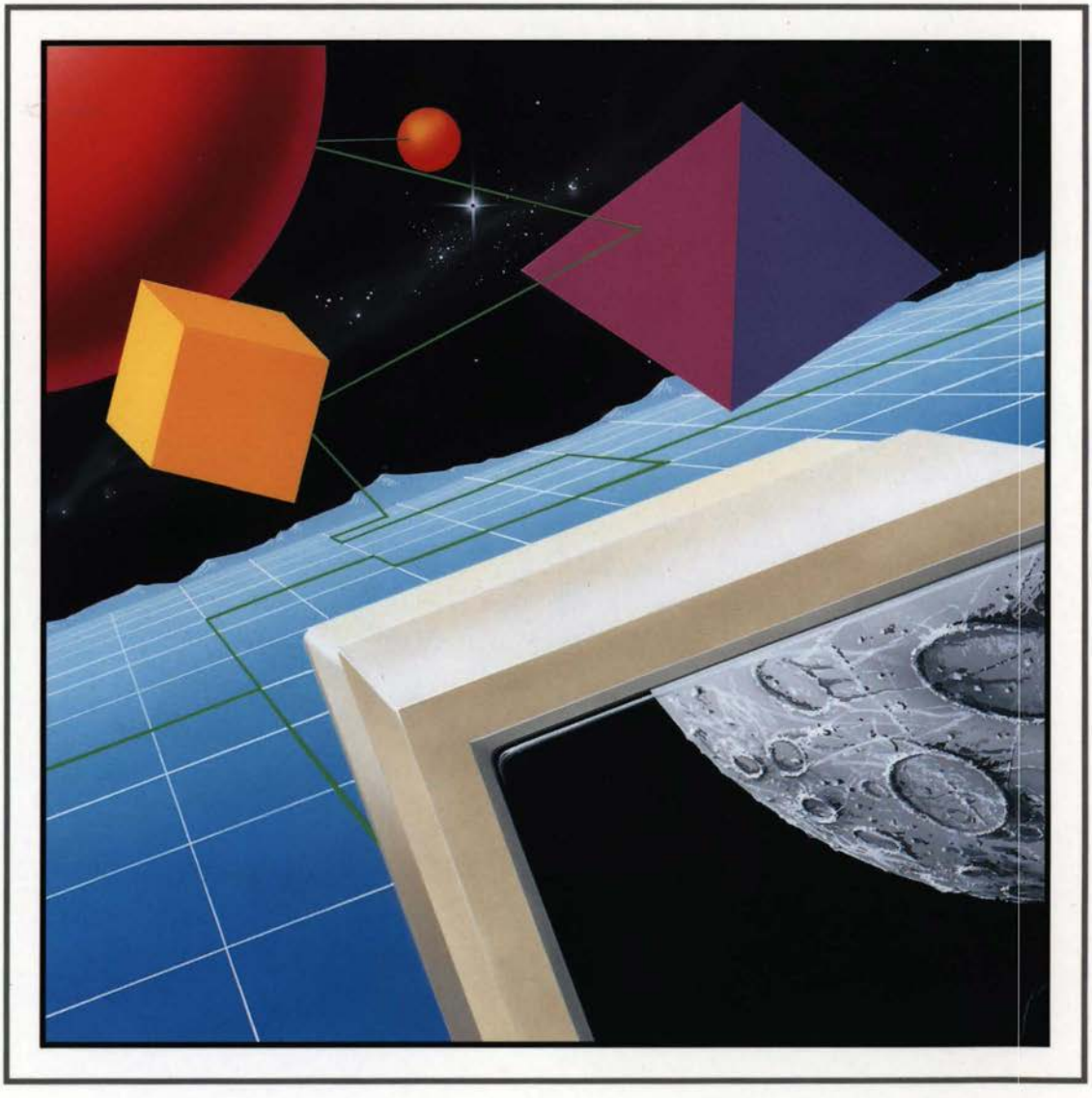
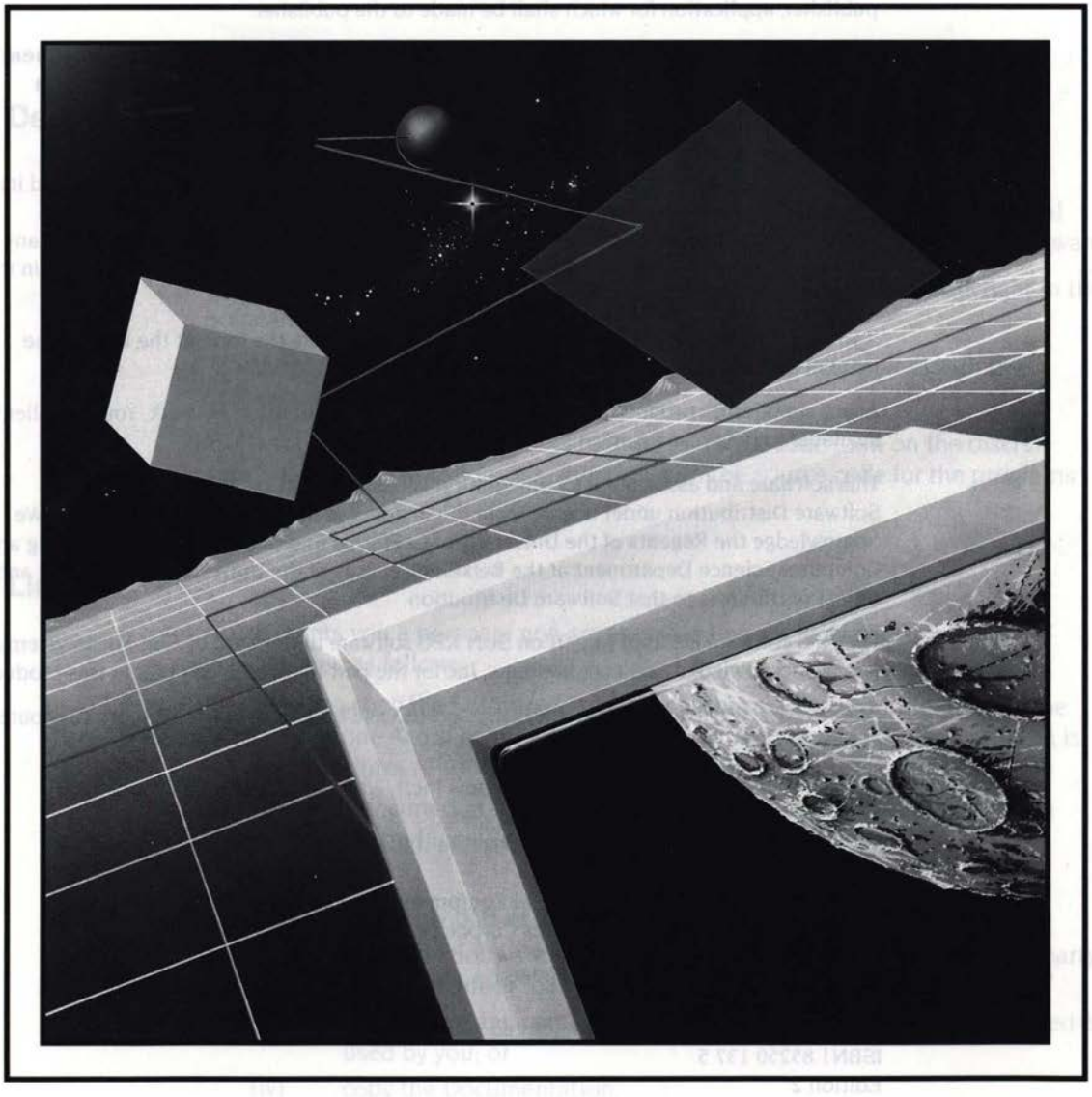
TCP/IP Protocol Suite (Release 2) Installation Guide



Acorn

TCP/IP Protocol Suite (Release 2)
Installation Guide



Acorn

Published by Acorn Computers Technical Publications Department.

