

# Integration Test Plan Document for ANFS

Advisor:

Yuanfang Cai

Technical Stakeholder:

Bill Mongan

By:

Lee Trager

Arkadiy Sherman

Matt Wyatt

Harry Anuszewski

Mike Dalton

# Revision History

| Name   | Date   | Reason for Changes | Version |
|--|--------|--------------------|---------|
| Lee Trager, Harry Anuszewski, Mike Dalton, Arkadiy Sherman, Matt Wyatt | 2-8-11 | Initial version    | 1.0     |

# Table of Contents

## [1. Introduction](#)

### [1.1 Purpose](#)

### [1.2 References](#)

### [1.3 Glossary](#)

## [2 Test Approach and Constraints](#)

### [2.1 Test Objectives](#)

### [2.2 Test Structure](#)

### [2.3 Test Constraints](#)

## [3 Test Assumptions and Exclusions](#)

### [3.1 Test Assumptions](#)

### [3.2 Test Exclusions](#)

## [4 Entry and Exit Criteria](#)

### [4.1 Entry Criteria](#)

### [4.2 Exit Criteria](#)

## [5 Testing Participants](#)

### [5.1 Roles and Responsibilities](#)

### [5.2 Training Requirements](#)

### [5.3 Problem Reporting](#)

### [5.4 Progress Reporting](#)

## [6 Test Cases](#)

### [6.1 Redundancy Test Cases](#)

#### [6.1.1 Adding a Server](#)

##### [6.1.1.1 Add server command](#)

##### [6.1.1.2 Checking Data While No Data is Being Manipulated](#)

##### [6.1.1.3 Checking Data While Data is Being Manipulated](#)

##### [6.1.1.4 Notify all clients and servers](#)

#### [6.1.2 Intentional Removal of a Server](#)

##### [6.1.2.1 Server Removal Command](#)

##### [6.1.2.2 Client and Server Notification](#)

#### [6.1.3 Unintentional Removal of a Server \(Server Failure\)](#)

##### [6.1.3.1 Client discovery of a failed server](#)

##### [6.1.3.2 Server discovery of a failed server](#)

##### [6.1.3.3 Server notified by client of a failed server](#)

##### [6.1.3.4 Server notified by another server of a failed server](#)

##### [6.1.3.5 Client notified by server of a failed server](#)

#### [6.1.4 Reading or writing data to a server](#)

##### [6.1.4.1 Reading Data](#)

##### [6.1.4.2 Writing Data](#)

### [6.2 Encryption](#)

#### [6.2.1 Encryption Enabled](#)

[6.2.2 Generate RSA Key Pair with OpenSSL](#)

[6.2.3 Exchange RSA Public Key](#)

[6.2.4 Encrypt data using AES from Crypto API](#)

[6.2.5 Decrypt data using AES from Crypto API](#)

### [6.3 Compression](#)

[6.3.1 Compression Enabled](#)

[6.3.2 Compression Disabled](#)

[6.3.3 Auto Compression Continue Compression](#)

[6.3.4 Auto Compression Stop Compression](#)

[6.3.5 Server Does not support Data Compression](#)

[6.3.6 Data Stream Inflated Client](#)

[6.3.7 Data Stream Inflated Server](#)

### [6.4 Server Operations](#)

[6.4.1 Client Accesses Servers](#)

[6.4.2 Server Accesses Servers](#)

[6.4.3 List Servers](#)

[A full list of available servers is not printed with three columns for each server.](#)

[6.4.4 Notify Servers](#)

# 1. Introduction

## 1.1 Purpose

This document will provide the plan for the Integration Testing phase of the design of ANFS. It will cover the preconditions, actions, expected results, and failure conditions of the test we will execute to ensure complete coverage of the functional requirements and their implementation outlined in the requirements and design documents.

## 1.2 References

The Requirements Document for the ANFS project.

The Design Document for the ANFS project.

## 1.3 Glossary

AES (Advanced Encryption Standard) - a popular private-key cryptography algorithm

Administrator - The user on a system who has the ability to configure any part of the system

Client - A machine which is connected to an ANFS server. All files are read and written to the server

Compression - The process of encoding information using less bits than the original information would use.

Encryption - The process of using an algorithm to make information unreadable by anyone except those possessing special knowledge.

Inflation - Returning compressed data to its original format and size

Mirrored Servers - Any other server connected and synchronized with the current server.

