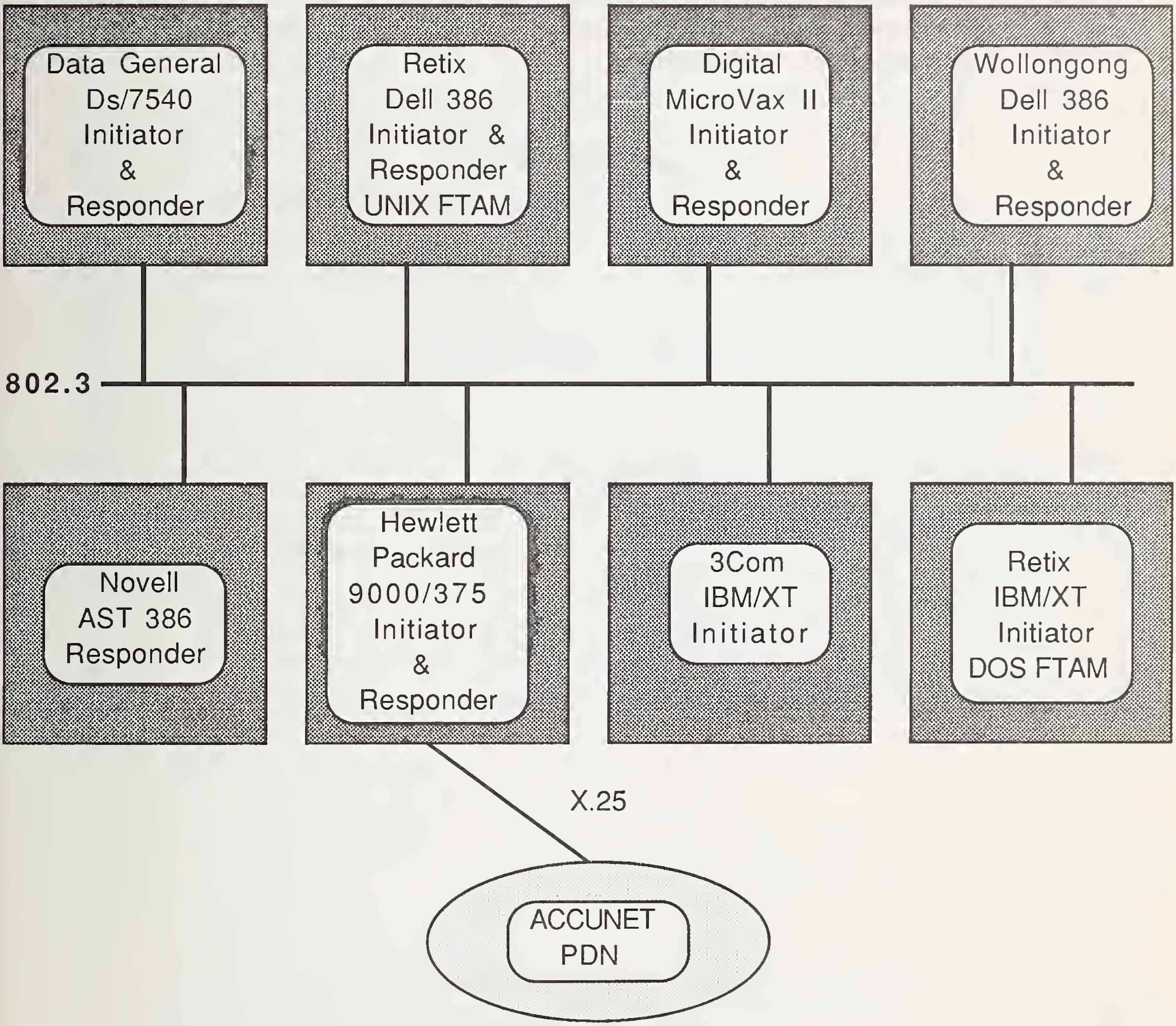
The guidelines for evaluating FTAM implementations contain a procedure for rating FTAM implementations based on the user's requirements, and a list of factors which can not be rated by the user, but should be taken into consideration when selecting an FTAM implementation.

The procedure for rating FTAM implementations includes evaluating the functional and performance capabilities of FTAM implementations based on the user's requirements, and rating FTAM implementations based on their functional and performance evaluations. The functional evaluation is important because FTAM implementations may vary in the provision of non-standard functions, as well as functions categorized as optional in the FTAM standard. The performance evaluation is important because FTAM implementations may vary in their level of performance. The overall rating of an FTAM implementation is a combination of its functional and performance evaluations.

