18th January 1995

Support Group Application Note
Number: 267
Issue: 1.02
Author:CAS/RCE

Acorn

Networks:
Fileserver organisation
The Theory

This application note provides information on management techniques and directory structure. It can also be regarded as a supplement/replacement for Chapter 4 of the publication "Networking Acorn Computers" by G Preston, since it includes information which was not available when this book was written. An example of the structure recommended for servers is documented in a separate application note, and is supplied, in outline, on disc with that note.
This document will describe the process of integrating the services offered by Acorn Access and Level 4 into a network. Whilst this is specific to these products, the information is of such a general nature that it can easily be adapted to other Third Party solutions which support Acorn computers.

Applicable Hardware : All RISC OS based computers fitted with network interfaces.

Related Application Notes:

222: Fonts : a shared resource for RISC OS 2 and RISC OS 3.
228: Purchase and installation of a simple AUN network.
231: Optimising AUN network performance.
244: Disabling relocatable modules and other resources in RISC OS.
251: Risc PC boot sequences: Their use in a hard disc or network environment.

Every effort has been made to ensure that the information in this leaflet is true and correct at the time of printing. However, the products described in this leaflet are subject to continuous development and improvements and Acorn Computers Limited reserves the right to change its specifications at any time. Acorn Computers Limited cannot accept liability for any loss or damage arising from the use of any information or particulars in this leaflet. Acorn, the Acorn Logo, Acorn Risc PC, ECONET, AUN, Pocket Book and ARCHIMEDES are trademarks of Acorn Computers Limited.

Support Group
Acorn Computers Limited
Acorn House
Vision Park
Histon, Cambridge
CB4 4AE

Contents:

Introduction

Networks originally found favour in educational establishments because of their ability to share expensive resources. This premise is often overlooked when comparing business models, and even those of higher education, with those found in the typical secondary or primary school. In the past the sharing of user data, as opposed to applications, has differentiated the models, although the usage pattern has also played a large role.

Usage models can be defined in the following ways:

Monogamous Model	A single machine with exclusive use by a single user.
Polygamous Model	A single machine used by a succession of users.
Communal Model	Multiple machines used interchangeably by multiple users.

The monogamous model is commonly found in a business environment, although the polygamous model may exist in small numbers. In the educational enviroment the communal model is the norm with a much higher percentage of machines falling into the polygamous category. Monogamous machines are almost non existent in this environment. An understanding of these models is important as this usually indicates other characteristics of the machine.

It would not be unreasonable for a monogamous machine to have its own hard disc on which applications can be stored, and indeed it is this type of hardware configuration which allows businesses to use networks effectively for data sharing. Until the introduction of the Risc PC it was quite unrealistic for either of the other usage models to have a hard disc **and** to ensure the integrity of the data which is stored on them.

It is for these reasons that many schools are network oriented as the network allows them to protect vital areas of shared applications as well as reduce the effective cost of expensive hardware by sharing it amongst many users. In implementing this model, the network manager usually places quite considerable demands on the network at the start and end of each teaching period - demands which are never realised or even approached in a business environment. In the business environment the demand for applications is made almost exclusively to the local storage media, thereby freeing the network for data sharing. It is a lack of recognition of these demands which leads to the myth that businesses enjoy a higher standard of networking than the education user. It is quite simply that businesses are able to make more appropriate use of the technology available to them without huge penalties in perceived computer down time due to software deletion, accidental or malicious, by the previous user. Further to this, businesses are able to use storage media more appropriately and efficiently, hence reducing the load placed on the network.

In the past, the 8 bit machines such as the BBC Model B and the Master 128 have caused loading problems only when present in large numbers (40+) on a single network. The arrival of the 32 bit RISC machines, such as the Archimedes and A3000, aggravated this overnight as a single application for these computers could be 400K or more in size ; consequently, a single RISC OS computer could almost load the network by the same amount as 40 Master 128s loading a 10K program.

As schools have placed more demands on the network, so software has been developed which alleviates some of the problems. This software, along with appropriate network design and organisation, has enabled schools to continue to use network technology to good effect.

This document will examine the how this can be achieved by considering the setting up of a network based on the Level 4 Fileserver Release 3 and Acorn Access software. In addition it will discuss and introduce mechanisms for providing more flexible computer organisation, configuration and security.

Network models

There are two basic network models which can be mixed and matched accordingly. In Acorn networking the historical order is:

Client/Server model Peer to Peer model

Client/Server Model

This consists of a dedicated server or servers which provide fileserving, printer serving or application serving. In some cases, complementary processes can be utilised on a single computer, however, this can only be recommended if the number of clients is small or the pattern of usage excludes the possibility of many clients attempting to access the same resource simultaneously. This model usually involves some form of validation from the client computer before the service can be used. The illustration below shows 6 stations with access to (from left to right) a printer spooler, application server and Level 4 Fileserver.

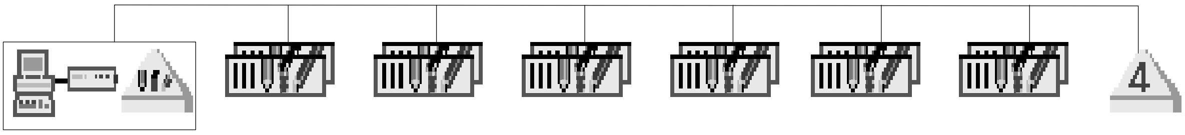


Figure 1.0: A Client/Server network.

Peer to Peer Model

This consists of a number of client computers sharing their resources with each other in a cooperative way. There is usually no requirement for a user to validate their use of a particular resource. The illustration below shows 5 stations sharing a single printer and their individual hard discs.

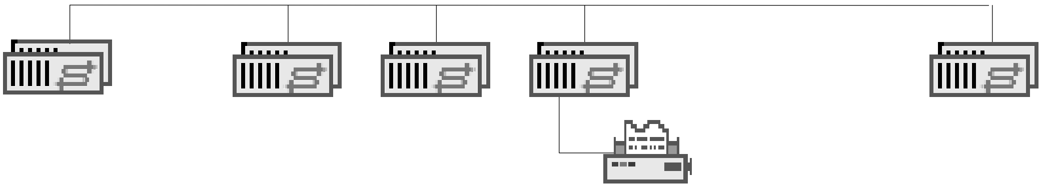


Figure 1.1: A Peer to Peer network.

Combined Client/Server and Peer to Peer

There can be considerable advantages in combining these two models. The combination which Acorn currently recommends is illustrated in figure 1.2. This is based on the use of Acorn Access to provide the client stations with access to the application server, and shared discs and printers. The use of a Level 4 Fileserver ensures that important work can be stored in a secure place. One very important concept of this model involves the use of repeaters (both simple and multiport), bridges and Acorn computers acting as Gateways. Repeaters enable the length and number of computers on a network to be extended (within limits).

Bridges extend these limits further, since they operate in a different manner. Gateways also provide a similar function, however, gateways can form a barrier which prevents data from Peer to Peer services reaching other sub-networks. All computers can see all the fileservers and all !Spooler based printer servers, allowing access to secure personal data stored on a fileserver from anywhere on the network.