Network File System - File system that allows access to files from multiple hosts shared via a computer network.

OpenSSL - Toolkit that implements the SSL(Secure Sockets Layer) and TLS (Transport Layer Security) protocols. We use this library for its Cryptographic algorithms.

RSA (Rivest, Shamir and Adleman) - a popular public-key cryptography algorithm

Redundancy - Repetition of information in order to avoid errors, computer failure, and slow connections.

Server - The machine which has all of the data. It also keeps track of which servers are available and communicates with clients.

Wireshark - Application that analyzes packets/data as they are sent over your network.

XDR Datastream - A data type standard for description and encoding of data (RFC 4506).

## **2 Test Approach and Constraints**

The following section of the document will cover the testing approach and any tools and techniques necessary during the Integration Testing phase of development for the ANFS project. This section will also cover any constraints due to hardware or software that will affect the testing phase.

### **2.1 Test Objectives**

The Integration Test Plan will outline the procedures that the client and administrator must complete to fully evaluate the functionality of ANFS as outlined in the Design Document.

### **2.2 Test Structure**

All tests will be performed on properly configured machines running the latest version of ANFS. Any test which involves redundancy will include at least two servers. A client machine will be needed for some of the tests.

The Integration Test Plan will include Module Testing and will verify that the three modules of ANFS, Redundancy, Encryption, and Compression, function according to the architecture described in the Design Document. ANFS modules will be designed and built to function independently of each other, and upon successful testing of each module, the Integration Test phase will cover the three modules in the following combinations: all disabled, enabled one-by-one, enabled in pairs, and all enabled at once.