It is recommended that users, in order to procure the "best" FTAM implementation for their needs, follow these guidelines when selecting an FTAM implementation.

APPENDIX A. Lab Configuration

An FTAM laboratory, containing a representative sample of the FTAM implementations currently available, was established. Figure 20 depicts the configuration of the FTAM Laboratory used in this project.



Data General Ds/7540 Initiator & Responder
Retix Dell 386 Initiator & Responder UNIX FTAM
Digital MicroVax II Initiator & Responder
Wollongong Dell 386 Initiator & Responder
802.3
Novell AST 386 Responder
Hewlett Packard 9000/375 Initiator & Responder
3Com IBM/XT Initiator
Retix IBM/XT Initiator DOS FTAM
X.25
ACCUNET PDN

Figure 20. NIST Network Applications Lab.

APPENDIX B. Tables of Functions

This appendix contains a listing of the FTAM functions described in these guidelines. The functions are presented here in tabular form. Each table consists of a title and a list of entries. The title corresponds to a category of functions detailed in section 3.3.1. The list of entries correspond to the set of functions contained in that category. Each function listed is preceded by two numbers separated by a period. The numbers to the left of the periods match a category subsection of 3.3.1. The numbers to the right of the periods represent a numerical listing of functions within a category.

Mandatory FTAM Functions
1.1 Limited-Purpose FTAM System
1.2 Kernel Attribute Group
1.3 Called Application Title
1.4 FTAM Role (Note that only one of the described roles - Initiator-Receiver, Initiator-Sender, Responder-Receiver, or Responder-Sender - must be supported.)
1.5 FTAM-3 Document Type
1.6 FTAM-1 Document Type
1.7 Simple File Transfer Implementation Profile (T1)
1.8 Management Implementation Profile (M1)
1.9 Override Values
1.10 Read and/or Replace Requested Access
1.11 Read Attribute Requested Access
1.12 Change Attribute Requested Access
1.13 Delete File Requested Access

Optional FTAM Functions	
2.1	Full-Purpose FTAM System
2.2	Access Passwords
2.3	Account
2.4	Storage Attribute Group
	Functions 2.5 through 2.14 are optional attributes within the storage attribute group.
2.5	Date and Time of Creation Attribute
2.6	Date and Time of Last Modification Attribute
2.7	Date and Time of Last Read Access Attribute
2.8	Date and Time of Last Attribute Modification Attribute
2.9	Identity of Creator Attribute
2.10	Identity of Last Modifier Attribute
2.11	Identity of Last Reader Attribute
2.12	Identity of Last Attribute Modifier Attribute
2.13	Future Filesize Attribute
2.14	Storage Account Attribute
2.15	Security Attribute Group
	Function 2.16 is an optional attribute within the security attribute group.
2.16	Legal Qualifications Attribute
2.17	Charging
2.18	Concurrency Control
2.19	FTAM Roles
2.20	FTAM-2 Document Type (required for Full-Purpose System)
2.21	NBS-6 Document Type
2.22	NBS-7 Document Type
2.23	NBS-8 Document Type
2.24	NBS-9 Document Type
2.25	Create Password
2.26	Diagnostics
2.27	Filestore Password
2.28	Positional File Transfer Implementation Profile (required for Full-Purpose System)
2.29	Full File Transfer Implementation Profile (required for Full-Purpose System)
2.30	Simple File Access Implementation Profile
2.31	Full File Access Implementation Profile

Optional FTAM Functions (continued)
2.32 Initiator Identity
2.33 Override
2.34 Recovery
2.35 Insert Requested Access
2.36 Erase Requested Access
2.37 Restart Data Transfer

FTAM Interfaces
3.1 Method of Implementation (i.e., program vs. operating system extension)
3.2 Execute Multiple FTAM Commands During One FTAM Association
3.3 Passwords Concealed
3.4 Type of Interface (i.e., window-driven vs. command-driven)
3.5 Menu Options for Command-Driven Interface
3.6 User-Defined Macros
3.7 Execute in Batch or Background Mode
3.8 Emulate FTP User Interface
3.9 High Level API
3.10 Low Level API
3.11 POSIX Conformant API