A complete discussion of the differences between repeaters, bridges and routers (of which !Gateway is an example) is beyond the scope of this document : network installers should be aware of these differences and their implications.

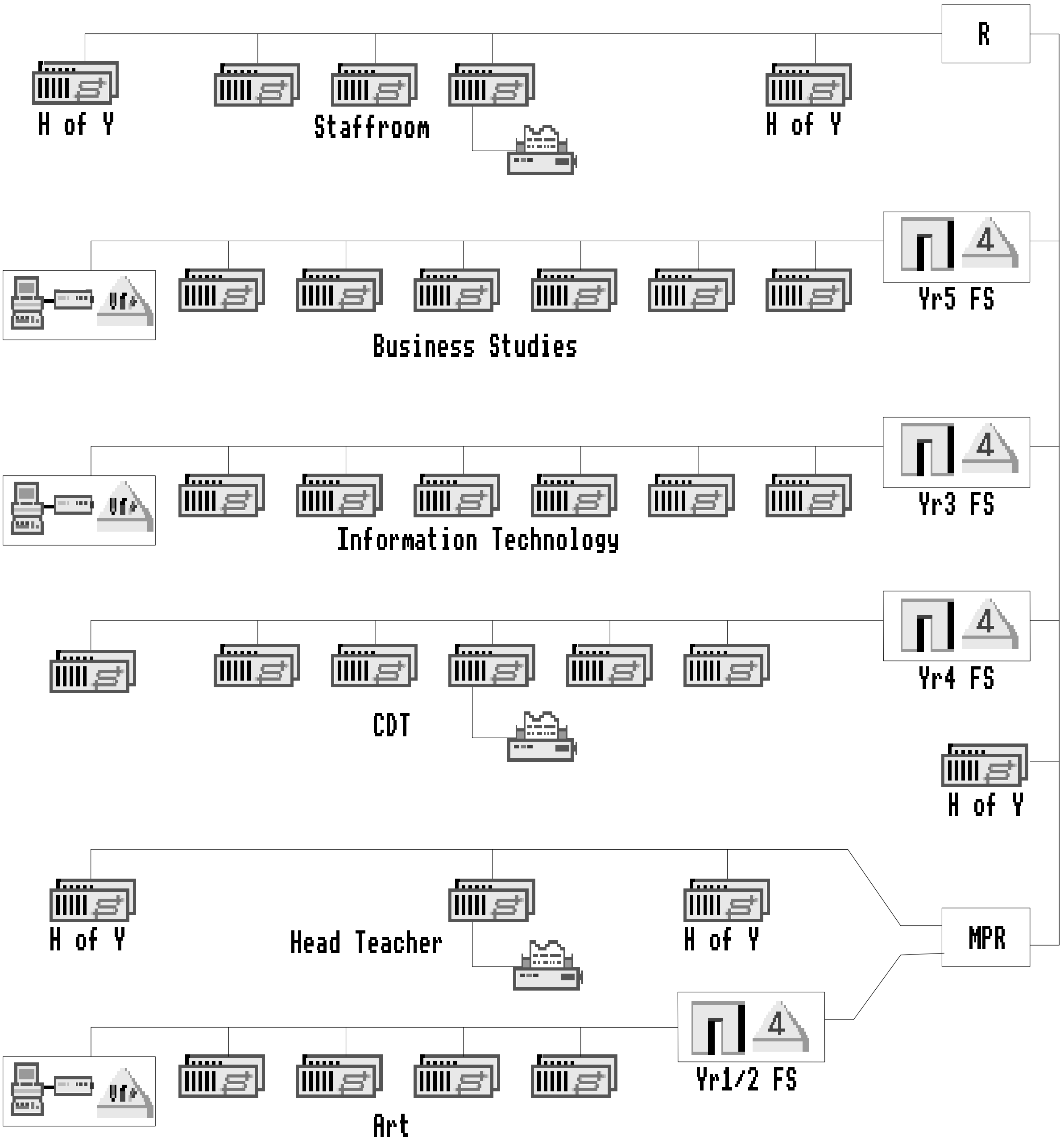


Figure 1.2: The combined model.

It is the building and use of this last model which will be the focus of this document.

Software overview

This section will deal with the software which forms the basis for the two basic network models. It will cover the application software supplied by Acorn for use in these models. Some of this information can be applied to other network software from external Acorn developers.

The Client/Server software: AUN/Level 4 Fileserver (Release 3)

!Server

This application provides clients with access to remote hard disc storage. Network Managers may allow clients (also known as users) to have their own private area on a disc which the user can protect from unauthorised access via a password.

To date there have been three releases of this application. The first release can be easily identified from subsequent releases by the icon which is used to represent the server. The icons are:

!Server: release 1 version 1.00

!Server: release 2 and 3 version 1.10 or later

Sites are strongly advised to upgrade from Release 2 to Release 3 of the fileserver as the later release can improve performance by a worthwhile factor. Sites with Level 4 Release 1 should budget for an upgrade immediately as there are significant functional differences between the two types which will benefit existing sites.

Many sites use the fileserver as a management machine. This is a good and appropriate use of the hardware, however, great care must be taken when performing management tasks in order to avoid problems with !Server. (See later).

!Manager

This application is used to create, delete and otherwise manage the profile of user accounts on !Server. It is supplied as part of the Level 4 Fileserver software.

It was the first application to provide a Desktop means of managing user profiles, but it has since been superseded by other network management utilities such as !NetManage by Suitable Software.

!AAServer

This software provides fast, read only access to the whole or part of a disc for the purpose of loading applications. There has only ever been one release of this software. Generically it is a sub-set of the Acorn Access Peer to Peer software and its speed and user interface are very similar. It provides a performance enhancement for loading applications which is typically 4 times faster than that which !Server is able to provide. It requires a complementary application !AAClient to be loaded into the client station from the fileserver or local disc. Client stations which are fitted with Acorn Access cards can utilise !AAServer resources without any additional software. It is possible with this software to easily hide exported areas and to then provide access to them via a menu system such as !Waiter. Discs which are are exported by !AAServer are always represented by the icon:

Exported !AAServer disc icon

!AAClient

This is the complementary application to !AAServer. It must be loaded into the client from the fileserver or a local disc. It has the same user interface as Acorn Access except that it does not allow client stations to view and hence use, Acorn Access resources.

!Spooler

This is a printer spooler which is capable of supporting any Acorn based client computer on the network. It provides support for up to 8 printers on the network; either 8 physical, 8 logical or a combination of both. Performance can vary depending upon the type and quantity of printers attached. It requires a local hard disc and it is recommended that a computer running !Spooler is not used as a client station as well.

!Bootnet

This is the disc based equivalent of some of the software provided in the Ethernet card EPROM. It usually contains more recent versions of the software which, if loaded into computers which act as servers, can improve performance of the network. It is worth noting that unless the client computers have a local hard disc there is little benefit in attempting to load this software from floppy disc. It is not possible for client computers to load this software from other network based resources as this will sever all network connection and leave the software in an undefined and hence unuseable state.

!Gateway

This software provides a means of linking two different or similar network types.
eg.