The product described in this manual is not intended for use as a critical component in life support devices or any system in which failure could be expected to result in personal injury.

The product described in this manual is subject to continuous development and improvement. All information of a technical nature and particulars of the product and its use (including the information and particulars in this manual) are given by Acorn Computers Limited in good faith. However, Acorn Computers Limited cannot accept any liability for any loss or damage arising from the use of any information or particulars in this manual.

If you have any comments on this manual, please complete the form at the back of the manual and send it to the address given there.

Acorn supplies its products through an international distribution network. Your supplier is available to help resolve any queries you might have.

This software and associated documentation is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California. We acknowledge the Regents of the University of California and the Electrical Engineering and Computer Science Department at the Berkeley Campus of the University of California and other contributors to that Software Distribution.

The NFS software is based in part on SUN RPC software distributed by Sun Microsystems, Inc. We acknowledge Sun Microsystems, Inc for the part that software plays in this product.

Published by Acorn Computers Limited
ISBN1 85250 137 5
Edition 2
Part number 0484,182
Issue 1, April 1993

End-user licence conditions for TCP/IP Protocol Suite (Release 2)

1. Definitions

The following expressions have the meanings given here:

'Acorn' means Acorn Computers Limited, being either owner of all intellectual property rights in the Software, or having the right to grant licences of the Software.

'Developer' means any third party software developer who retains copyright in the Software.

'Documentation' means the printed user documentation supplied with the Software inside the pack.

'Software' means the programs contained in object-code form on the disc(s) supplied with these conditions, together with the source code for the programs MailMan, pcnfsd and m.send:

2. Licence

Acorn grants you a personal non-transferable non-exclusive licence (or sub-licence), as follows:

1 You may copy the Software for backup purposes, to support its use on one stand-alone Acorn computer system. (Separate provision for site licences is made on form APP157 available from your Acorn Authorised Dealer.)

2 You must ensure that the copyright notices contained in the Software are reproduced and included in any copy of the Software.

3 You may not:

(i) copy only part of the Software; or

(ii) make the Software or the Documentation available to any third party by way of gift or loan or hire;

(iii) incorporate any part of the Software into other programs developed or used by you; or

(iv) copy the Documentation.

3. Term

This licence remains in effect unless you terminate it:

1 by destroying the Software and all copies, and the Documentation, or

2 by failing to comply with the Conditions.

4. Limited warranty and disclaimer of liability

1 Acorn warrants the disc(s) upon which the Software is supplied to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of purchase, as evidenced by a copy of your receipt. Your Acorn Authorised Dealer will replace a defective disc if returned within ninety (90) days of purchase.

2 The Software is supplied 'as is'; neither Acorn nor the Developer makes any warranty, whether express or implied, as to the merchantability of the Software or its fitness for any particular purpose.

3 In no circumstances will Acorn be liable for any damage, loss of profits, goodwill or for any indirect or consequential loss arising out of your use of the Software, or inability to use the Software, even if Acorn has been advised of the possibility of such loss.

5. General

These conditions supersede any prior agreement, oral or written, between you and Acorn relating to the Software.

Contents

About this Guide

This Guide tells you how to install the RISC OS TCP/IP Protocol Suite (Release 2). There are five chapters:

- *Introduction to the* TCP/IP *Protocol Suite* (*Release* 2) introduces you to the important concepts behind TCP/IP, and explains the requirements of the software and how it interworks with other Acorn networking software.
- *Design and installation of your network* provides an introduction to Ethernet, outlines how to connect an Econet to an existing TCP/IP network, and gives hints on using other manufacturers' proprietary networks.
- *Installing the software on* RISC OS tells you the choices you need to make before installing the software; and how to a copy of the distribution software on all your RISC OS machines, changing it to include the information that is specific to each computer.
- *Setting up your* UNIX *network* outlines how to set up a UNIX network so that you can use this software with it.
- *Transferring system files to* UNIX tells you how, once you have got a working network, you can complete the installation by transferring some files from the distribution software to your UNIX computers.

There is also an appendix:

- *Using the serial port* gives guidance on how to make connections between a RISC OS computer's serial port and other equipment.

Finding out more

For general information on the use of a RISC OS computer and its desktop interface, see the *Welcome Guide* and RISC OS 3 *User Guide* supplied with it.

For details of how to use the TCP/IP Protocol Suite (Release 2), see the TCP/IP *Protocol Suite* (*Release* 2) *User Guide*.

For details of how to use the programming interfaces provided by the TCP/IP Protocol Suite (Release 2), see the TCP/IP *Protocol Suite* (*Release* 2) *Programmer's Guide*, available separately from your Acorn supplier. This includes a disc of useful C libraries.

You should also see any relevant documentation supplied with other computers you plan to be on your TCP/IP network.

Finally, you can get more detailed information from *Internetworking with* TCP/IP. Douglas Comer (1988) Prentice-Hall, Englewood Cliffs, NJ, USA.

Reader comments

If you have any comments on this Guide, please complete and return the reader comment form on the last page to the address given there.

Part 1 – Installation

1 Introduction to the TCP/IP Protocol Suite (Release 2)

The TCP/IP Protocol Suite (Release 2) enables you to connect Acorn RISC OS computers to a TCP/IP network, and to access computers on that network in a variety of ways.

You can use the TCP/IP Protocol Suite (Release 2) to connect RISC OS computers directly to an existing Ethernet-based TCP/IP network. To do so, your RISC OS computers will obviously need to have an Ethernet interface fitted.

As well as supporting the TCP/IP protocol over Ethernet, this software also supports TCP/IP over Econet. This means you can connect an entire existing Econet-based network to an Ethernet-based TCP/IP network. You need one or more computers with both types of interface, to join the two types of network together. These can be RISC OS or RISC *i*X computers. Such a machine is called a *gateway*.

TCP/IP concepts

When you install the TCP/IP Protocol Suite (Release 2), you will have to assign certain names and numbers to the computers on your TCP/IP network, and to their network interfaces. This section explains those names and numbers.

If you've already got a TCP/IP network running...

If you've already got a TCP/IP network running on your site, you should already have naming and numbering schemes set up. Make sure that any names and numbers you assign conform to this scheme, and that you first contact anyone who administrates their allocation.

Host names

Each computer on your network must have a *principal host name*, or *host name* for short. Your users will use this name to refer to the computer. The name must be unique on your site – you can't have two computers with the same name.

It helps your users if each host name is easy for them to remember. One way to do this is to use a theme, such as planets (eg `saturn`, `uranus`); another way is to give names that have some relationship to the computer's function on your

network (eg `accounts1`, `accounts2`). You can combine these ideas – so you might name the graphics department's computers after famous artists (eg `turner`, `vangogh`).

Interface names

Each network interface in each computer – whether it be an Ethernet or Econet interface – must also have an *interface name*. Again, this name must be unique on your site – you can't have two interfaces with the same name.

If there's only a single interface in a computer it's normal to use just the principal host name as the interface name. If there are two interfaces in a name it's normal to refer to the principal host name in each interface name: so a machine named `saturn` may have interfaces named `saturn_eco` and `saturn_ether`.

Internet addresses, netmasks and subnets

Furthermore, each interface must also have a unique numerical address, known as its *Internet address*. It is this address that the TCP/IP protocol uses to communicate; if a user specifies a host name or interface name, the software automatically converts it to an Internet address.