Transferring Files
4.1 Transfer Multiple Files With One Command
4.2 Transfer Multiple Files Conforming to Criteria
4.3 User Confirmation Prior to File Transfer
4.4 Allow Wildcards
4.5 Move a File
4.6 Override Options Additional to Those Defined in FTAM Standard
4.7 List Known Remote Filestores
4.8 Transfer File to Different Local Directory
4.9 Simplify Document Type
4.10 Transfer File Between Two Remote Filestores
4.11 Rename File When Transferring Between Two Remote Filestores
4.12 View File Transfer Status Information
4.13 Specify Status Information to be Displayed
4.14 Write Status Information to a File

Deleting Files
5.1 Delete Multiple Files With One Command
5.2 Delete Multiple Files Conforming to Criteria
5.3 User Confirmation Prior to Deleting File
5.4 Delete Files on Local Filestore
5.5 View File Deletion Status Information
5.6 Write Status Information to a File

Viewing Files
6.1 View Files
6.2 Viewing Commands
6.3 View Local Files

File Directory Operations
7.1 Type of File Listing (i.e., long vs. short)
7.2 Listing Files Conforming to Criteria
7.3 File Listing Summary Information
7.4 File Listing Format Control
7.5 Listing Local Files
7.6 Change Remote Current Working File Directory

Default Databases
8.1 Default Configuration Database
8.2 Filestore Database
8.3 Display Database Entries
8.4 Modify Database Entries

On-Line Help Facilities
9.1 Provide On-Line Help Facility

System Interface
10.1 Execute Operating System Commands
10.2 Printing Resources

Administrative Functions	
11.1	On-Line Training Program for Installing and Configuring FTAM Implementation
11.2	Automated Installation Procedure
11.3	Installation Verification Facility
11.4	Implementation Start-Up at System Start-Up
11.5	Manual Start and Stop
11.6	Run As Initiator or Responder Only
11.7	FTAM Statistics
11.8	Separate Statistics for Initiator and Responder
11.9	Modify FTAM Parameters
11.10	Utility Program for Administrative Tasks
11.11	Help Facility for Utility Program
11.12	Backup/Restore FTAM Implementation
11.13	Restore Implementation to Different Machine

Debug Capabilities	
12.1	Log File
12.2	Display Errors on Console
12.3	FTAM Tracing Utility
12.3	Underlying OSI Layer Tracing Utility

Access Control	
13.1	Deny Access Based on Network Address
13.2	Deny Access Based on Initiator Identity Value
13.3	Limit Access Based on Initiator Identity Value
13.4	Anonymous FTAM Account

Underlying OSI Layers	
14.1	Presentation Services
14.2	Session Services
14.3	Transport Class 4 Over CLNS
14.4	Transport Class 4 Over CONS
14.5	Transport Class 0 Over CONS
14.6	X.25 Version
14.7	Pre-Defined X.25 Parameter Values
14.8	Modify X.25 Options
14.9	Lower Layer Statistics
14.10	Optimization Facility
14.11	Separate Log File for Each Layer

Conformance and Interoperability Testing and Registration	
15.1	Passed Government-Approved Conformance Test Procedure
15.2	Passed Government-Approved Interoperability Test Procedure

Hardware Requirements	
16.1	CPU
16.2	Disk Space
16.3	Memory
16.4	External Devices

Software Requirements	
17.1	Number of Software Components
17.2	Underlying OSI Software Installation
17.3	Operating System and Version

Documentation	
18.1	Guide for Installing
18.2	Guide for Using
18.3	Guide for Administrating
18.4	Guide for Troubleshooting
18.5	Guide for Quick Reference

Pragmatic Constraints	
19.1	Simultaneous FTAM Associations
19.2	Number of Users
19.3	Number of Filestores
19.4	Filenames

APPENDIX C. Tables of Experiments

This appendix contains a listing of the FTAM experiments described in these guidelines. The experiments are presented here in tabular form. Each table consists of a title, which corresponds to an experiment described in section 3.4.1, a purpose, and a list of experiment inputs and outputs.

Experiment 1
Purpose: To measure the optimum file transfer time interval, and to obtain a base measurement against which message transfer times intervals of other experiments can be compared.
Inputs: 1. The size of the files to be transferred. 2. The document type of the files to be transferred.
Outputs: 1. The file number for each file transferred. 2. The beginning file transfer time for each file transferred. 3. The ending file transfer time for each file transferred. 4. The transfer time interval for each file transferred. 5. The minimum file transfer time interval. 6. The maximum file transfer time interval. 7. The average file transfer time interval.

Experiment 2

Purpose:

To measure the effect of varying the filesize on the file transfer time interval.