Econet/Ethernet
Ethernet/Ethernet
Nexus/Ethernet
ClassNet/Ethernet

As well as providing the physical link between the two network types it also holds a list (called a map) of the networks on the whole site. It uses this information to forward data to other stations on the site. This application ***cannot*** be used if any of the computers on the network are running Acorn's TCP/IP suite, since it only understands the subset of TCP/IP that is used by AUN, thus only Acorn RISC OS computers can communicate and pass data to other computers via !Gateway.

!NetUtils

This application is required by ***all*** RISC OS 3.10 ***and*** 3.11 computers. It resolves problems with files which are created in certain ways and is essential if data is not to be lost. RISC OS 3.11 includes the NetUtils module, however, it is not guaranteed that it will be activated if the network is anything other than Econet.

It is sufficient to simply treat this in the same way as !Fonts and !System. It will not be activated by versions of RISC OS above 3.11 or below 3.10.

The Peer to Peer software: Acorn Access

Acorn Access

This is the software which provides direct access, via a network, to remote hard discs and/or printers. In order to do this the remote computer must also be fitted with an Acorn Access card. Whole or part discs can be exported to other stations on the network in one of two forms:

Protected

or

Unprotected

These discs can also be identified as "hidden" in which case they will not appear in Access disc windows. The concepts behind Protected and Unprotected will be discussed later in this document.

!Printers (Version 1.22 or later)

Supplied with the Acorn Access hardware is a new version of the !Printers application. This version will become the standard !Printers application for all Acorn computers irrespective of the presence of a network connection.

This version has the ability to recognise computers which are fitted with an Acorn Access card and allow the sharing of printers.

Computers which are used to provide shared printing resources via Acorn Access should have their own local hard disc. Whilst it is possible to use a floppy disc this is not to be recommended as performance will suffer as a result.

Important concepts and information

This section will deal with some of the less obvious concepts and points of setting up a network. This section will not offer a solution, but will make the reader aware of some of the important issues which will affect the solution which they may ultimately choose.

Access permissions

Both Acorn Access and !Server can provide restricted file access to users. This is achieved by the use of the access permissions which are attached to a file or directory. Acorn Access and !Server treat the file permissions in subtly different ways. If you are not familiar with access permissions you should read the section *Access* in the chapter **The Desktop Filer** of your computers ***User Guide***.

!Server access permissions

Files

Locked: means that the file cannot be deleted. By default, this flag is not set.

Owner read: means that the file's owner can read the file (for example, by loading it into an editor). By default, the owner has read access.

Owner write: means that the file's owner can write to the file (ie, can change it and save it in the same place with the same name). By default, the owner has write access.

Public read: means that other network users can load your files or run an application which is stored in your network space. By default, other network users do not have read access.

Public write: means that other network users can write to the file. They cannot change or overwrite the file in the normal way, only by using bytewise file access mechanisms, such as OPENIN, OPENOUT, etc.
By default, other network users do not have write access.

Directories (including applications)

Locked: means that the directory cannot be deleted. By default, this flag is not set. If !Server has ***hidden objects*** set then this attribute is used to prevent the directory or its contents appearing in directory displays for all but system privileged users, or its owner.

Owner read: Not applicable.

Owner write: Not applicable.

Public read: Recognised, but has no effect.

Public write: Recognised, but has no effect.

Acorn Access access permissions

Note: If an Acorn Access disc is shared "Unprotected" then all users are treated as owners.

Files

Locked: means that the file cannot be deleted. By default, this flag is not set.

Owner read: means that the file' s owner can read the file (for example, by loading it into an editor). By default, the owner has read access.

Owner write: means that the file' s owner can write to the file (ie, can change it and save it in the same place with the same name). By default, the owner has write access.

Public read: means that other network users can load your files or run an application from your network space. By default, other network users do not have read access.

Public write: means that other network users can overwrite (eg save to) the file. By default, other network users do not have write access.

Directories (including applications)

Locked: means that the directory cannot be deleted. By default, this flag is not set.

Owner read: Not applicable.

Owner write: Not applicable.

Public read: means that the directory will be visible on a protected Acorn Access disc.

Public write: means that the directory can be written to on a protected Acorn Access disc. eg. sub-directories or files can be created inside the specified directory. This is an important feature for addressing !Scrap issues on protected Acorn Access discs.

Application and directory organisation

The organisation of RISC OS applications and data is very important if the network is to provide a speedy response to the client station. Frequently applications are simply grouped into a directory called Apps with little or no consideration on the effect this will have on performance : this is one of the most overlooked aspects of setting up a file or application server.

The contents of a directory on a fileserver may change between one network filer operation and the next, and of course the filer operations may be coming from different client computers. This means that the network filing system in the client computer cannot assume that the information it holds is a true reflection of the current state of the directory. This means that the the network filing system has to re-examine the directory contents each time the client station accesses files etc on the fileserver. Clever caching mechanisms in RISC OS 3 ensure the best performance possible, but these will not take effect until after the first Desktop access to a particular directory.

Careful structuring of the directory contents will assist these mechanisms and ensure that the system will provide the optimum performance. An understanding of the filer operations (in very simple terms) should illustrate why this is so important.

When a computer requests a catalogue of a directory, the filer has to interrogate the contents of the directory to know what it contains and how to display its contents on the screen. This is what happens when a directory containing a text file is opened:

The Filer gets a list of the contents.
It scans the list to see what objects are present.
It finds a single entry which is a file of type text.
It scans its pool of sprites to see if it has a sprite which represents a text file.
It does, so the sprite is used to display the text file in the directory viewer.

If the directory were to contain an application then the following would happen:

The Filer gets a list of the contents.
It scans the list to see what objects are present.
It finds a single entry which is an application.
It scans its pool of sprites to see if it has a sprite which represents the application.
It doesn' t, so it opens the application and looks for a !Boot file.
If found, it *RUNs the !Boot file. This could involve setting filetypes, aliases, etc. and should also load a sprite representing the application.
It scans its pool of sprites to see if running the !Boot file has added the sprite which represents the application.
If it hasn' tfound a !Boot file (and/or the sprite), it opens the application and looks for a !Sprites file.
It finds the sprite file and places its contents in the WIMP sprite pool.
It scans its pool of sprites to see if it has a sprite which represents the application.
It does, so the sprite is used to display the application in the directory viewer.

This process is repeated for every entry in the directory until every object is examined and an appropriate sprite identified to represent each object. As you can see, a directory containing 10 applications is going to take many times longer to display than one containing a similar number of files.

It is for these reasons that a well organised hierarchical structure can improve performance. Here are some golden rules which you should adhere to when designing your directory structure:

Golden Rules

1. Place vital system resources at the entry point into the structure.

2. Keep the number of entries in the exported root to a sensible minimum. If you have more than about 10 objects in the root directory then you have probably not organised your data well. Keep the number of entries in other directories manageable - less than 70 is sensible.

3. Give things meaningful and obvious names so that people can find their way around easily.

4. Place resources that clients are going to need on the path to the files that use them.

5. Place example files in a sub-directory of the main application directory or even on a separate resource to prevent unnecessary filer operations.

