NetFS - Documentation and Usage Manual

Panagiotis Galiotos, Joe Touch, Joshua Train, USC/ISI
{galiotos,touch}@isi.edu and joshtrain@gmail.com

Abstract—**NetFS provides a unified interface that maps the network components of a system to a virtual file system. Thus it significantly simplifies the configuration of the network or the set up of a virtual network by just using well-known file system commands. It also provides the ability to define different policies for different users and components.**

This document describes the installation procedure and the operation of NetFS as well as its requirements.

I. Introduction

THIS document describes NetFS, its requirements, the compilation and installation procedure and its operation. NetFS is a virtual file system that was designed for simplifying and unifying the existing network configuration procedures. It provides a common interface based on the file system (FS) as developed for the UNIX flavored operating systems. Its main operation is to map a system's network components to files and directories and then use several file system commands to access and configure them. It has been implemented as a Kernel Loadable Module (KLM) for the FreeBSD Operating System and its installation and operation is easy and straightforward.

II. Main Characteristics

A. Network Configuration

Network configuration is one of the main operations found in any operating system (OS). However, the amount of different commands and their parameters make this procedure quite complex and cumbersome. Besides, the root access that is required for this procedure sets the system in a vulnerable state. As a result simple users cannot create themselves the network "instances" that they need. NetFS tries to overcome these problems by reusing the FS and its operations to control the network. In that way the user gets a unified API which offers fine-grained control and network virtualization.

B. Main advantages

NetFS provides an interface for configuring the network using well-known file operations. In that sense it unifies the existing interfaces by adding their functionalities to the existing FS. It also provides fine-grained control by using different permission policies for different users. It introduces the concept of hierarchical structure for the network elements of the system which are divided into the categories of either interfaces or routes. NetFS also offers network virtualization.

III. Requirments

NetFS has been implemented as a Kernel Loadable Module. It loads and operates properly in FreeBSD 5.4. It has also been tested in all the previous versions starting from 5.2.

The source code is also available for compilation under these FreeBSD versions.

IV. Compilation

To compile NetFS in your own platform, use the downloadable package. Extract the package under the root directory and execute the joinscourse.sh script. This will create two pointers to the directories fs and modules located in the /usr/src/sys directory.

Next you need to execute make from within the modules directory. If the compilation is completed the klm netfs.ko will be created. Next section describes how to load it and start using it.

V. Installation

A. Install the Module

In order to verify the proper installation of the module you need to copy the netfs.ko to the /boot/kernel directory. Execute the kldstat command to check the already loaded modules in your system. You load the netfs by executing the

kldload netfs.ko

command. Once the module is loaded the user can verify that by executing kldstat again. Note that you don't need to perform this procedure each time you want to use NetFS. Mounting it, as described in next section, automatically loads the module.

B. Mount NetFS

The last step before start using the module is to mount NetFS to your system. We use the mount_std command which is created for mounting "standard" file systems. A "standard" file system is based on a kernel module like NetFS. Do the following steps:

1. create a new directory under your home directory called netfs

   ```
   mkdir netfs
   ```

   This directory will be the mount point for the NetFS.
2. create a new mount command by going to /sbin directory and copying and renaming the existing mount_std to mount_netfs

   ```
   cp mount_std mount_netfs
   ```
3. mount netfs. This is done by going to the home directory and executing:

   ```
   mount_netfs /netfs netfs
   ```

When this procedure is completed you can go and check inside the netfs directory. Two subdirectories have been created; ifaces and routes. The first contains all the information needed to control and configure the network interfaces of your system. The second one refers to the existing routes and the gateways of the system.

VI. OPERATION

Once the module is loaded and netfs is mounted the user can experiment on how to configure the network parameters. The netfs directory contains two subdirectories namely ifaces and routes. Inside these directories the user can find other subdirectories and files that include all the necessary configuration details for each network element.

The main operations currently supported by NetFS are as follows:

the user can view all the interfaces installed in the system by executing:

```
ls netfs/ifaces
```

Inside each interface there is a subdirectory with the IP address. Using the *ls* command as in the example above the user can read the address.

The same applies on the case of the routes:

```
ls netfs/routes
```

Similarly the user can use *ls* to check the gateway used for this route.

To delete an interface the user executes:

```
rmdir netfs/ifaces/em1
```

where instead of *em* the user can have the loopback *lo* interface, the tunnel interface *gif*, the printer port device *plip* or the Ethernet emulation driver (for Firewire) *few*. Note that this is a powerful features provided by NetFS and one should use it with caution. The deletion of an Ethernet interface will result in disconnecting the system from the network without being able to re-install it using NetFS.

Routes can also be deleted in a similar way.

The user can also create an interface by executing:

```
mkdir netfs/ifaces/gif0
```

The current version of NetFS supports the creation of *gif*, and *lo* interfaces.

To change the ownership of an interface the user executes:

```
chown newuser netfs/ifaces/em1
```

To read information from a gateway we use the cat command:

```
cat netfs/routes/route1/gateway
```

Similarly the *cat* command can be used to read information from any file in the ifaces or routes directories.

To change the permissions on a route the user executes:

```
chmod 744 netfs/routes/route1
```

By using these commands NetFS offers a unified API for network configuration that was previously performed by commands like ifconfig. However it retains the compatibility with the old interfaces.

VII. CONCLUSION

The first version of NetFS offers many fundamental network configuration capabilities. It also provides the user with the ability to setup a virtual network according to its needs.

For software updates and latest information please visit our website at http://www.isi.edu/netfs.