Inputs:

1. The size of the files to be transferred.
2. The document type of the files to be transferred.

Outputs:

1. The file number for each file transferred.
2. The beginning file transfer time for each file transferred.
3. The ending file transfer time for each file transferred.
4. The transfer time interval for each file transferred.
5. The minimum file transfer time interval.
6. The maximum file transfer time interval.
7. The average file transfer time interval.
8. The difference between the average file transfer time intervals of Experiment 2 and Experiment 1.

Experiment 3

Purpose:

To measure the effect of varying the document type on the file transfer time interval.

Inputs:

1. The size of the files to be transferred.
2. The document type of the files to be transferred.

Outputs:

1. The file number for each file transferred.
2. The beginning file transfer time for each file transferred.
3. The ending file transfer time for each file transferred.
4. The transfer time interval for each file transferred.
5. The minimum file transfer time interval.
6. The maximum file transfer time interval.
7. The average file transfer time interval.
8. The difference between the average file transfer time intervals of Experiment 3 and Experiment 1.

Experiment 4

Purpose:

To measure the effect of estimated file transfer usage on the file transfer time interval.

Inputs:

1. The size of the files to be transferred.
2. The document type of the files to be transferred.
3. The file transfer timetable.

Outputs:

1. The file number for each file transferred.
2. The beginning file transfer time for each file transferred.
3. The ending file transfer time for each file transferred.
4. The transfer time interval for each file transferred.
5. The minimum file transfer time interval.
6. The maximum file transfer time interval.
7. The average file transfer time interval.
8. The difference between the average file transfer time intervals of Experiment 4 and Experiment 1.

Experiment 5

Purpose:

To measure file transfer capacity.

Inputs:

1. The size of the files to be transferred.
2. The document type of the files to be transferred.
3. The number of files to be transferred.
4. The file transfer period.

Outputs:

1. The file number for each file transferred.
2. The beginning file transfer time for the first file.
3. The ending file transfer time for the last file.
4. The total file transfer time interval.

Experiment 6

Purpose:

To measure CPU and I/O utilization.

Inputs:

1. The size of the files to be transferred.
2. The document type of the files to be transferred.
3. The number of files to be transferred.
4. The length of time during which the files are to be transferred.
5. The file transfer period.

Outputs:

1. The number of files transferred.
2. The length of time during which the files were to be transferred.
3. The percentage of CPU utilization.
4. The percentage of I/O utilization.

APPENDIX D. Abbreviations

This appendix defines the abbreviations used in this document.

API	Application Program Interface
CCITT	Consultative Committee on International Telegraphy and Telephony
CLNS	Connectionless Network Service
CONS	Connection-Oriented Network Service
DU	Data Unit
FADU	File Access Data Unit
FIPS	Federal Information Processing Standard
FTAM	File Transfer, Access and Management
FTP	File Transfer Protocol
GOSIP	Government OSI Profile
ISDN	Integrated Services Data Network
ISO	International Organization for Standardization
LAN	Local Area Network
MHS	Message Handling System
NIST	National Institute of Standards and Technology
OIW	OSI Implementors Workshop
OSI	Open Systems Interconnection
POSIX	Portable Operating System Interface for Computing Environments
VT	Virtual Terminal
WAN	Wide Area Network

APPENDIX E. Glossary

This appendix provides a glossary of FTAM terms.

access context – The specification of an algorithm defining a subset of the structuring information and user information in a file's contents, when reading the file for transfer or access.

accounting regime – The period during which a particular set of accounting information applies.

activity attributes – The attributes describing the activity of using the file service. The attributes are local to one FTAM regime (or any regime nested within it).

arc – A direct link between two nodes.

arc length – A positive integer expressing the difference in levels between a child node and its parent node.

attribute – A piece of information stating a property of something, taking one of a set of defined values, each value having a defined meaning.

child (of a node) – A node at which an outbound arc of the node concerned terminates.

concatenation (of documents) – The combination of two documents to form a single document.

constraint set – A set of restrictions and refinements of a general file model which specifies a less general model tailored to the needs of a particular class of applications.

data element – The smallest piece of data whose identity is necessarily preserved when transferred by the Presentation Service. A data element can convey file contents information, file structuring information, or protocol control information.

data unit – The smallest unit of a file's contents which the filestore action can manipulate. Each data unit is associated with a node of the file access structure. A data unit is a set of data elements.

document – A collection of information with known abstract syntaxes and partially known semantics, and a known set of possible transfer syntaxes.