6. Never duplicate resources in sub-directories as this will make updating more difficult in the future.

Directory organisation.
As can be inferred from the above description, the directory structure which is adopted on a server can have a profound effect on the overall performance.

The maximum number of files allowed in a single directory on Econet fileservers is 255 with the exception of the FileStore series of fileservers where the limit is 254. The 254 limit was imposed on FileStores so that it could implement the `*Dir ^` command on machines which could support it.

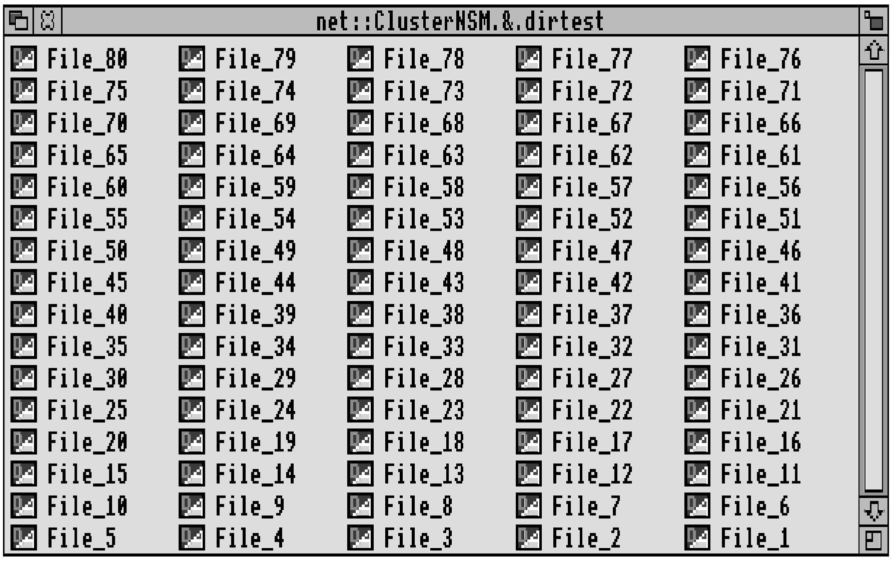
All the fileservers prior to Level 4 implemented their own filing system and consequently direct access from the "normal" local filing system was not possible. With the Level 4 Fileserver the normal filing systems available on the machine are used. This typically imposes a limit of 77 entries per directory. The Level 4 Fileserver is able to support 77 files in the exported root ($) for the network and up to 255 files per sub-directory of that root. This limit is achieved through the use of a clever software technique called *Extended directories*. This is what happens:

The fileserver will create files etc in a directory as normal using the correct software interface for the exported filing system. It will do this until there are 76 entries in the directory. When the creation of the 77th entry is requested the fileserver will create a sub-directory of the parent and place the 77th file in there. This continues until the sub-directory' s76th file is created, whereupon the process is repeated. This continues until the total number of network valid entries reaches 255 : however, just because these large numbers of entries are supported does not mean that it is a good idea to use them - it is a good idea to limit the number of entries in any directory to those with can easily be seen on screen in one window, about 40 maximum. If more than this are required, it suggests that the organisation of the server is less than optimal, and that some re-thinking of the structure may be called for - the following paragraphs expand on why this is the case.

Another factor that has a profound influence on the server' sperformance is the size of the configured directory cache - that area of memory which is set aside for the system to hold the details of recently accessed directories. If this is set to too small a value, then it will significantly degrade the performance of the server, since it will constantly have to access the disc to read directory information, in order to access files and directories lower in the structure, or to move to different parts of it - this information can usefully be held in memory ready for immediate use. It is suggested that between 64K and 128K be allocated to the directory cache of the fileserver' s main filesystem(s).

When a client machine catalogues the directory, the Level 4 Fileserver provides information about all the files and directories except those which it created to extend the number of entries to 255. If the directory is catalogued at the fileserver, then the directory entries and the special ones created by Level 4 are displayed as normal.

The special directories appear to have no title; in fact they use the hardspace character (ASCII 160) which cannot be generated accidentally at the keyboard. If we examine a directory from a client machine which has 80 files in it, it will look like this:

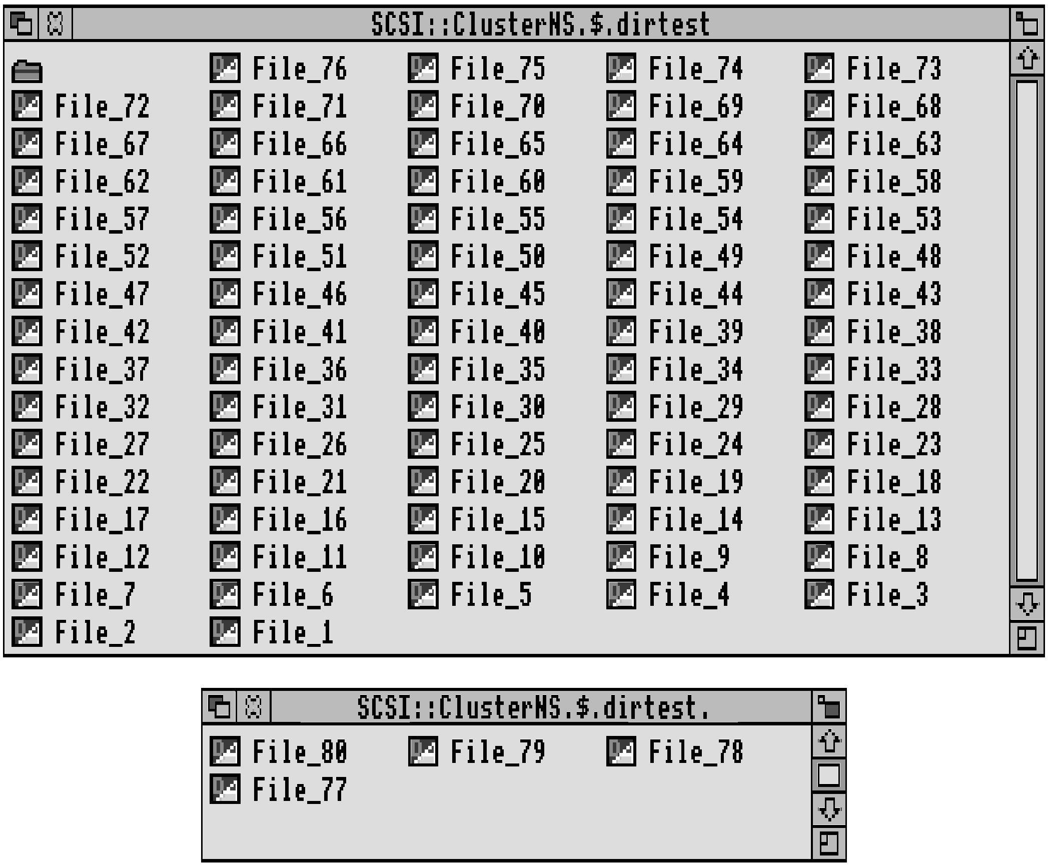


```
net::ClusterNSM.&.dirtest

File_80   File_79   File_78   File_77   File_76
File_75   File_74   File_73   File_72   File_71
File_70   File_69   File_68   File_67   File_66
File_65   File_64   File_63   File_62   File_61
File_60   File_59   File_58   File_57   File_56
File_55   File_54   File_53   File_52   File_51
File_50   File_49   File_48   File_47   File_46
File_45   File_44   File_43   File_42   File_41
File_40   File_39   File_38   File_37   File_36
File_35   File_34   File_33   File_32   File_31
File_30   File_29   File_28   File_27   File_26
File_25   File_24   File_23   File_22   File_21
File_20   File_19   File_18   File_17   File_16
File_15   File_14   File_13   File_12   File_11
File_10   File_9    File_8    File_7    File_6
File_5    File_4    File_3    File_2    File_1
```

Figure 1.3: Directory window containing 80 files viewed from a client station onto a fileserver.

