

# Network File System (NFS)

- References: Douglas Comer, Internetworking with TCP/IP, p430-432
- Simson Garfinkel and Gene Spafford, Practical UNIX and Internet Security, p605-631
- NFS is used by GNU/Linux client/server file access
  - the Ubuntu clients import user files (held on /home) across NFS
- NFS is used by LTSP
  - the root filesystem is held on NFS
  - includes **all** system binaries

# Network File System

- NFS available on many operating systems
  - OSX, UNIX, GNU/Linux, Free BSD, Solaris
  - Windows
  
- NFS built on top of Sun's RPC (remote procedure call) mechanism
  - open standard, originally produced by Sun Microsystems
  - uses RPC authentication

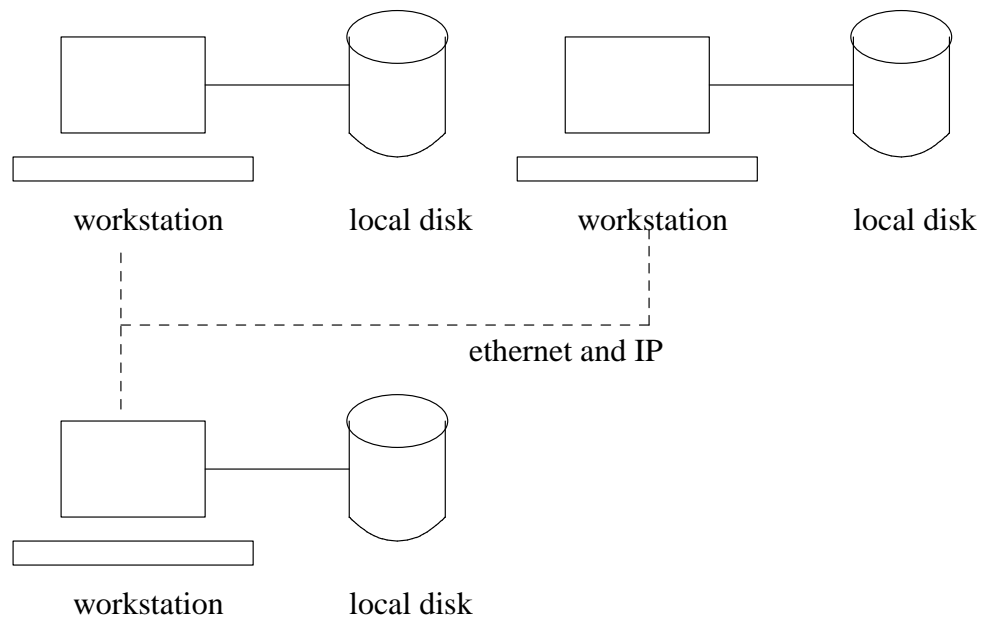
## Network File System Disadvantages

- uses RPC authentication
  - easily spoofed
- filesystem data is transmitted in cleartext
  - data could be copied

## NFS Advantages

- stateless server and client
- server can be rebooted and user on client might be unaware of the reboot
- client/server distinction occurs at the application/user level *not* the system level
  - highly flexible, so we need to be disciplined in our administration/configuration

# A typical GNU/Linux network



- configuration is *very* flexible

## GNU/Linux configuration

- flexibility could be a disadvantage as system administrators have to fight their way through many different options for various components just to perform simple operations
- it is possible that all machines are configured identically (except for their hostname, IP address etc)
  - and ghost a hard disk
- could configure the network operating systems so that all local disks can be seen by all machines

## A typical GNU/Linux network

- alternatively machines could be very different
- one machine has the common file system
- one machine has the system executables

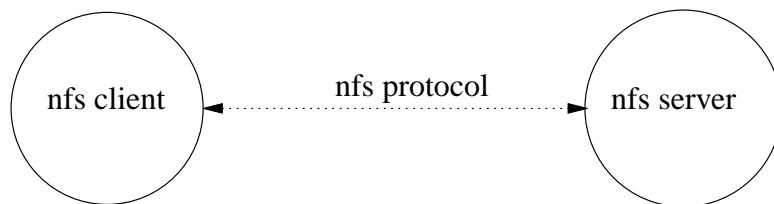
## Disadvantages of GNU/Linux configuration

- complexity
- flexibility requires system administrators be disciplined otherwise overall system may become unmanageable
  - when more machines are added



## NFS configuration example

- can network file systems between different architectures
  - little endian, big endian - it makes no difference
- NFS design is built from three main parts: a protocol, a server and a client