document type – The specification of a class of documents, which states their necessary semantics, abstract syntaxes, transfer syntaxes, and dynamics.

dynamics (of a document) – The concatenation and simplification properties of a document.

empty file – A file whose contents consist of only a root node with no associated data unit, and no node name.

external file service – File transfer, access, and management as seen by the file service user.

file access – The inspection, modification, replacement, or erasure of part of a file's contents.

file access data unit – A unit of the file access structure on which the actions of transfer, deletion, extension, replacement, or insertion can be performed. A file access data unit contains zero or more data units.

file access structure – The data structure of a file that relates the file access data units, allowing their identification, description, and manipulation.

file attributes – The name and other identifiable properties of a file.

file contents – The data units, node names and structuring information contained in the file, which may be manipulated during the file open regime; the file attributes do not form part of the file's contents.

file management – The creation and deletion of files, and the inspection and manipulation of the file attributes.

file model – A model of the access structure of a file's contents.

file service user – That portion of the application entity which conceptually invokes the FTAM service.

filestore action – One of the actions specified as part of the definition of the virtual filestore.

file transfer – A function which moves a part or the whole of a file's contents between open systems.

flat (constraint set) – A constraint set which, when applied to the general hierarchical file model, generates an access structure that consists of two levels, at the levels zero and one, and that may have data units at only the leaf nodes and has no data unit at the root node.

general hierarchical file model – A model in which the file access data units are organized in a hierarchical tree.

hierarchical (constraint set) – A constraint set which, when applied to the general hierarchical file model, generates an access structure which is still hierarchical, but in which the form of the node descriptions and data unit is restricted.

hierarchical file model – A model of the internal structure of a file which takes the form of a tree of nameable file access data units.

initiator – That file service user which requests FTAM regime establishment.

internal file service – The service used by the file error recovery protocol machine to transmit both file error recovery protocol information and normal file control information.

leaf – A node of a tree that has no outbound arcs.

level (of a node) – The sum of all the arc lengths from the root to the node concerned.

node – The elementary component from which a tree is built up.

parent (of a node) – The node from which the inbound arc of the node concerned originates.

path – A sequence of directed arcs which links one node to another node.

phase – The period of time in which protocol exchanges have a particular purpose, such as establishing or releasing an application context; for each phase a set of valid messages is defined in terms of state transitions.

real file – The named collection of information and its attributes which reside in a real system and to which the references to virtual files are mapped.

real filestore – An organized collection of files, including their attributes and names, which reside in a real system and to which the virtual file references are mapped.

receiver – The entity which receives part or all of the file's contents during the file data transfer regime.

regime – The period during which the entity is in a subset of its possible states for which particular actions are permitted.

responder – That file service user which accepts an FTAM regime establishment requested by the initiator.

root – The unique node of a tree that has no inbound arcs; it is at level 0.

sender – The entity which sends part or all of the file's contents during the file data transfer regime.

service element – A unit of standardization specifying a complete group of functions.

service primitive – The smallest defined interaction between the user and the provider of a communication service.

simplification (of a document) – The process of deriving one document from another of a different type by discarding structuring information.

subtree – A part of a tree comprising an arbitrary node as the subtree root node and all other nodes can be reached by a path from the subtree root node.

tree – A connected structure in which each node is linked to other nodes by directed arcs in such a way that one node has no inbound arcs and all other nodes have exactly one inbound arc.

unstructured (constraint set) – A constraint set which, when applied to the general hierarchical file model, generates an access structure that consists only of the root node with one data unit.

virtual file – An unambiguously named collection of structured information having a common set of attributes.

virtual filestore – An abstract model for describing files and filestores, and possibly the actions on them.

REFERENCES

The documents referenced in this project include:

[1] File Transfer, Access and Management International Standard, ISO 8571.

[2] Stable Implementation Agreements for Open Systems Interconnection Protocols Version 3 Edition 1, NIST Spec. Publ. 500-177; 1989 December.

[3] Boland, T., Government Open Systems Interconnection Profile Users' Guide, NIST Spec. Publ. 500-163; 1989 August.

[4] Government Open Systems Interconnection Profile (GOSIP), Version 1. FIPS 146.

[5] Government Open Systems Interconnection Profile (GOSIP), Version 2. FIPS 146-1.

[6] NIST FTAM Interoperability Tests, NISTIR 4435.

[7] GOSIP Conformance and Interoperation Testing and Registration, NISTIR 4594.

[8] COMSAT Laboratories, 1988 COMSAT/NIST OSI Standard Protocols Experiments.