If we examine the same directory from the keyboard of the fileserver we see that it is made up from two directories as shown in the following illustration:



```
SCSI::ClusterNS.$.dirtest

          File_76   File_75   File_74   File_73
File_72   File_71   File_70   File_69   File_68
File_67   File_66   File_65   File_64   File_63
File_62   File_61   File_60   File_59   File_58
File_57   File_56   File_55   File_54   File_53
File_52   File_51   File_50   File_49   File_48
File_47   File_46   File_45   File_44   File_43
File_42   File_41   File_40   File_39   File_38
File_37   File_36   File_35   File_34   File_33
File_32   File_31   File_30   File_29   File_28
File_27   File_26   File_25   File_24   File_23
File_22   File_21   File_20   File_19   File_18
File_17   File_16   File_15   File_14   File_13
File_12   File_11   File_10   File_9    File_8
File_7    File_6    File_5    File_4    File_3
File_2    File_1

SCSI::ClusterNS.$.dirtest.

File_80   File_79   File_78
File_77
```

Figure 1.4: The same directory viewed on the fileserver, showing how the 80 entries are organised.

Whilst all fileservers can support large numbers of entries, it is bad practice to organise the files on the fileserver so that they form a "flat" structure : such a structure can take a considerable amount of time to access and locating files or applications can be time consuming and tedious.

We may consider the fileserver to be just a high tech filing cabinet. A group of users may be allocated a drawer in the filing cabinet and their own folder within that drawer. Each user can easily identify their own work, because it is in their folder which is separate from all the other folders. The owner of the folder may further subdivide their folder so that they can find things more easily. This type of organisation is based upon a hierarchy or ' family tree' style structure.

If the drawer within the filing cabinet was badly organised, then finding things would become increasingly difficult. Imagine a drawer in a filing cabinet which contained no folders, just documents. Finding a particular document may take a considerable amount of time.

Network directory structures follow the same principles. A group of users may be allocated a directory (drawer) on the fileserver (filing cabinet) and their own sub-directory (folder) within that directory (drawer). Each user can easily identify their own work because it is in their sub-directory which is separate from all the other sub-directories. The owner of the sub-directory may further subdivide their sub-directory so that they can find things more easily.

Figure 1.5 illustrates part of a well organised fileserver which uses a hierarchy or ' family tree' type directory structure.

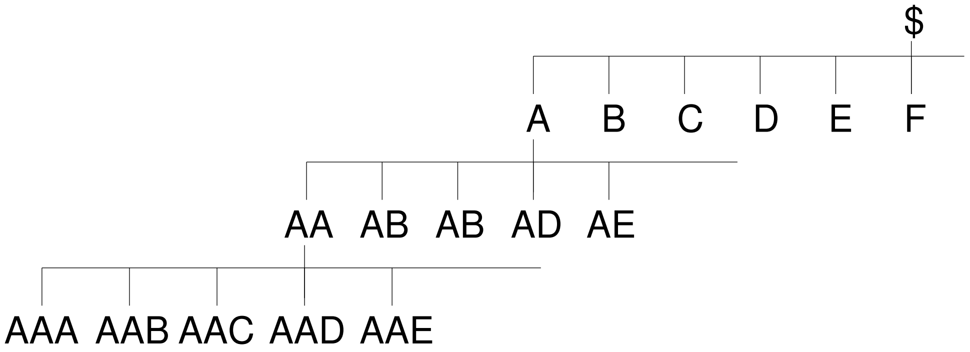


Figure 1.5: A hierarchical directory structure.

Figure 1.6 illustrates part of a badly organised disc which utilises a flat directory structure.

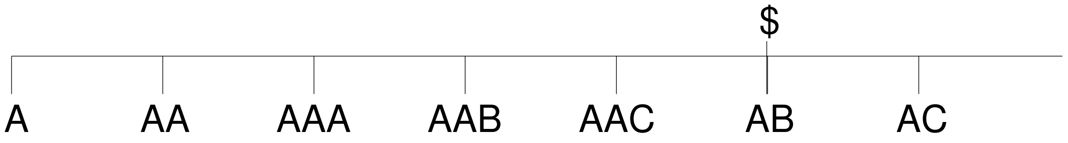


Figure 1.6: A flat directory structure.

Frequently the directory structure which exists on the fileserver is a combination of both of these models, but the effort involved in establishing a hierarchical structure will pay dividends may times over in terms of the performance and useability of the network.

It is strongly recommended that the URDs for all normal users are place in Net:$.Users, as opposed to being placed in the network root directory, since this leads to a tidier structure, as described above.
Suitable grouping of users can also greatly reduce the administration overhead of the system - the suggested grouping is based on the pupil’s year of entry to the school, and any ‘house’ or tutor group assignment that does not change on a year to year basis.
For example, Yr94red is a much better group name that Yr7RCE, where RCE is the form tutor, since at the end of each year, year 7 becomes year 8, year 8 becomes year 9 etc., and it is also likely that the form tutor

will change, leading to a great deal of administrative work each summer. The suggested group name will remain with the pupil throughout their school career. If the school does not have a ‘house’ system, it is worth while trying to find something similar - anything that will not change regularly, even the month of their birthday if all else fails !
The actual username should be as simple as practicable, so as to uniquely identify the individual within their group - typically a few letters of the surname and an initial suffice.

In situations where there are multiple servers, or multiple discs on a single server, I strongly suggest that the URDs are distributed across the discs based on a simple, alphabetic split, based on the username, and obviously, the size of the disc ; the aim of this is to split the groups as evenly as possible, to distribute the workload - a single class will result in all servers and/or discs being in use, as opposed to all users accessing one, and to split the requirement for disc space - generally, older pupils require more space than younger ones, this strategy tries to fill all discs at about the same rate. The disadvantage to this policy is that it does increase the amount of work to be done when creating groups of new users - you have two or more groups to create, split over the multiple servers - it is well worth the work involved !

Autobooting client stations

It is important to ensure that all client stations boot from the fastest filing system available to them so that the computer is ready to use in the least amount of time.

On a network the preferred order is :-

	Filing system	Boot name
Fastest	Local hard disc (ADFS, SCSI etc)	!Boot
	Acorn Access	!ShareBoot
	Network fileserver & Application Accelerator	!ArmBoot & !ShareBoot
Slowest	Network fileserver	!ArmBoot

Obviously the use of some of these options is precluded by the hardware, but with Release 3 of the Level 4 Fileserver and Ethernet there is no reason why a client station should boot solely from the fileserver.