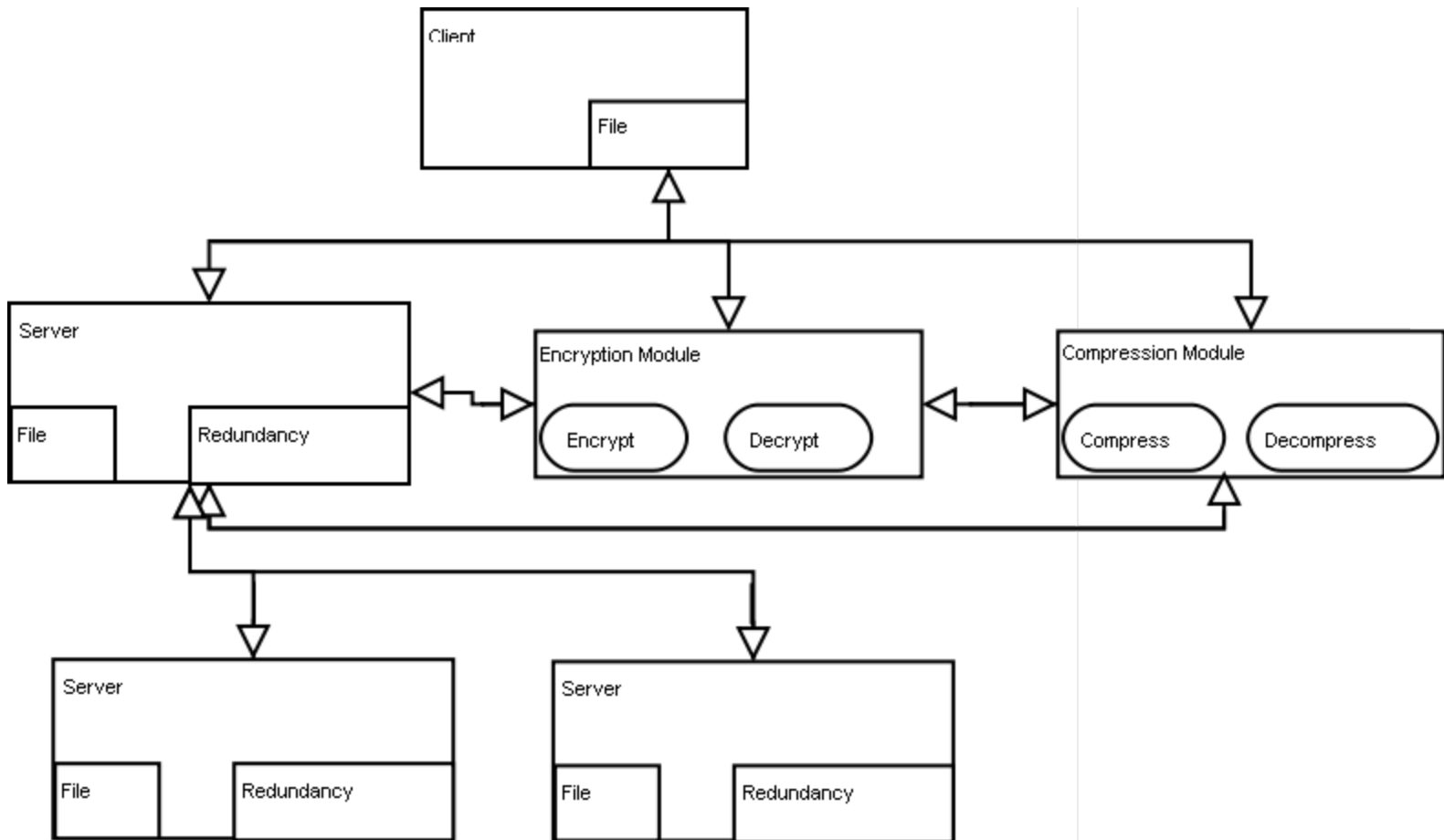


Figure 2.2.1 ANFS architecture. This diagram shows the communication relationships between the modules of ANFS, the client machines, and the connected mirrored servers.

Each scenario will consist of a precondition, an action, and a post-condition. The precondition defines the state of the software before the tested action occurs. The action defines how the integration test will be triggered by running the functions of a component. The post-condition defines the desired state of the software after the tested action.

To automate integration testing, each test case must:

- be self contained with appropriate setup and tear-down mechanisms
- be runnable from a script
- not require any user interaction while running
- provide a single pass or fail result

## 2.3 Test Constraints

The Integration Test Plan will be limited to testing the integration of components of the software as defined in the Design Document for ANFS. It will not test existing Linux kernel code.

## 3 Test Assumptions and Exclusions

This section outlines the level of functionality of ANFS required to complete the Integration Test and further details which features will not be covered in this phase of testing.

### 3.1 Test Assumptions

It is assumed that the software has reached a stable build with all Priority 1 features enabled. The test plan will cover the functionality of the system and the consistency of the functionality with the architecture outlined in the Design Document.

### 3.2 Test Exclusions

The Integration Test Plan will not cover the testing of existing Linux kernel or NFS functions.

## 4 Entry and Exit Criteria

This section details the criteria required to begin and end the Integration Test phase.

### 4.1 Entry Criteria

The following criteria must be met before ANFS can undergo the Integration Test phase:

- The testing environment satisfies the hardware and software requirements outlined in the non-functional requirements section of the Requirements Document.
- The latest stable build of the ANFS software has been set up.
- The Project Leader and Test Team Leader have given their consent.

### 4.2 Exit Criteria

Integration Testing should be stopped if any of the following criteria are met:

- At least one (1) Priority 1 feature deviates from the specifications outlined in the Requirements Document.
- All Priority 1 features are shown to function without deviation from expected behavior as outlined in the Requirements Document.

## 5 Testing Participants

This section describes the roles and responsibilities of any team members involved in the Integration Test phase, as well as the procedure of reporting the test results and any problems



that occur during testing.

## 5.1 Roles and Responsibilities

- Lee Trager: Project Lead
- Arkadiy Sherman: Server Tester
- Matt Wyatt: Quality Assurance
- Mike Dalton: Encryption Tester
- Harry Anuszewski: Compression Tester

## 5.2 Training Requirements

All persons involved in the development and execution of the Integration Test Plan should be familiar with the ANFS documentation, specifically the system commands and requirements.

## 5.3 Problem Reporting

Any bug or deviation from the Requirements Document or the Design Document found during the Integration Test phase must be documented and reported to the correct bug system. Kernel related bugs must be reported to the ANFS Kernel Issue list (<https://github.com/ANFS/ANFS-kernel/issues>) while user space or utility related bugs must be reported to the ANFS Utilities Issue list (<https://github.com/ANFS/ANFS-utils/issues>).

## 5.4 Progress Reporting

Progress reports will be filed after the completion of each test regardless of outcome. A final report will be compiled once the Integration Test phase is complete. This report will be submitted to the Project Leader for record keeping.