An Internet address is four bytes long. These four bytes are split into fields:

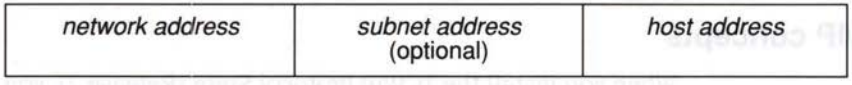


Figure 1.1 Fields within an Internet address

The *network address* identifies an entire network (which is typically a whole site). The *subnet address* is optional, and identifies a local network that forms part of the main network. The *host address* identifies a host on that network.

A *netmask* specifies the portion of the address used by the network and subnet addresses. For example, if the network address is held in the top byte, and no subnets are used, the netmask would be 0xFF000000 (i.e. FF000000 hexadecimal).

Unlike the interface name, the Internet address must be unique on all networks with which the interface will ever communicate.

If you plan to connect to other sites...

If you plan to connect to other sites over the Internet, you need to ensure not only that Internet addresses are unique to your site, but also that they are unique to the entire Internet. The Internet already connects together thousands of sites, each with many hosts. Clearly it's impossible to keep so many Internet addresses unique on an informal basis. Consequently there is an administrative body responsible for allocating network addresses. You must contact them before you use the Internet to connect to other sites; write or send email to:

DDN Network Information Center
SRI International
Room EJ217
333 Ravenswood Avenue
Menlo Park, CA94025
USA email: HOSTMASTER@SRI-NIC.ARPA

Depending on the size of your network, you will be allocated a Class A, B or C address: these use respectively the top one, two or three bytes for the network address. It is your responsibility how you use the remaining unallocated bytes to specify subnets and hosts. For example, let's say you've been allocated a Class B network address, and so have two bytes free for your own use:

- If all your site's computers are connected to a single local network, you won't need to use subnets, and so might use all two bytes for the host address (allowing 64k hosts). In this case, you'd use a netmask of 0xFFFF0000.

  Note that an Ethernet generally behaves as a single network, even if it is made up of multiple segments of cable.
- However, if your site's computers are connected to different local networks (such an Ethernet and some Econets), you'll need to use subnets. You might decide to use 5 bits for the subnet address (allowing 32 subnets), and the remaining 11 bits for the host address (allowing 2000 hosts). In this case, you'd therefore use a netmask of 0xFFFFF8000.

  Note that separate Econets (i.e. those not connected together by Econet bridges) form separate subnets. This is further explained in *Connecting an Econet to a* TCP/IP *network* on page 11.

If you don't plan to connect to other sites...

If you don't plan to connect to other sites over the Internet, all you need to ensure is that each interface's Internet address is unique on your own site. We suggest you use the following scheme:

network address	*host address* (high byte)	*host address* (middle byte)	*host address* (low byte)

Figure 1.2 Suggested local TCP/IP *numbering scheme*

Number your local networks from one: for example, you might number your Ethernet as net 1, and an Econet as net 2. Likewise, number your hosts (not your interfaces) from one. Your available Internet addresses and their meanings would then be:

Ethernet	**Meaning**	**Econet**	**Meaning**
1.0.0.1	host 1 on Ethernet	2.0.0.1	host 1 on Econet
1.0.0.2	host 2 on Ethernet	2.0.0.2	host 2 on Econet
1.0.0.3	host 3 on Ethernet	2.0.0.3	*and so on up to...*
1.0.0.255	host 255 on Ethernet	2.0.0.255	host 255 on Econet
1.0.1.0	host 256 on Ethernet	2.0.1.0	host 256 on Econet
1.0.1.1	host 257 on Ethernet	2.0.1.1	*and so on...*

Of course, if a machine has only got one interface fitted, you'll only use one of the addresses assigned to it; one of the addresses will be 'wasted'. But if you later upgrade the machine to add a second interface, you'll already have a meaningful Internet address reserved for it.

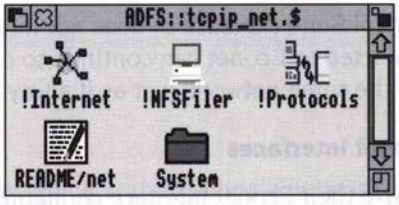
Physical addresses

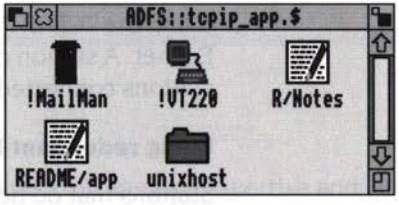
Each interface also has a six byte *physical address* (alternatively known as its MAC *address*). You shouldn't need to do anything to set this up, because:

- An Ethernet interface's physical address is unique worldwide, and is set in the hardware at the time of manufacture.
- An Econet interface's physical address is based on its network and station numbers. So long as you've correctly installed the Econet, these should be unique to your site.

Software

The software is supplied on two discs: the *Network* disc and the *Applications* disc.





The discs are organised to try and minimise the need for disc swapping when running the software from floppy disc, since both the NFS Filer and the Protocols applications require the Internet application to run. If you have a computer with a floppy disc drive that supports the new ADFS 1.6M F format, you will find it useful to combine both the distribution discs onto a single one. If your machine is an older one that will only support 800K discs, you might find it useful to make up discs with alternative combinations of application to those supplied.

The `R/Notes` file and the `README` files may contain information that is neither in this guide, nor in the TCP/IP *Protocol Suite* (*Release* 2) *User Guide*. Please ensure that you read these files.

The applications themselves are detailed in the TCP/IP *Protocol Suite* (*Release* 2) *User Guide*.

Machine requirements

You can use the TCP/IP Protocol Suite (Release 2) on any computer fitted with RISC OS 3 (version 3.10) or later, and that has at least 1 Mbyte of memory. If you want to run the VT220 application using the Ftp protocol you'll need a machine with 2 Mbytes or more of memory.

Coexistence with existing machines

Econet

Using the TCP/IP Protocol Suite (Release 2) does not interfere with normal use of Econet. A station connected to Econet can continue to communicate with other stations connected to the same network, just as it always has.

Using redundant Econet interfaces

Stations that do not have their Econet interface configured for TCP/IP may still use it as a native Econet interface. For example, a station with an TCP/IP-configured Ethernet interface may also have a non-TCP/IP-configured Econet interface, so that it can continue to access existing FileStores on an adjacent Econet.

AUN

Your Acorn machines – such as Level 4 FileServers – can now co-exist on the same network as other machines that use TCP/IP – such as an Ethernet to which UNIX workstations and NFS file servers are connected. You can follow this path by using the TCP/IP Protocol Suite (Release 2) in conjunction with its sister product, AUN.

AUN forms the core component of Acorn's new networking strategy, called *Acorn Universal Networking* (AUN). AUN uses the TCP/IP standard in such a way as to retain Econet's existing interfaces – both to users and to programs – so your users won't need to learn new skills, and your existing network programs should continue to work. AUN will work over your existing Econet network, but because it uses TCP/IP protocols, it will also work over an existing TCP/IP network. You can hence use AUN-configured Acorn machines connected to a TCP/IP network in exactly the same way as if they were connected to an Econet.

Using the TCP/IP Protocol Suite (Release 2) and AUN together is described in an application note, available from Acorn Customer Services.

2 Design and installation of your network

Designing and installing a network requires considerable expertise and knowledge, both technical and practical. We strongly recommend that you get an Acorn dealer to do this work for you. They have the necessary skills to ensure that your network delivers the best possible response across the site, avoiding any bad bottlenecks in the system.

This chapter is not a tutorial in network design, nor does it tell you how to install a network.