The use of the fileserver/Application Accelerator to boot client stations is a somewhat lengthy topic, which will be discussed elsewhere : the process can be replaced and enhanced by the use of Acorn Access. On a correctly setup network, booting from Level 4 and the Application Accelerator, but loading no other applications, it should be possible to have a working network of 16 client stations in about 90 seconds from switch on.

CPU usage

This section will look at the load placed on the CPU (Central Processing Unit) by various network services. The figures are of a very general nature and are a projection of the load over a 2 hour period in a school classroom, assuming a 1 hour teaching period. In short the figures are not, and cannot be, accurate, but do indicate which processes can work successfully together and which cannot.

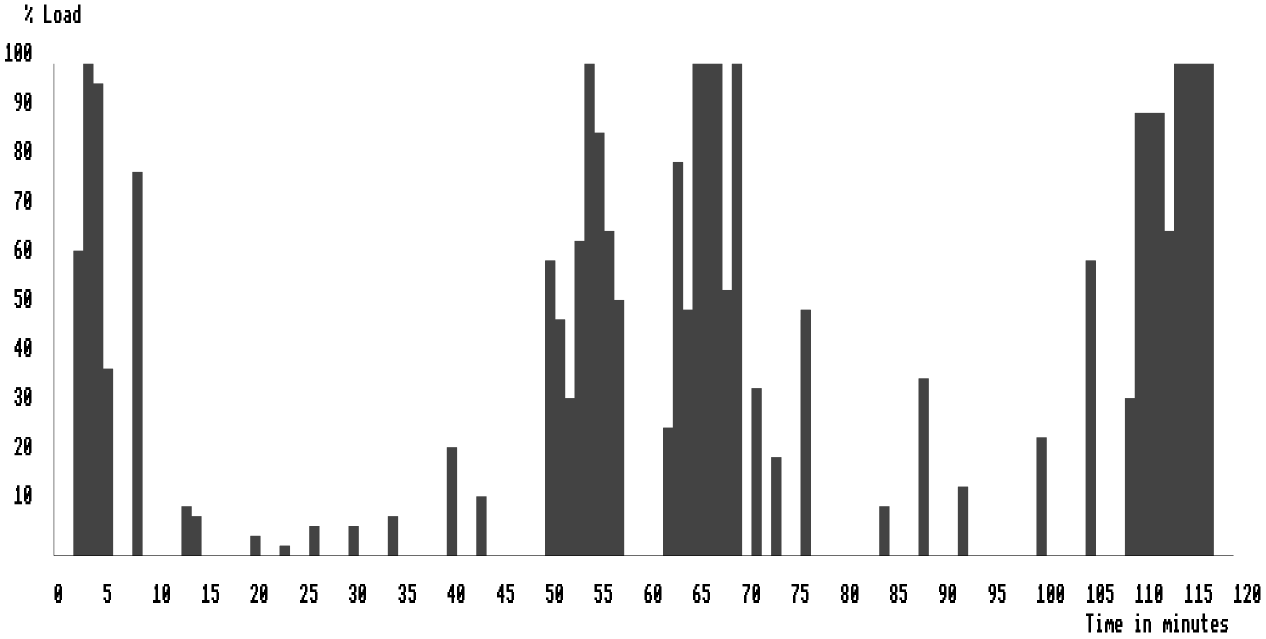


Figure 1.7: Load placed upon the CPU by the Level 4 Fileserver software in a typical classroom environment.

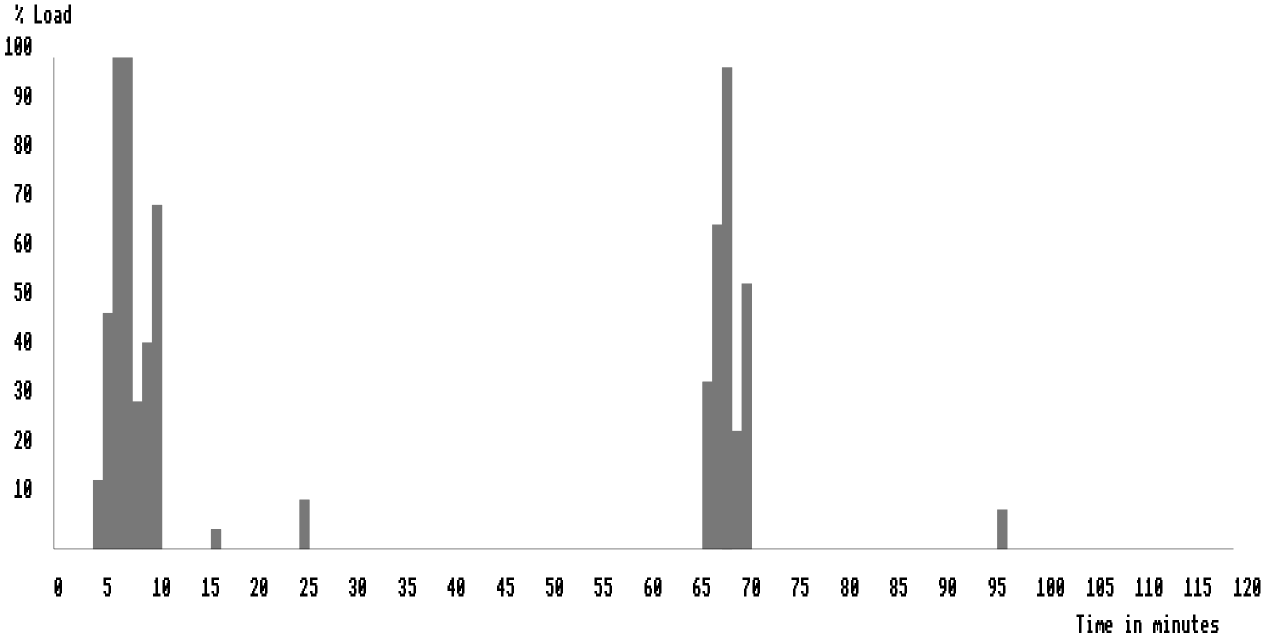


Figure 1.8: Load placed upon the CPU by the Application Accelerator server software in a typical classroom environment.