[9] Data General, Installing, Starting, and Stopping AOS/VS II.

[10] Data General, AOS/VS and AOS/VS II.

[11] Data General, Managing XTS II With DG/OpenNMS.

[12] Data General, Using DG/FTAM.

[13] Data General, Managing DG/FTAM With DG/OpenNMS.

[14] Data General, Managing Your Network With DG/OpenNMS.

[15] Data General, Configuring Your Network With DG/OpenNMS.

[16] Retix, Network Directory Compiler Software Reference Manual.

[17] Digital, VOTS 2.0 Introduction.

[18] Digital, VOTS 2.0 Installation Guide.

[19] Digital, VOTS 2.0 Management Guide.

[20] Digital, VOTS 2.0 User's Guide.

[21] Digital, VOTS 2.0 Program Guide.

[22] Digital, VOTS 2.0 Problem Solving Guide.

[23] Digital, VAX/FTAM Installation Guide.

[24] Digital, VAX/FTAM User's Guide.

[25] Digital, VAX/FTAM Management Guide.

[26] Hewlett Packard, FTAM 9000 Reference Manual.

[27] Hewlett Packard, FTAM 9000 User's Guide.

[28] Hewlett Packard, FTAM 9000 Programmers Guide.

[29] Novell, Network 386 OSI FTAM Server's Supervisors Guide.

[30] Retix, LT-610 Administrator Guide.

[31] Retix, LT-610 Programmer Guide.

[32] Retix, LT-605 Programmer Guide.

[33] Retix, LT-605 Administrator Guide.

[34] Retix, FT-820 Administrator Guide.

[35] Retix, FT-820 User Guide.

[36] Retix, FT-810 Administrator Guide.

[37] Retix, FT-810 User Guide.

[38] Wollongong, Introduction to OSI.

[39] Wollongong, WIN/LLS Administration Guide.

[40] Wollongong, Upper Layer Services WIN/ULS Reference Manual.

[41] Wollongong, Upper Layer Services WIN/ULS Administration Guide.

[42] Wollongong, Upper Layer Services WIN/ULS Programming Guide.

[43] Wollongong, WIN/FTAM FTAM-TP Application Gateway.

[44] Wollongong, WIN/FTAM FTAM User's Guide.

[45] 3COM, OSI/ES With Demand Protocol Architecture.

[46] 3COM, OSI/ES Internet BIOS Guide.

NIST-114A
(REV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NIST/SP-500/196

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
October 1991

4. TITLE AND SUBTITLE
Guidelines for the Evaluation of File Transfer, Access and Management Implememtations

5. AUTHOR(S)
Paul Markovitz, Steve Trus, Curtis Royster

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)
U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)
U. S. Army Information Systems Command, Ft. Huachuca, AZ, 85613-5300; U.S. Air Force Communications Command, Scott Air Force Base, IL, 62225-6001;
Internal Revenue Service, 1111 Constitution Ave., NW, Wash., DC 20224

10. SUPPLEMENTARY NOTES

□ DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

This document advances the goals of the Government Open Systems Interconnection Profile (GOSIP) by providing guidelines for evaluating File Transfer, Access and Management (FTAM) implementations. These guidelines can assist a user in determining which implementation, among several candidates, will best meet the functional and performance requirements of that user. Specifically, this document contains: (1) guidelines for evaluating the functional specifications of FTAM implementations, (2) guidelines for measuring the performance of FTAM implementations, and (3) guidelines for matching the functional and performance specifications of an FTAM implementation to the functional and performance requirements of the user.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

File Transfer, Access and Management; functional evaluation; Government Open Systems Interconnection Profile; initiator; Open Systems Interconnection; performance evaluation; responder

13. AVAILABILITY

☒ UNLIMITED

□ FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).

☒ ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.

☒ ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
107

15. PRICE

ELECTRONIC FORM

✿U.S. GOVERNMENT PRINTING OFFICE:1991 -311 -944/52627

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SYSTEMS TECHNOLOGY

Superintendent of Documents
Government Printing Office
Washington, DC 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Institute of Standards and Technology Special Publication 500-.

Name ____

Company ____

Address ____

City ____________________ State ________________ Zip Code ______________

(Notification key N-503)

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences.

Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published bi-monthly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW., Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NIST research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NIST publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency Reports (NISTIR)—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Institute of Standards and Technology
Gaithersburg, MD 20899

Official Business
Penalty for Private Use $300