An introduction to Ethernet

One common use for the TCP/IP Protocol Suite (Release 2) is to connect RISC OS computers directly to an existing Ethernet-based TCP/IP network.

There are three types of Ethernet: *thick Ethernet* (also known as 10BASE5), *thin Ethernet* (also known as Cheapernet, or as 10BASE2), and *twisted-pair Ethernet* (also known as 10BASET). If you're connecting your computers to an existing network, you'll probably find it most convenient to use the same type of Ethernet as is already installed. However, you may have to choose which type of Ethernet to use, but have little previous experience of Ethernet to help you make a well-informed decision; the information in this section will give you some of the background you need.

Limitations

Each type of Ethernet has limitations. Some of these are theoretical limits beyond which the network cannot ever work; others are practical limits, beyond which performance degrades to an unacceptable level. The main differences are:

- Thick Ethernet cable segments can be up to 500m long, and may have up to 100 connections made to them. Connections must be at least 2.5m apart. The cable is thick, and so is relatively inflexible; its minimum radius of curvature is about 25cm. It is very resilient to physical damage and to electrical interference.

- Thin Ethernet cable segments can only be up to 185m long, and may only have up to 30 connections made to them, as close together as 0.5m apart. However, this is a theoretical maximum, and we recommend that to get acceptable performance you connect no more than 20 stations to a single segment.

  The cable is thin, and so is comparatively flexible.

- Twisted-pair Ethernet differs fundamentally from the other two types in that it is a *star network* rather than a *bus network*. This means that instead of there being a single length of cable to which all computers connect, each station must have its own length of cabling which plugs into a central *hub* box.

  Each cable may be up to 100m long; each hub can support up to 12 stations.

  The cable is thin and flexible, and is similar to telephone cables.

An Ethernet is entirely passive, and unlike Econet does not require a clock signal.

Speed

All types of Ethernet cable can transfer data at the same rate of 1.25 Mbytes/sec. Since this is much faster than the TCP/IP Protocol Suite (Release 2) is able to transfer data, a large number of stations can be simultaneously using the network before it slows down.

Cost

As with most things, you get what you pay for!

Cabling

The cost of Ethernet cabling varies:

- Thick Ethernet cabling is the most expensive of all, but it can support the greatest number of machines over the longest length, and is also the most rugged type.
- The cabling for thin Ethernet (as befits its alternative name of Cheapernet) is less expensive than that for thick Ethernet, but cannot support as many stations over as long a length of cabling as thick Ethernet can.
- Twisted-pair Ethernet cabling is the cheapest of all, but a hub device is required for every 8 - 12 stations, which adds to the cost.

The ruggedness and long maximum length of thick Ethernet makes it the most suitable cabling for outside, should you have to connect together buildings on a campus-style site. In general thin and twisted-pair Ethernet are more suited to indoor use, for which they are now the norm. Various ways of protecting the cable from damage – accidental or otherwise – are available from Ethernet suppliers.

Interfaces

Ethernet interfaces vary in capability, size and cost. Factors to consider are:

- Type(s) of cabling supported:
  Obviously your interface must be able to connect to the cabling you install! Some interfaces provide two types of socket, and you can choose which of them you use, giving you flexibility in your choice of cabling. Others may only provide a single socket, to reduce their cost and/or their size.
- Type of expansion slot required:
  Some interfaces are the right size for the expansion sockets on the backplane of larger computers (such as the A400 series or the A5000). Others are suitable for the internal expansion slot in smaller RISC OS computers (such as the A3000). There are also interfaces available for fitting to the network interface connector in more recent RISC OS computers (such as the A3010, A3020 and A4000), and adaptors for plugging into the parallel port of more recent RISC OS computers (in particular the A4, which does not have the expansion sockets required to fit any of the other types of interface just mentioned).

You should check with your Acorn supplier which types are currently available for RISC OS computers.

If you need to fit Ethernet interfaces to any other makes of computers you must use a card supplied or approved by the computer's manufacturer – don't try to use an Acorn card.

Connecting an Econet to a TCP/IP network

Apart from connecting RISC OS computers directly to an existing Ethernet-based TCP/IP network, the other main use for the TCP/IP Protocol Suite (Release 2) is to connect an entire existing Econet-based network to an Ethernet-based TCP/IP network. However, you must bear in mind that an Econet transfers data at a much slower rate than an Ethernet, and so cannot possibly give as good a performance.

To join an Econet-based network to an Ethernet-based TCP/IP network, you need one or more computers fitted with both types of interface, and which can therefore 'talk' to both types of network. These can be RISC OS or RISC *i*X computers. Each such machine is called a *gateway*. The section that follows gives guidance on adding gateways to an Econet, paying particular attention both to the *topologies* (or layouts) that you can use, and to avoiding bottlenecks in the system.

Topologies

Unbridged networks with a single gateway

About the simplest topology you can have is an Econet network connected by a single gateway to an Ethernet:

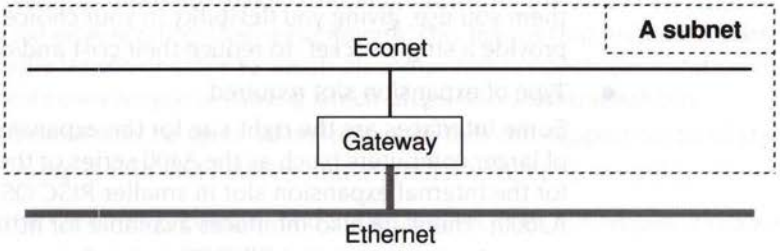


Figure 2.1 An Econet TCP/IP subnet

The Econet network is also a TCP/IP *subnet*. Its machines' Internet addresses (see *Internet addresses, netmasks and subnets* on page 4) must all use the same unique subnet number.

When a network packet goes through a gateway there are inevitable delays as it's merged with the existing traffic on the other side. Furthermore, if too many network packets are using the same gateway to cross from the Ethernet to Econet, you can get a backlog of packets building up. This is because the Ethernet is a faster network, and can potentially deliver packets to the gateway more quickly than they can be placed on the Econet.

Unbridged networks with multiple gateways

If you find a gateway is consistently overloaded, you can add extra gateways. For example:

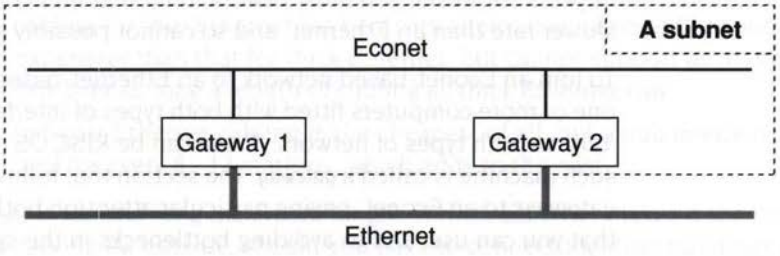


Figure 2.2 An Econet TCP/IP subnet with two gateways

You can then configure the TCP/IP Protocol Suite (Release 2) so that different computers on the Econet use different gateways to reach the Ethernet. In the example above, you'd probably get half the stations to use Gateway 1, and the

other half to use Gateway 2. (This procedure is explained in *Installing the software on* RISC OS on page 17, and in the configuration files you'll need to edit.) Likewise, you'd need to configure half your computers connected to the Ethernet to use Gateway 1 to reach the Econet, and the other half to use Gateway 2, thus ensuring that reverse traffic is also evenly balanced between the two gateways. See your other computers' manuals for further details.