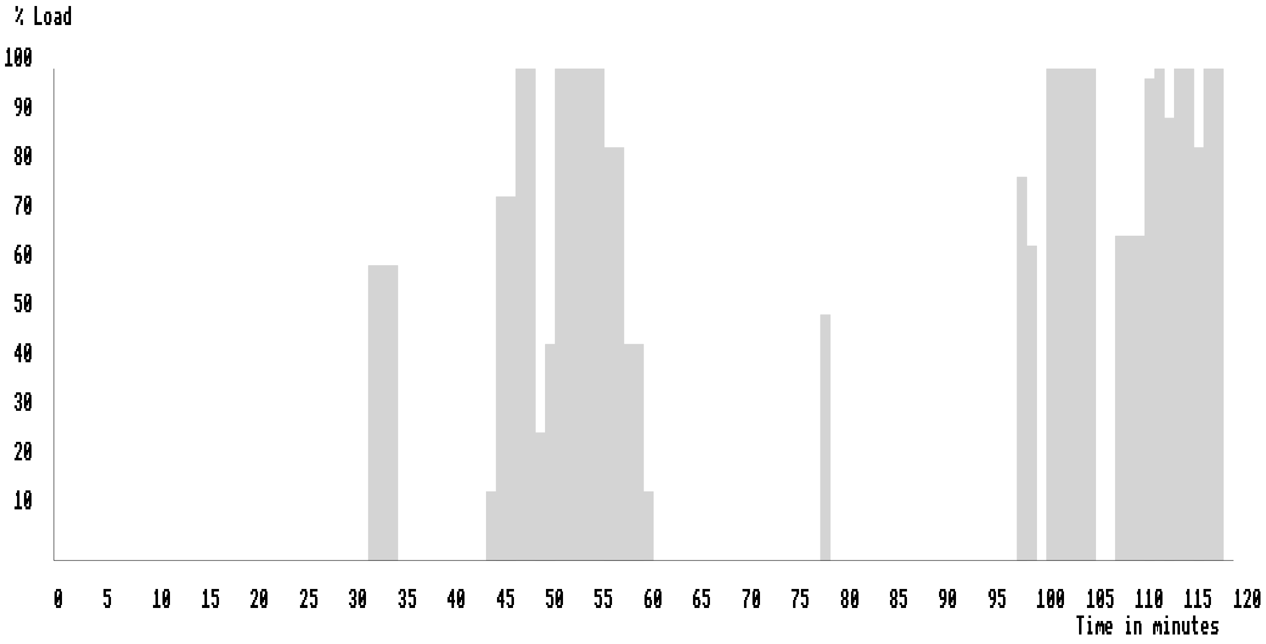


Figure 1.9: Load placed upon the CPU by the !Spooler software in a typical classroom environment.

As you can see from the illustrations above it would be unwise to utilise a single computer as a fileserver and printer server due to the load placed upon the CPU by the !Spooler and !Server applications. However, whilst the application server and fileserver could coexist reasonably well together, the ideal combination is for a combined printer and application server. The CPU loading of these processes is illustrated below:

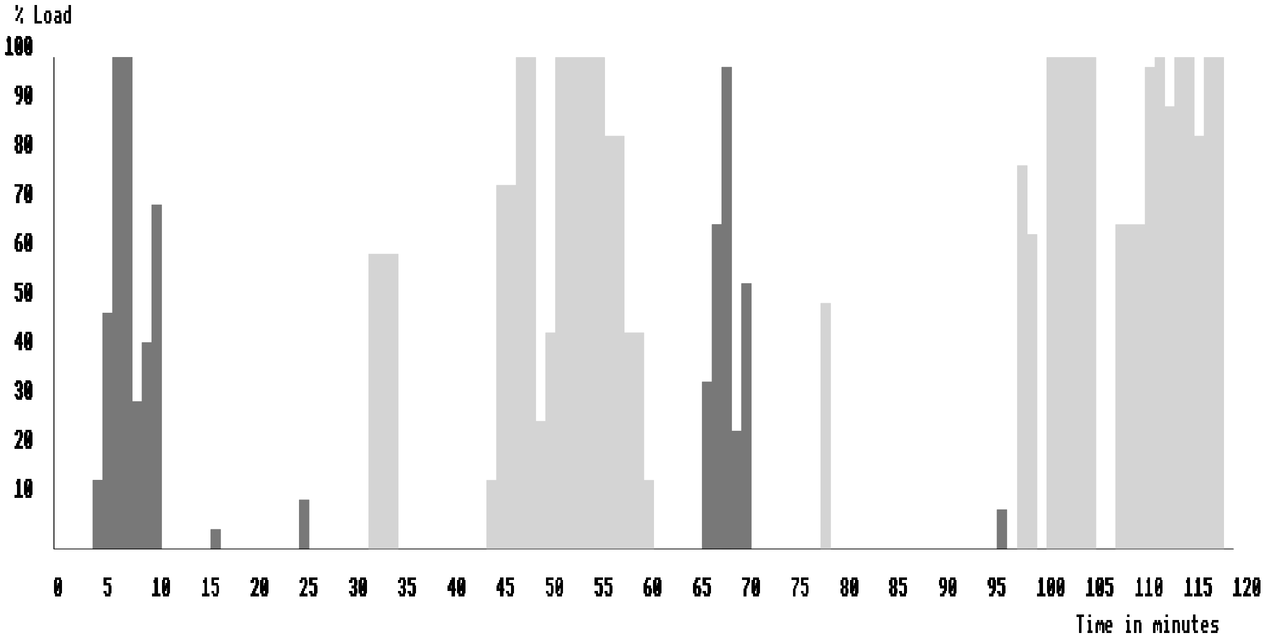


Figure 2.0: Load placed upon the CPU by the Application Accelerator and Printer Spooler software in a typical classroom environment.

Barriers

This section will examine the mechanisms for providing natural barriers to network traffic from certain types of application, and the implications of this for resource availability.

If we re-examine our ideal network we find that there are actually 7 network segments which form a total of 5 networks as shown below.

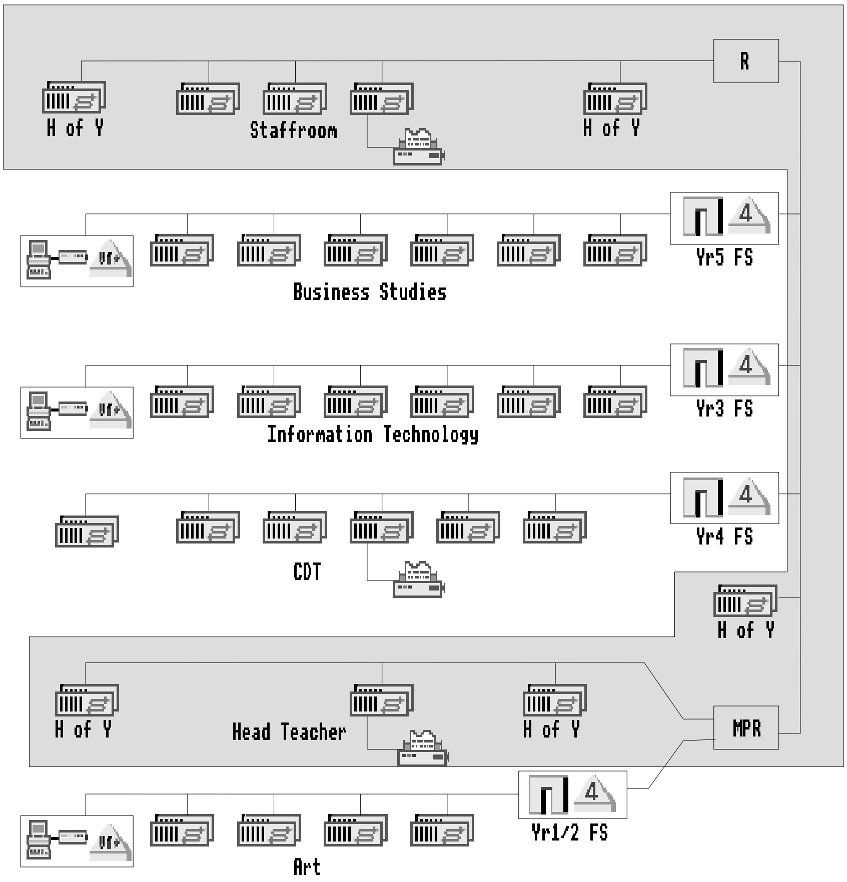


Figure 2.1: Ideal network configuration showing main networks

These are:

Backbone Highlighted by the grey backbround and consisting of 3 segments separated by 2 repeaters. This network provides for various administrative points in the school such as in the Staffroom, Head of Years offices and the Headmasters office.