# 6 Test Cases

The test cases below are organized according to the availability of the functionality to either an administrative or client machine. All test cases will be written in the following format:

**Name:** The name of the test case.

**Description:** The general summary of the test case.

**Preconditions:** Any conditions which must be met prior to running the case.

**Actions:** The actions a tester must perform to complete the test case.

**Expected Result:** The expected outcome of the test case.

**Failure Condition:** Any deviation from this is considered a failure.

## 6.1 Redundancy Test Cases

## 6.1.1 Adding a Server

### 6.1.1.1 Add server command

|                   |   |
|-------------------|---|
| ID                | TC-R1   |
| Description       | Add an additional server to test redundancy.  |
| Precondition      | A target server is configured, turned on, and ready to be added.  |
| Action            | The admin user runs the command to add a server to the list of available servers while on an existing server. |
| Expected Result   | The target server is added to the list of servers and all clients and other servers are notified.             |
| Failure Condition | 1. The target server is not added to the list of servers.<br>2. No clients or servers are notified.           |

### 6.1.1.2 Checking Data While No Data is Being Manipulated

|                   |   |
|-------------------|---|
| ID                | TC-R2   |
| Description       | Ensure all data is consistent between all current servers and the newly added server. While the check is happening no data is being manipulated by other users. |
| Precondition      | A target server is configured, turned on, and an existing server had added the target.  |
| Action            | The server which was told by the administrator to add a target server ensures all data between the two servers are identical.                                   |
| Expected Result   | The target server has an identical data set to all other servers.   |
| Failure Condition | 1. The target server does not have all data.<br>2. The target server data is not identical.   |

### 6.1.1.3 Checking Data While Data is Being Manipulated

|             |  |
|-------------|--|
| ID          | TC-R3  |
| Description | Ensure all data is consistent between all current servers and the newly added server. While the check is happening data is being modified on the server. |

|                   |  |
|-------------------|--|
| Precondition      | A target server is configured, turned on, and an existing server had added the target.   |
| Action            | The server which was told by the administrator to add a target server ensures all data between the two servers are identical. This includes any data modified while the check is being done. |
| Expected Result   | The target server has an identical data set to all other servers.  |
| Failure Condition | 1. The target server does not have all data.<br>2. The target server data is not identical.  |

#### 6.1.1.4 Notify all clients and servers

|                   |   |
|-------------------|---|
| ID                | TC-R4   |
| Description       | After an administrator has added a target server and the data has been checked to be consistent all servers and clients must be notified that the target server is available for use. |
| Precondition      | A target server has been added and all data is consistent.  |
| Action            | The server which was told by the administrator to add the target server tells all other servers and clients about the new target server.  |
| Expected Result   | All servers and clients are aware that the target server is available for use.  |
| Failure Condition | Not all servers and clients are notified of the new server.   |

### 6.1.2 Intentional Removal of a Server

#### 6.1.2.1 Server Removal Command

|                 |  |
|-----------------|--|
| ID              | TC-R5  |
| Description     | Ensure that a server can be gracefully removed.  |
| Precondition    | A target server is available and is not the only server available.                       |
| Action          | The admin user runs the command to remove a server on the server he/she wants to remove. |
| Expected Result | The server is removed from the list of available servers.                                |

|                   |                            |
|-------------------|----------------------------|
| Failure Condition | The server is not removed. |
|-------------------|----------------------------|

#### **6.1.2.2 Client and Server Notification**

|                   |  |
|-------------------|--|
| ID                | TC-R6  |
| Description       | All servers and clients are notified of the target server being removed.   |
| Precondition      | All servers and clients know about the target server.  |
| Action            | The target server notifies all known servers and clients that it is being removed.   |
| Expected Result   | All clients and other servers have removed the target server from the list of available servers.   |
| Failure Condition | 1. A client or server is not notified that the target server is being removed.<br>2. A client or server maintains a connection with the target server. |

### **6.1.3 Unintentional Removal of a Server (Server Failure)**

#### **6.1.3.1 Client discovery of a failed server**

|                   |  |
|-------------------|--|
| ID                | TC-R7  |
| Description       | A client discovers that a server is no long able to be reached.  |
| Precondition      | A client which is connected to multiple ANFS servers. The client has not been notified of any server being removed.                                    |
| Action            | While preforming any NFS operation the client is unable to reach a target server.  |
| Expected Result   | 1. The client informs the fastest known server that the target is unreachable.<br>2. The client removes the server from the list of available servers. |
| Failure Condition | 1. Not all known reachable servers have been notified.<br>2. The client does not remove the server from the list of available servers.                 |