Bridged networks with a single gateway

Your existing Econet may use *bridges* to link together two or more Econet networks. TCP/IP traffic can cross a bridge from one Econet network to another, and so connected Econet networks can share a single gateway. They still form part of the same TCP/IP subnet:

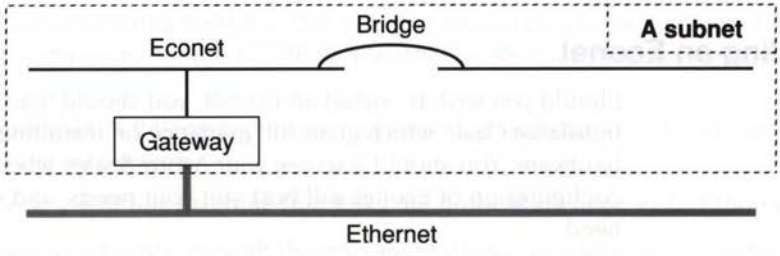


Figure 2.3 A bridged Econet TCP/IP subnet

Bridged networks with multiple gateways

Just as with gateways, network packets are delayed as they cross a bridge and are merged with the existing traffic on the other side. So, the fewer Econet bridges a network packet has to cross before reaching the Ethernet, the quicker it will reach a machine on the Ethernet. (This is, of course, also true of packets going in the reverse direction.) You can avoid such delays by putting a gateway on each Econet network. For example:

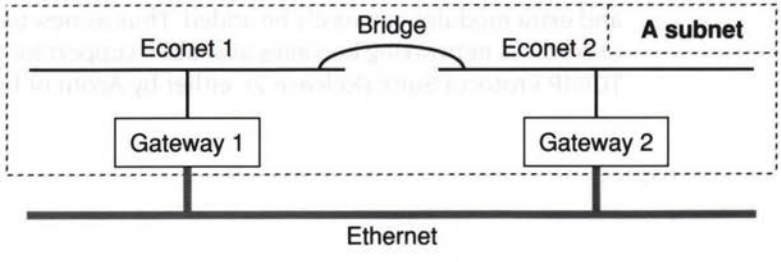


Figure 2.4 A bridged Econet TCP/IP subnet with two gateways

You'll then need to ensure that all traffic to and from stations on Econet network 1 uses Gateway 1, and that network 2 traffic uses Gateway 2. This is a similar process to that used for an unbridged Econet network that has multiple gateways: see page 12 for further details.

Summary

In general, the more gateways you provide, the better performance will be; although there will come a point where you'll have so many gateways that they're almost never overloaded. It's hard to give more precise guidelines, because of the wide range in how network-intensive computer usage is at different sites. If you're already running a network, you'll have a good feel for how things work out at your site.

Installing an Econet

Should you wish to install an Econet, you should read the *Econet Design and Installation Guide*, which gives full guidance on installing and maintaining Econet hardware. You should also see your Acorn dealer, who can advise you which configuration of Econet will best suit your needs, and supply any hardware you need.

You will probably find the AUN/*Level* 4 *Fileserver* (*Release* 2) a particularly useful product. It contains both the AUN software (described on page 8), and also software to use RISC OS computers as file and print servers. The manuals for this product – the AUN *Manager's Guide* and the *Level* 4 *Fileserver Network Manager's Guide* – are available separately, should you require them.

Future developments in networking technologies

Only a very small part of the TCP/IP Protocol Suite (Release 2) handles hardware interfaces. The software for each type of interface is held in its own *driver module*, and extra modules can easily be added. Thus as new technology such as fibre optic or cordless networking becomes available, support for it can easily be added to the TCP/IP Protocol Suite (Release 2), either by Acorn or by a third party.

Cabling

You must use the correct standard of cabling for the type of network you choose. Resist the temptation to install a cheaper grade of cable which is superficially similar to the specified grade, as your network will not work properly. For example, you mustn't try to run Ethernet over coaxial cable designed for use with TV aerials, even though it may appear similar to the coaxial cable used for thick Ethernet.

Again, your Acorn dealer can advise you about the correct specification of cable to use.

Installing other proprietary networks

You may wish to network some of your UNIX computers using other manufacturer's proprietary networking systems. This is a very similar case to using Econet for RISC OS computers; for the TCP/IP Protocol Suite (Release 2) to work properly you must:

- ensure that TCP/IP protocols are implemented and supported over the proprietary network
- provide a gateway machine that links the proprietary network to Ethernet.

If you have any doubts, consult the documentation supplied with your other computers or ask your supplier.

3 Installing the software on RISC OS

In the earlier chapter *Introduction to the* TCP/IP *Protocol Suite* (*Release* 2) on page 3 we outlined different ways you could set up the TCP/IP Protocol Suite (Release 2). You'll have to edit some configuration files to do so. This chapter tells you how to make those changes and install the software. It assumes you are using the desktop, and are familiar with simple use of it. If you have any problems refer to the RISC OS *User Guide* supplied with your RISC OS computer.

The configuration files

The configuration files you need to edit are held within the Internet application. Rather than refer to them all the time by their lengthy full pathnames, we'll just use the leafname. The files are supplied on the *Network* distribution disc as:

Filename	Leafname	
`$.!Internet.!Configure`	`!Configure`	
`$.!Internet.files.hosts`	`hosts`	databases
`$.!Internet.files.networks`	`networks`	
`$.!Internet.files.protocols`	`protocols`	
`$.!Internet.files.services`	`services`	
`$.!Internet.files.startup`	`startup`	

The `hosts`, `networks`, `protocols` and `services` files are collectively known as the *databases*.

What the files do

- The `!Configure` file does most of the configuration of the software. It sets the principal host name of a computer. It configures each interface, setting their Internet addresses, their netmasks, and the driver modules to be used. It defines where to find the other configuration files – one location for the *databases*; and another location for the `startup` file. It specifies whether the RouteD module should be run to establish routing information. Finally, it sets whether or not the station will forward packets between multiple interfaces, and hence whether it acts as a TCP/IP gateway.

- The `hosts` file gives the host names and Internet addresses of all the computers you wish to refer to by their host name.
- The `networks`, `protocols` and `services` files contain databases of network, protocol and service names. These files are unused by the TCP/IP Protocol Suite (Release 2), and are provided to support any extra software that uses TCP/IP protocols.

  The files specify the default values normally used on all computers that support the TCP/IP protocols; consequently, you shouldn't ever need to edit them. If you do, you should see respectively the UNIX *networks* (5), *protocols* (5), and *services* (5) manual pages.
- The `startup` file initialises your computer's interface(s), and also establishes routes to remote networks or hosts – if you're not using RouteD to do so.

If you have access to a file server...

By default, both the databases and the `startup` file are kept on each RISC OS computer's local disc. However, if your computers have access to a native Econet file server they can instead read them over the net. Using central copies of these files will make it much easier for you to maintain them; for example when you want to add the host name of a new computer to the `hosts` database.

You can, if necessary, make different RISC OS computers access different databases, either on the same file server, or on different ones. In practice it's unlikely you'll need to as these files are usually identical for all machines on a network.

Again, you can, if necessary, make different RISC OS computers access different files of routing information, either on the same file server, or on different ones. For instance, you might have several gateways from your subnet to an Ethernet subnet; you could force different computers to use different gateways, and so spread the load.

There are detailed explanations of how to implement all of the above in the `!Configure` file.

Note that you cannot use an AUN file server in this manner.

Different ways to configure the software

There are some decisions you have to make on how to configure the software. This section outlines what those decisions are. For details of how to configure the choices you make, see *Installing the* TCP/IP *Protocol Suite* (*Release* 2) on page 20, and the instructions and examples in the files you'll need to edit.