## NFS versions 1 and 2

- both suffered from performance problems
- both suffered from security problems
  - security dependant upon IP address or worse DNS entries

## NFS Version 3

- defined in RFC1813, June 1995
  
- fixes some of the limitations of v2
  - support for 64-bit file sizes and offsets
  - supports files  $\geq$  2GB

## NFS Version 3

- provides support for asynchronous writes on the server
- reply information in the RPC contains additional file attributes
  - which avoids the need to refetch them
- added a READDIRPLUS operation
  - to optimize directory scanning
- obtains multiple file name, handles and attributes

## NFS Version 3

- much effort was put into improving the performance of TCP
  - some implementations rivalled UDP
  
- NFS v3 can operate across TCP as well as UDP
  
- improves performance, and allowed it to work more reliably across the Internet

## NFS Version 4

- RFC3010, December 2000
  - revised in RFC 3530, April 2003
  
- includes more performance improvements
  
- mandates strong security
  - introduces a stateful protocol
  
- developed with the IETF (Internet Engineering Task Force)
  
- NFS v 4.1 extends the protocol to exploit clustered servers
  - distributes files across multiple servers

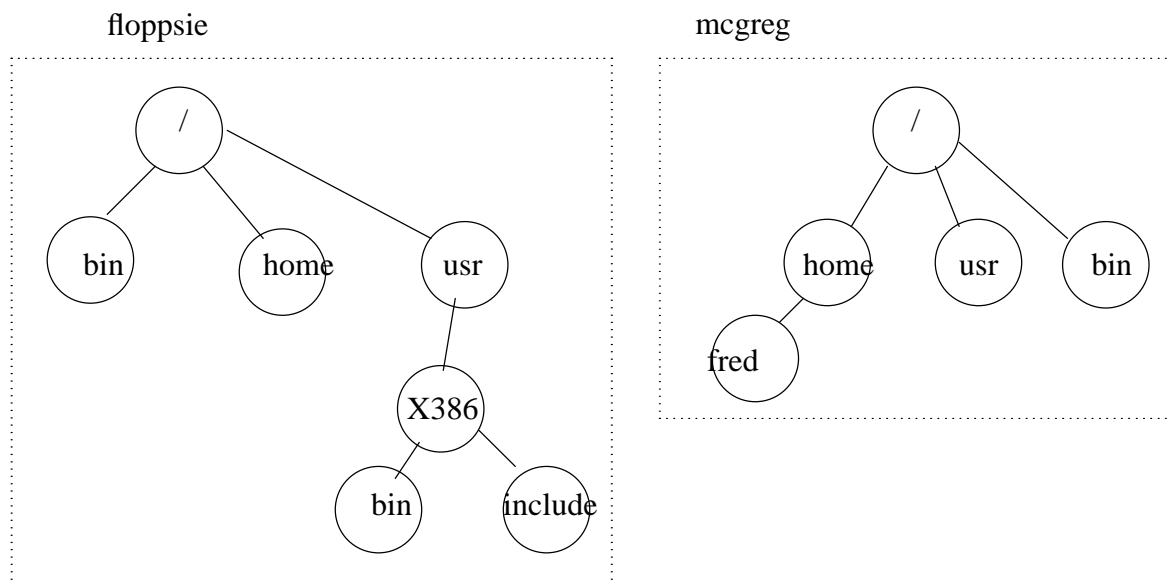
## NFS configuration example

- components fit into the ISO-7 layer model

Level 7	Application	NFS
Level 6	Presentation	XDR
Level 5	Session	RPC
Level 4	Transport	TCP    UDP
Level 3	Network	IP
Level 2	Data Link	Logical
Level 1	Physical Link	Ethernet

# NFS example

- suppose we have two machines





## NFS configuration example

- our task is to allow machine `floppsie` see directory `/home/fred` on `mcgreg`
- tell `floppsie` where this directory resides
- need to tell `mcgreg` that `floppsie` is trusted

## NFS configuration of floppsie

- edit `/etc/fstab` on floppsie and add the following line

```
# remote fs      mount point  type  options
mcgreg:/home    /home/mcgreg nfs
```

## NFS configuration of mcgreg

- need to tell machine mcgreg that /home is exported and that floppsie is allowed to see it
- edit /etc/exports on mcgreg and add the following line
- ```
/home 193.63.128.0/255.255.255.0(rw) \  
193.63.152.84(rw)
```
- why do we use 193.63.152.84 rather than floppsie?