#### **6.1.3.2 Server discovery of a failed server**

|              |   |
|--------------|---|
| ID           | TC-R8   |
| Description  | A server discovers that a server is no longer able to be reached.   |
| Precondition | Multiple servers are setup and configured for use under redundancy. |

|                   |   |
|-------------------|---|
| Action            | When a server performs any NFS action and is unable to reach the server.  |
| Expected Result   | All clients and servers are notified that the server is no long available and all remove the failed server from their lists of available servers. |
| Failure Condition | 1. Not all clients and servers are notified.<br>2. Not all clients and servers remove the server from the list of available servers.              |

#### 6.1.3.3 Server notified by client of a failed server

|                   |  |
|-------------------|--|
| ID                | TC-R9  |
| Description       | A client notifies the server of a failed server. The notified server verifies this news.   |
| Precondition      | Multiple servers are setup and configured for use under redundancy. A client has notified the server of a possible failed server.          |
| Action            | The server tries to connect to the possible failed server.   |
| Expected Result   | If the target server is actually failed all other clients and servers are notified. If the target server is not failed no action is taken. |
| Failure Condition | 1. The target server is not checked.<br>2. Not all servers and clients are notified.   |

#### 6.1.3.4 Server notified by another server of a failed server

|                   |   |
|-------------------|---|
| ID                | TC-R10  |
| Description       | A server notifies the server of a failed server. The notified server notifies all known clients of the failed server.   |
| Precondition      | 1. Multiple servers are setup and configured for use under redundancy.<br>2. Another server has notified the target server to remove a failed server.                         |
| Action            | The target server notifies all known clients that a server has failed. The server removes the failed server from the list of redundant servers.                               |
| Expected Result   | All notified clients have removed the failed server from the list of redundant servers. The target server has removed the server from the list of redundant servers.          |
| Failure Condition | 1. Not all clients are informed.<br>2. Clients still try to use the failed server.<br>3. The target server does not remove the failed server from the list of failed servers. |

### 6.1.3.5 Client notified by server of a failed server

|                   |   |
|-------------------|---|
| ID                | TC-R11  |
| Description       | A client is notified of a failed server by another connected server.  |
| Precondition      | Multiple servers are setup and configured for use under redundancy.<br>The client has been notified that a server has failed. |
| Action            | The client removes the failed server from the list of redundant servers.  |
| Expected Result   | The failed server is no longer in the list of redundant servers.  |
| Failure Condition | The failed server is in the list of redundant servers.  |

### 6.1.4 Reading or writing data to a server

#### 6.1.4.1 Reading Data

|                   |  |
|-------------------|--|
| ID                | TC-R12   |
| Description       | Ensure data can always be read as long as at least one server is available.                            |
| Precondition      | Multiple servers are setup and configured for use under redundancy.                                    |
| Action            | The client tries to read data from a server. If that server has failed it tries the next known server. |
| Expected Result   | The client can preform read operations as long as at least one redundant server is available.          |
| Failure Condition | The client can not preform read operations when there is another available server.                     |

#### 6.1.4.2 Writing Data

|                   |  |
|-------------------|--|
| ID                | TC-R13   |
| Description       | Ensure data can always be written as long as at least one server is available. |
| Precondition      | Multiple servers are setup and configured for use under redundancy.            |
| Action            | Any write operations is sent to all redundant servers.                         |
| Expected Result   | The write operation is sent to all redundant servers.                          |
| Failure Condition | The write operation is not sent to all redundant servers.                      |

## 6.2 Encryption

### 6.2.1 Encryption Enabled

|                   |   |
|-------------------|---|
| ID                | TC-E1   |
| Description       | In order to encrypt (or decrypt) data, the system administrator must enable encryption. |
| Precondition      | A client is connected to one or more servers.   |
| Action            | The system administrator runs the command to enable encryption on the server.           |
| Expected Result   | Data is successfully encrypted.   |
| Failure Condition | Data is not successfully encrypted.   |