Ways to set the Internet address of each interface

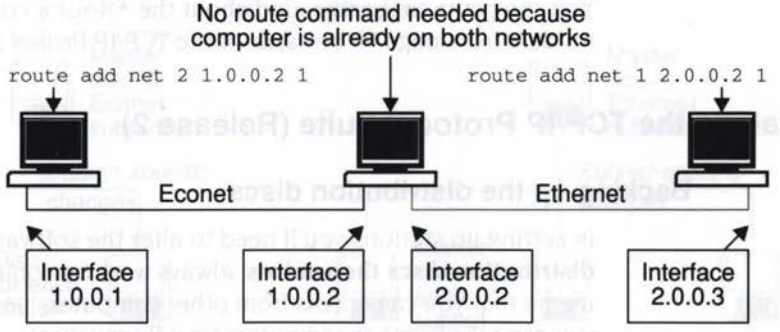
You can set the Internet address of each interface in three ways:

- you can set it explicitly on the computer to which it's fitted
- you can use the interface name to look up the Internet address in the `hosts` database
- you can use the physical address to look up the Internet address using *Reverse* ARP exchanges with an ARP *server* (see *Setting up an* ARP *server, if required* on page 25).

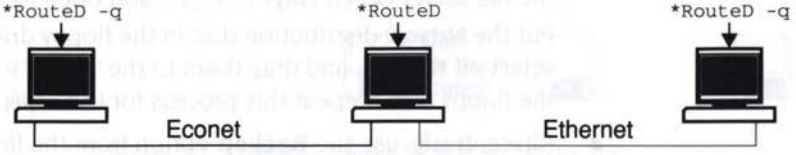
Ways to establish routing information

If you are using gateways, you need to ensure that each RISC OS computer knows their location, and the route to subnets other than the one to which it is connected. There are three ways you can do this:

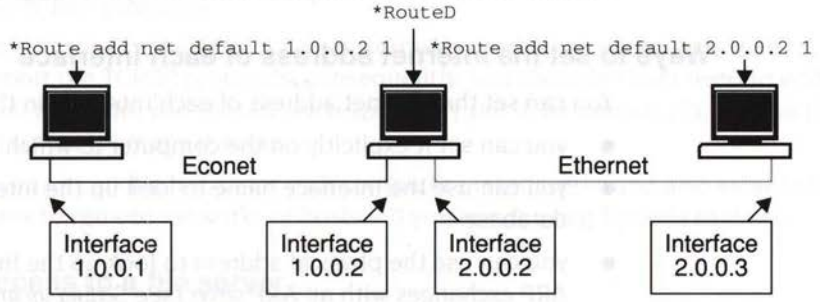
- Use the `*Route` command to explicitly define the routing of each gateway. For example:



- Use the `RouteD` module to perform the routing for you. For example:



- Use a hybrid of the above two methods. For stations with a single interface, use the `*Route` command to define the location of a gateway. Use the `RouteD` module on each gateway to establish the route to other networks and gateways. For example:



The `RouteD` module uses the Routing Information Protocol (or RIP) to communicate with other computers that implement RIP – such as other RISC OS gateways running `RouteD`, or UNIX computers running the `routed` daemon – and hence to establish routes. This generates a lot of network traffic. We recommend that you only run `RouteD` if you have a very complex network on which other computers use RIP. There are no real advantages to using `RouteD` on a simple network, where it degrades performance unnecessarily; we recommend that you instead use `*Route` commands, as outlined in the first example above.

You can get more background about the `*Route` command and the `RouteD` module from page 90 onwards of the TCP/IP *Protocol Suite (Release 2) User Guide*.

Installing the TCP/IP Protocol Suite (Release 2)

Backing up the distribution discs

In setting up stations you'll need to alter the software. **You must not alter the distribution discs themselves**; always work on copies of them, made either on media that you can access from other computers (ie an Acorn file server) or that you can take to other computers (ie a floppy disc):

- If your RISC OS computers can access an Acorn file server, make a directory on the file server called (say) `TCP_IP`, and open its directory display.

  Put the *Network* distribution disc in the floppy drive, open its directory display, select all the files and drag them to the `TCP_IP` directory display. Dismount the floppy disc. Repeat this process for the *Applications* distribution disc.
- Alternatively, use the **Backup** option from the floppy disc drive's icon bar menu to back up the two distribution discs to another pair of floppy discs.

Where you can install the software

When you actually install the software for a particular station, you can do so on any media you like, such as a hard disc, floppy discs or a remote file server. All that is important is that you preserve the directory structure that is on the distribution discs– things that are in the same directory must stay together.

- If you're installing the software on a hard disc or an Acorn file server, we suggest you make a directory called (say) `TCP_IP` to hold the software. This directory can be anywhere you like – it needn't be in the root directory.
- You don't have to do this, and may prefer to put the software in a directory that already contains other applications.

Avoiding too much work

The instructions that follow tell you how to alter the software to set up a single station. In practice, you'll find yourself repeating many of the changes for every station on your site, or on a particular subnet. We suggest that rather than repeating the same work, you create several 'levels' of 'masters'. For example, if you were installing from floppy discs onto a single Econet subnet and several Ethernet subnets, you might follow this scheme:

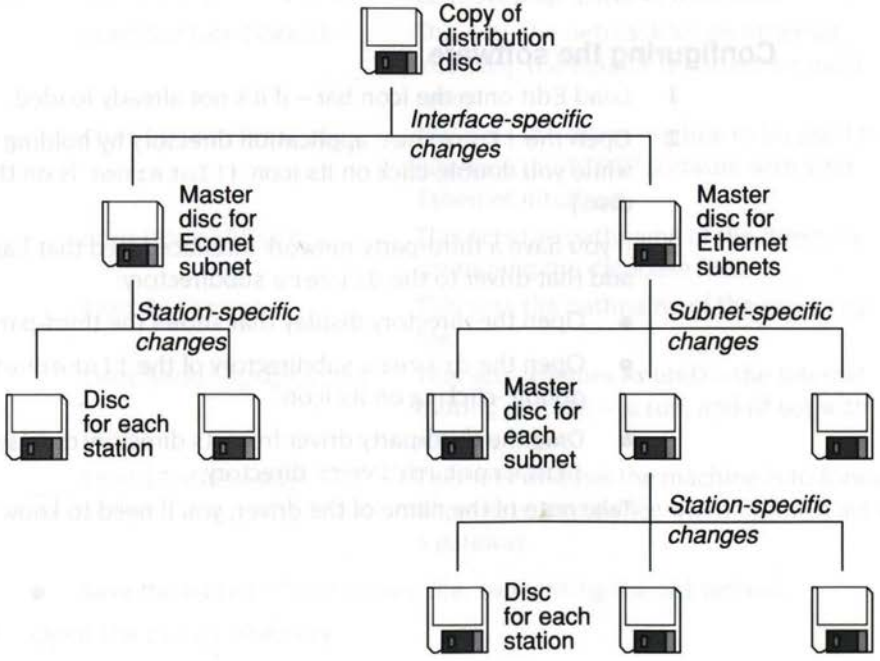


Figure 3.1 Example scheme for making master discs

- You would take two copies of the distribution disc, and make any changes you need to make to cater for the different types of interface. This would give you two master discs: one for Ethernet stations, and one for Econet stations.
- You would then take the master disc for stations on the Econet subnet, and alter copies of that to give you discs suitable for each individual station.
- Likewise, you would take the master disc for Ethernet stations, and alter copies of that. However, this is done in two stages: one to make the changes necessary for each subnet, and one to make the changes for each station.

We suggest that you use a similar scheme for your site. It's best if you read through this chapter, and then devise a plan suited to your site, trying to make as little work for yourself as possible. If you can see a way of only making a change once, do so.