Business Studies This is a single network segment.

IT This is a single network segment.

CDT This is a single network segment.

Art This is a single network segment.

The use of a single port repeater (R) and a multi port repeater (MPR) means that traffic on the backbone is not restricted in any way. This allows any Acorn Access based client stations connected directly to the backbone to communicate with each other and hence share discs and printers. This would also be the case if Ethernet bridges were used, but here there would be traffic segregation, which could improve performance if the nets were heavily loaded.

The use of the !Gateway application on each fileserver prevents Acorn Access discs and printers on the backbone from being utilised by the stations on the other networks. This retains the security of the *"Admin"* computers which are connected directly to the backbone. The same is also true of the computers on the remaining networks.

Printer services provided by !Spooler are accessible by all computers on the network, but the Acorn Access printers are restricted to the local network. The Application Accelerator is also unable to communicate to stations beyond the first !Gateway station it encounters.

!Manager

It is unwise to use !Manager at the machine which is running the fileserver software as this can cause unexpected results ***unless*** !Server has been quit from the icon bar first. Failure to do this can result in the changes which you have made to the ' Users' file being lost when !Server is next quit.

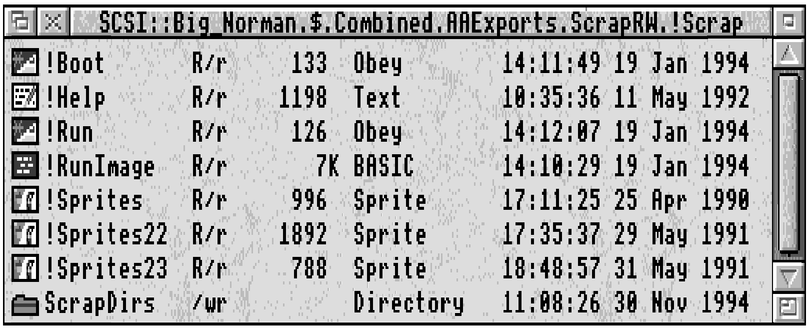
The reason for this is quite simple. When !Server is loaded into the machine it caches information about the ' Usersfile. When users make changes to this file, changing their password for example, the changes are made via the !Server application. !Server is therefore able to update its cache and the ' Usersfile as each change is made. When !Server is quit from the icon bar it saves the information stored in its cache to ensure that all the data which it accessed is in a stable and complete state. If !Manager is used on the machine which is running !Server it accesses the ' Usersfile directly rather than via !Server. This means that the copy of the ' Usersfile on the fileserver disc will be different to the cached copy which !Server is using. When !Server is quit it may overwrite the updated ' Usersfile with some or all of the old one, hence all the changes made via !Manger are lost.
It is for this reason that Acorn recommend that management tasks which have to be performed on a Level 4 Fileserver whilst !Server is running are done via !Manager on a remote terminal.
This comment also applies to management carried out using third party software, which may offer improved facilities to those in !Manager, especially concerning the handling of groups of users.

!Scrap

This is an essential application and must be accessible to all Acorn computers in order for them to function correctly. !Scrap is used to provide temporary file space for some operations such as printing. For this reason it ***must*** always be stored in a writable area.

On Level 4 Fileservers !Scrap is usually placed in each users URD (User Root Directory). This can result in many duplicate copies of the application taking up valuable disc space on the fileserver. The use of Acorn Access simplifies this as it makes it possible to have a single copy of !Scrap per network, which is accessed by all Acorn Access client stations. This greatly simplifies the management of this resource and results in a more effective use of the available disc space.

When implementing this on Acorn Access, all that is needed is to place !Scrap in the root of a protected Access disc (see later) and ensure that it has the following read/write access:



```
SCSI::Big_Norman.$.Combined.AAExports.ScrapRW.!Scrap
!Boot        R/r    133  Obey       14:11:49 19 Jan 1994
!Help        R/r   1198  Text       10:35:36 11 May 1992
!Run         R/r    126  Obey       14:12:07 19 Jan 1994
!RunImage    R/r     7K  BASIC      14:10:29 19 Jan 1994
!Sprites     R/r    996  Sprite     17:11:25 25 Apr 1990
!Sprites22   R/r   1892  Sprite     17:35:37 29 May 1991
!Sprites23   R/r    788  Sprite     18:48:57 31 May 1991
ScrapDirs    /wr         Directory  11:08:26 30 Nov 1994
```

Figure 2.2: Access permission settings for !Scrap on a protected Acorn Access disc.

Station numbering.

The station number of each machine on the network must be unique. The fileserver should, ideally, have a larger station number than the terminals it supports. Normally the fileserver is numbered 254 and it is recommended that the client terminals start from a minimum of 2 upwards. The reasons for this are, firstly a new network terminal always defaults to the number 1, hence, no Econet communication can take place from that machine if there is already a station 1 : AUN does not allow a station 1. Secondly, if a machine detects that the network is busy it will wait for a period of time before attempting to use the network again. If the machine is a Model B or a Master the period of time it would wait before attempting to access the network again would be determined by the station number. The bigger the number the less time there is between retries, hence the recommendation that fileservers have large station numbers. This feature was not implemented in RISC OS, but for organisational reasons it is still good practice on RISC OS machines. It is recommended where possible that client machines are numbered in increments of 10.

e.g. On the 1st network which is installed the numbering may be 10, 20, 30, 40 etc. Additional networks may then have the stations numbered 11, 21, 31, 41 etc and 12, 22, 32, 42 etc. This ensures that wherever a machine is placed on the network there will never be a station number clash. It may also be possible to use the room number to provide the least significant digit of the station numbers, making it easier to keep a track of machines.

A new version of the SetStation utility (Version 2.03) is available for setting the station numbers. This version performs improved CRC checks on the CMOS RAM and is compatable with all versions of RISC OS up to and including RISC OS 3.50. SetStation is a dangerous utility to have freely available on the network and so Acorn recommends that it is placed on all fileservers in $.Arthurlib and that its access permission is set to WR/.

Conclusion

The above sections cover some ideas on the theory of network server planning, structure and implementation. An actual implementation, based on field experience gained by the author and colleagues is presented in a separate application note and discs as a template for your use. This structure has been found to work in the field, and, once set up, to present only a small administrative load to the system manager. It has also been optimised to give good performance for network only stations, whilst retaining the ability to be customised to an individual' srequirements. Management of local hard discs on network stations is also covered - utilities which use the network to manage the discs.