### 6.2.2 Generate RSA Key Pair with OpenSSL

|                   |  |
|-------------------|--|
| ID                | TC-E2  |
| Description       | The user space will generate an RSA Public Key with OpenSSL and then send that key to the kernel.                          |
| Precondition      | N/A  |
| Action            | The client and server will automatically generate these keys when they connect.  |
| Expected Result   | Public Key will be stored in /etc/ssh/ssh_host_rsa_key.pub and the Private key will be stored in /etc/ssh/ssh_host_rsa_key |
| Failure Condition | 1. The public key is not stored.<br>2. The private key is not stored.  |

### 6.2.3 Exchange RSA Public Key

|              |  |
|--------------|--|
| ID           | TC-E3  |
| Description  | The client and server will each generate their own public keys and then they will send each other their public keys. |
| Precondition | Public key has been generated and stored in /etc/ssh/ssh_host_rsa_key.pub  |
| Action       | The client and server will automatically exchange public and private keys.   |

|                   |   |
|-------------------|---|
| Expected Result   | The client has the server's public key and the server has the client's private key.                           |
| Failure Condition | 1. The client does not have the server's public key.<br>2. The server does not have the client's private key. |

#### 6.2.4 Encrypt data using AES from Crypto API

|                   |   |
|-------------------|---|
| ID                | TC-E4   |
| Description       | When a client or server wants to send encrypted data, they do so by generating a private key and using the public key with AES encryption in the kernel's Crypto API.   |
| Precondition      | An RSA public key has been generated for both the client and the server.  |
| Action            | 1. The AES private key is generated with Crypto API using the recipient's public key.<br>2. Send a file to the server.<br>3. Use Wireshark to see that the data is encrypted (i.e. the data is different from what was sent). |
| Expected Result   | Unencrypted Data is now encrypted.  |
| Failure Condition | Data is unencrypted.  |

#### 6.2.5 Decrypt data using AES from Crypto API

|                   |  |
|-------------------|--|
| ID                | TC-E5  |
| Description       | When a client or server receives encrypted data, they can decrypt it by using their private key with AES decryption in the kernel's Crypto API.  |
| Precondition      | An RSA public key has been generated for the recipient.<br>Data has been encrypted.  |
| Action            | 1. The AES private key is generated with Crypto API using the client's public key<br>2. Use Wireshark to see that the data is encrypted (i.e. the data is different from what was sent).<br>3. View the file on the server and see that it is the original data. |
| Expected Result   | Encrypted data is now Decrypted.   |
| Failure Condition | Data is still Encrypted.   |

### 6.3 Compression



### 6.3.1 Compression Enabled

|                   |  |
|-------------------|--|
| ID                | TC-C1  |
| Description       | Configure ANFS with compression set to enabled.  |
| Precondition      | 1. A client is connected to one or more servers.<br>2. A server accepts compressed data.                             |
| Action            | 1. An administrator sets the compression flag to true in the config<br>2. The client saves a file to the ANFS share. |
| Expected Result   | Data is successfully compressed.   |
| Failure Condition | Data is not successfully compressed.   |

### 6.3.2 Compression Disabled

|                   |  |
|-------------------|--|
| ID                | TC-C2  |
| Description       | Configure ANFS with compression set to disabled.   |
| Precondition      | A client is connected to one or more servers.  |
| Action            | 1. A administrator sets the compression flag to false in the config<br>2. The client saves a file to the ANFS share. |
| Expected Result   | Data is not compressed.  |
| Failure Condition | Data is compressed.  |

### 6.3.3 Auto Compression Continue Compression

|                   |   |
|-------------------|---|
| ID                | TC-C3   |
| Description       | Auto compression continues to compress the entire stream.   |
| Precondition      | 1. A client is connected to one or more servers.<br>2. Compression is enabled.  |
| Action            | 1. The client saves a file to the ANFS share<br>2. 10% of the file has been compressed showing a 15% decrease in size.<br>3. The compression module continues to compress the stream. |
| Expected Result   | Data is fully compressed.   |
| Failure Condition | 1. Data compression stops.<br>2. Data is corrupted.   |

#### 6.3.4 Auto Compression Stop Compression

|                   |  |
|-------------------|--|
| ID                | TC-C4  |
| Description       | Auto compression does not compress the stream due to lack of reduction in file size.   |
| Precondition      | 1. A client is connected to one or more servers.<br>2. Compression is enabled.   |
| Action            | 1. The client saves a file to the ANFS share.<br>2. 10% of the file has been compressed showing less than a 15% decrease in data size. |
| Expected Result   | 1. The compression module stops the compression.<br>2. The compression module forwards the uncompressed data to the NFS xdr_stream.    |
| Failure Condition | 1. Data is compressed.<br>2. Data stream is corrupted.   |