Naming and labelling discs

As you can see from the above example, you can easily end up with many subtly different master copies of the software, particularly if all your stations install the software from floppy disc. It's important that you can distinguish between these. Give each master an appropriate name, based say on the subnet number or host name by which it will be used. To name floppy discs, choose **Name disc** from the floppy disc drive's icon bar menu; you should also label each disc.

Configuring the software

1. Load Edit onto the icon bar – if it's not already loaded.
2. Open the `!Internet` application directory by holding down the Shift key while you double-click on its icon. (`!Internet` is on the *Network* distribution disc.)
3. If you have a third-party network interface fitted that has a disc-based driver, add that driver to the `drivers` subdirectory:
   - Open the directory display that shows the third-party driver.
   - Open the `drivers` subdirectory of the `!Internet` application by double-clicking on its icon.
   - Drag the third-party driver from its directory display to the `!Internet.drivers` directory.

   Take note of the name of the driver; you'll need to know this for the next step.

4 Edit the `!Configure` file:

- Load it into Edit by dragging its icon to the Edit icon on the icon bar.
- Following the instructions in the file, edit the lines that set the system variables:

Variable	Notes
`Inet$HostName`	This sets a station's principal host name.
`Inet$EcoIPAddr`	This sets whether your station has an Econet interface that you wish to configure for use with TCP/IP, and (if so) sets its Internet address, or specifies that it be looked up using Reverse ARP.
`Inet$EcoIPMask`	This sets the netmask for an Econet interface; the default is correct for most cases.
`Inet$EtherIPAddr`	This sets whether your station has an Ethernet interface that you wish to configure for use with TCP/IP, and (if so) sets its Internet address, or specifies that it be looked up using Reverse ARP.
`Inet$EtherIPMask`	This sets the netmask for an Ethernet interface; the default is correct for most cases.
`Inet$EtherDevice`	This sets the driver module to be used to interface the TCP/IP software with your Ethernet interface.
`InetDBase$Path`	This sets the pathname of the directory containing the database files.
`Inet$Startup`	This sets the pathname of the `startup` file.
`Inet$RouteDOptions`	This sets whether RouteD – the Internet routing module – is run, and (if so) with what options.
`Inet$IsGateway`	This sets whether the machine is to forward IP packets, and hence whether it will act as a gateway.

- Save the edited `!Configure` file, overwriting the old version.

5 Open the `files` directory.

6 Edit the `hosts` file:

- Load it into Edit by dragging its icon to the Edit icon on the icon bar.
- Edit the file so that it contains the Internet addresses and host names of all the RISC OS computers you wish to refer to by host name. (If a computer's host name isn't in here you'll instead have to use its Internet address whenever you want to communicate with it.)
- If you want to add any UNIX hosts to the file, their addresses and names must match those in your UNIX `/etc/hosts` files. For a small number of hosts you'll probably find it best just to print out and copy a UNIX `hosts` file. For a larger number of hosts, you might find it better to copy across your UNIX `hosts` file later. We'll tell you how to do this in the section *Transferring the hosts file from* UNIX on page 41.

If you need help, see the comments in the supplied `hosts` file, and the UNIX *hosts* (5) manual page.

Then save the edited `hosts` file:

- If you did not change the value of `InetDBase$Path` in the `!Configure` file, just overwrite the file you loaded.
- Otherwise, save it to the new location you set up when you edited the `!Configure` file. Copy the other databases (the `networks`, `protocols`, and `services` files) to the same new location. Delete the old databases from the `files` directory.

7 If you are using gateways but are not using RouteD, edit the `startup` file:

- Load it into Edit by dragging its icon to the Edit icon on the icon bar.
- Add `*Route` commands to the file so that it contains details of all the gateways you will use.

For guidance and examples see the comments in the file, and the documentation of the *Route command on page 90 of the TCP/IP *Protocol Suite (Release 2) User Guide*.

Then save the edited `startup` file:

- If you did not change the value of `Inet$Startup` in the `!Configure` file, just overwrite the file you loaded.
- Otherwise, save it to the new location you set up when you edited the `!Configure` file. Delete the old version of the file from the `files` directory. If there's nothing left in the `files` directory, delete it too.

8 Finally, remove write permission from the files so that your work won't be undone. If you've been using floppy discs, you may also like to write protect them.

Updating the !System directory

You must also ensure that some system resources on the RISC OS computer are sufficiently up to date:

1. Open the `System` directory. (The `System` directory is on the *Network* distribution disc.)
2. Open its `Modules` subdirectory.
3. Open the computer's `!System` directory (as supplied on the RISC OS *Applications Disc*) by holding down the Shift key while you double-click on its icon.
4. Open its `Modules` subdirectory.
5. Choose **Full Info** from the Filer's **Display** submenu so you can see how old the modules are in both `Modules` directories.
6. Copy each of the modules supplied with the TCP/IP Protocol Suite (Release 2) to the computer's `!System` directory if either of these is true:
   - the same module is already in `!System`, but it's older than the version of it supplied with the TCP/IP Protocol Suite (Release 2)
   - there isn't yet a copy of the module in `!System`.

Once you've copied any necessary modules, you can delete the `System` directory from the installed software.

Setting up an ARP server, if required

If you want to use Reverse ARP exchanges to map physical addresses to Internet addresses you'll need to set up an ARP server. The machine doing this can be a RISC OS or a UNIX computer, and can do so over either Ethernet or Econet.

If you've already got a UNIX ARP server running, the most sensible thing to do is to add entries for your RISC OS machines to the database it uses; see *Using an existing server* on page 32.

If you need to set up a RISC OS ARP server it must *publish* its entries. Use the command:

```
*ARP -f filename
```

in a boot file.

You can find the physical address(es) of a RISC OS computer's interface(s) using the relevant `*EnInfo` command; see page 81 of the TCP/IP *Protocol Suite (Release 2) User Guide* for further details. You may find it helpful to know that the physical address of an Econet interface is:

`00.00.00.00.`*`station_number.net_number`*

For more details, see the documentation of the *ARP command on page 78 of the TCP/IP *Protocol Suite (Release 2) User Guide*.

If you instead decide to set up a UNIX ARP server, see *Setting up a new server* on page 32.

You've now finished installing the TCP/IP Protocol Suite (Release 2) on RISC OS.

Advanced installation

If you have a good knowledge of RISC OS system variables, you will be able to see the wide range of ways you can set up the TCP/IP Protocol Suite (Release 2). In the above text, and in the comments in the configuration files, we've laid out several different ways which should suit most possible installations – but you may be able to see a way that is better suited to your site. If you do, and if you know what you're doing, by all means further adapt the configuration files. Just make sure you're not working on your master copy of the distribution discs!

4 Setting up your UNIX network

Before you can use the TCP/IP Protocol Suite (Release 2) to communicate between your RISC OS computers and UNIX computers, you will need to ensure that your UNIX computers are correctly networked.

What this chapter tells you

This chapter tells you which parts of a standard UNIX network installation you need for the TCP/IP Protocol Suite (Release 2) to work correctly. It also tells you which UNIX manual pages will give you more information if you need it.

- You can either look these pages up in the manuals supplied with your computer, or use the UNIX `man` command.

What this chapter doesn't tell you

What this chapter doesn't tell you is exactly how to set up a complete UNIX network.

- We don't supply full information because there can be many subtle differences between one UNIX computer and another, especially if they're running different versions of UNIX (such as *System* V.4 and 4.3 BSD). If you do need this information you should see the manuals supplied with your UNIX computer(s), and follow up any cross references we give. You'll then get information that is accurate, and specific to your own UNIX computer.

  Of course, if you're an experienced system administrator with a network already running, our checklist may well be adequate on its own.