#### 6.3.5 Server Does not support Data Compression

|                   |   |
|-------------------|---|
| ID                | TC-C5   |
| Description       | ANFS client connects to a non-ANFS server or ANFS server has compression disabled.  |
| Precondition      | 1. A client is connected to one or more servers.<br>2. Compression is enabled.<br>3. A server is running ANFS with compression disabled or Server is using an older version of NFS.<br>4. A drive is mapped on the local machine.   |
| Action            | The client saves a file to the ANFS share.  |
| Expected Result   | 1. The server notified the client during connection that it does not support compression.<br>2. The client disables compression.<br>3. A warning log is created stating that the server does not support compression.<br>4. The data stream bypasses the compression module and is passed into the NFS xdrstream. |
| Failure Condition | 1. Log is not created.<br>2. Data transfer fails.<br>3. Data is sent compressed.  |

### 6.3.6 Data Stream Inflated Client

|                   |   |
|-------------------|---|
| ID                | TC-C6   |
| Description       | A client copies data from the ANFS share to local machine.  |
| Precondition      | 1. A client is connected to one or more servers.<br>2. Compression Enabled.<br>3. Server supports compression.<br>4. A client has a mapped drive to the server.<br>5. Compressed data stream reaches local machine. |
| Action            | 1. The client has requested Data from the ANFS Server.<br>2. The data stream is received by the client.<br>3. The xdr_Stream is converted into the compressed data stream.  |
| Expected Result   | 1. The Data stream is uncompressed.<br>2. Data integrity is a 100% match to original data.  |
| Failure Condition | 1. Data is not uncompressed.<br>2. Data does not match original data 100%.  |

### 6.3.7 Data Stream Inflated Server

|                   |   |
|-------------------|---|
| ID                | TC-C7   |
| Description       | A server inflates compressed data received from the client.   |
| Precondition      | 1. A Server is connected to one or more clients.<br>2. Compression is enabled.<br>3. Compressed data is received by the server. |
| Action            | 1. The xdr_Stream is parsed into the compressed stream.<br>2. The compressed stream is inflated.                                |
| Expected Result   | 1. Data stream is Inflated.<br>2. Inflated data matches original data 100%.   |
| Failure Condition | 1. Data inflation fails.<br>2. Inflated data does not match original data 100%.   |

## 6.4 Server Operations

### 6.4.1 Client Accesses Servers

|             |   |
|-------------|---|
| ID          | TC-M1   |
| Description | A client modifies data on one of several connected servers. |

|                   |   |
|-------------------|---|
| Precondition      | Several mirrored servers are available for connection and the current client user is connected to one of these servers. |
| Action            | The client user attempts to access and modify information on the server.  |
| Expected Result   | The client is able to access and modify the information on the server.  |
| Failure Condition | The client is not able to access or modify the information on the server.   |

#### 6.4.2 Server Accesses Servers

|                   |   |
|-------------------|---|
| ID                | TC-M2   |
| Description       | A server modifies data on one of several connected servers.                     |
| Precondition      | Several mirrored servers are available for connection.                          |
| Action            | The current server attempts to access and modify information on another server. |
| Expected Result   | The current server is able to access the information on another server.         |
| Failure Condition | The current server is not able to access the information on another server.     |

#### 6.4.3 List Servers

|                   |   |
|-------------------|---|
| ID                | TC-M3   |
| Description       | A client runs a command to list all available servers.  |
| Precondition      | Several mirrored servers are available for connection and the current user is connected to one of these servers.  |
| Action            | The client user runs the command to list all the available servers.   |
| Expected Result   | A list of all available servers is printed with three columns for each server, the DNS address, IP address, and the last recorded latency for the server. |
| Failure Condition | A full list of available servers is not printed with three columns for each server.   |

#### 6.4.4 Notify Servers

|    |       |
|----|-------|
| ID | TC-M5 |
|----|-------|

|                   |  |
|-------------------|--|
| Description       | Servers maintain a list of other available servers   |
| Precondition      | Several mirrored servers are available for connection.   |
| Action            | The current server attempts to update server group connection information on other servers.    |
| Expected Result   | The current server is able to update server group connection information on other servers.     |
| Failure Condition | The current server is not able to update server group connection information on other servers. |