Filenames

In the pages that follow we give filenames both for processes you need to have running and for configuration files you need to edit. These are likely to be correct for systems based on 4.3 BSD (such as RISC iX), but may differ for other UNIX systems. If you have any doubts, see the UNIX manual pages we refer you to. Remember, **filenames are case-sensitive in UNIX**.

Install your UNIX network

The first thing that you need to do is to install the network hardware and software for your UNIX computers. You'll need to refer to the manuals supplied with them for help on how to do this.

Becoming root

To make most of the necessary changes to your UNIX computer you'll have to first become *root* (or the *super-user*). Use the command:

```
su root
```

typing in the root password when prompted – see the *su* (1) manual page.

Daemons you need to have running

Once you have set up your UNIX network, you need to check that each computer offers all the services that the TCP/IP Protocol Suite (Release 2) needs. These are the daemons that need to be running or available:

Process	Notes (opposite)	Typical filename	UNIX manual page
portmap	*	/etc/portmap	portmap (8c)
nfsd	*	/etc/nfsd	nfsd (8)
sendmail	*	/usr/lib/sendmail	sendmail (8)
inetd	*	/etc/inetd	inetd (8)
cron	*	/usr/etc/cron	cron (8)
mountd	†	/usr/etc/rpc.mountd	mountd (8c)
tftpd	†	/etc/tftpd	tftpd (8c)
telnetd	†	/etc/telnetd	telnetd (8c)
ftpd	†	/etc/ftpd	ftpd (8c)
pcnfsd	‡	—	—

Notes

* These daemons are typically started at boot time by `rc` file(s) within the `/etc` directory – see the *rc* (8) manual page. You can check if the daemons are running by using the `ps` command. If they're not, you'll need to:

- Use the relevant manual page to check where the file(s) are stored on your computer (they may differ from the above).
- Make sure the file(s) are on the computer. (If they aren't, read them from your distribution tape or discs. If you can't find a file there either, contact your UNIX supplier and ask for a copy.)
- Edit the relevant `rc` file(s) in the `/etc` directory so that they run the missing daemons.

† These daemons are typically started by the `inetd` daemon when the corresponding service is needed. A configuration file (typically `/etc/servers` or `/etc/inetd.conf`) controls this. You'll need to check this file, and edit it if it's not starting the daemons. See the *inetd* (8), *servers* (5) and/or *inetd.conf* (5) manual pages for more details.

You may find that the default network configuration for your machine instead runs one or more of these daemons all the time. You can find out either by using the `ps` command to see if the daemon is running, or by examining your `rc` file(s) in the `/etc` directory. If your machine's manufacturer has chosen to do things this way, don't change it – there'll be a good reason for doing so, such as improving network performance.

‡ This daemon needs to be installed from the TCP/IP Protocol Suite (Release 2)'s *Application* disc. You won't be able to do so until the end of the installation process; see the chapter *Transferring system files to* UNIX on page 33 for instructions.

You only need the pcnfsd daemon on UNIX computers you intend to use as RISC OS NFS name servers or print servers. We list it here so you have a complete list of all the processes that might need to be running or available.

Add the RISC OS Internet addresses

The next stage is to set up your UNIX network so that each computer knows the Internet addresses of any new computers that you are adding – such as RISC OS computers running the TCP/IP Protocol Suite (Release 2). There are two ways you can do this for UNIX computers:

- Edit the `/etc/hosts` file on each computer so that it includes all the new Internet addresses. See the *hosts* (5) manual page.
- Use a network database system to keep track of host names, such as Sun's *Network Information Services* (which used to be known as *Yellow Pages*). See the documentation supplied with your UNIX computer.

  (Note that you cannot use this method for RISC OS computers.)

Set up any RISC *iX* Econet-Ethernet gateways

You may have decided to use a RISC *i*X computer as a gateway between an Econet and the Ethernet. To do so, it will obviously need to have an Econet interface; these are available from your Acorn Network Dealer, complete with fitting instructions.

You will also need to configure the Econet interface to be available, by editing the `/etc/rc.net` file, adding an `econetup` and an `ifconfig` command. You must pass the parameters `-trailers` and `arp` to the `ifconfig` command. See your RISC *i*X documentation, especially the RISC *i*X *System Administrator's Guide* (which has full examples), and also the *econetup* (8) and *ifconfig* (8) manual pages. If you still have trouble contact your RISC *i*X supplier.

By the way, you don't need to pass the `-trailers` and `arp` parameters to the equivalent RISC OS *IfConfig command.

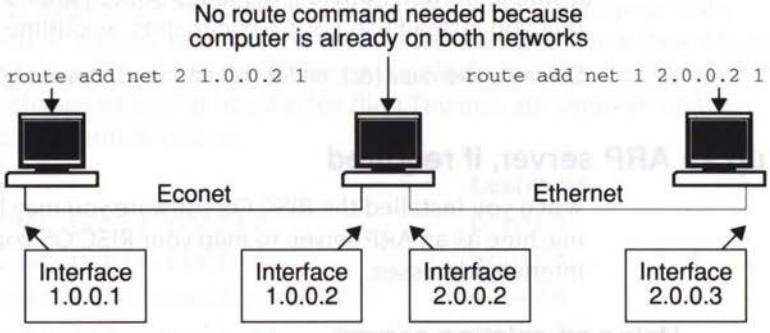
Internet packet forwarding on RISC *iX*

All current versions of RISC *i*X (up to 1.21c) are configured by default to forward Internet packets, so the computer can act as an Econet-Ethernet gateway. To comply with various standards, future versions may by default be configured not to forward packets. Should this be the case, you will have to build a reconfigured kernel for any gateway machines, setting the variable `ipforwarding` to 1; the RISC *i*X documentation will tell you how to do so.

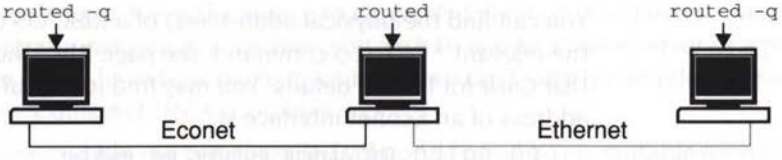
Add gateway routing information

Next you need to make sure that each UNIX computer knows of any gateways you will be using. Again, there are three ways you can do this:

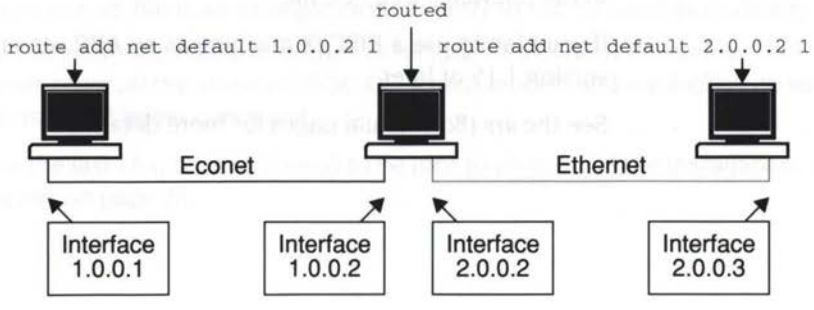
- Use the `route` command to explicitly define the routing of each gateway in the relevant `/etc/rc` file. For example:



- Use the `routed` daemon to perform the routing for you. You will need to edit the relevant `/etc/rc` file to start the daemon. For example:



- Use a hybrid of the above two methods. For stations with a single interface, use the `route` command to define the location of a gateway. Use the `routed` daemon on each gateway to establish the route to other networks and gateways. You will need to edit the relevant `/etc/rc` file to use the `route` command, or to start the `routed` daemon. For example:



The `routed` daemon uses the Routing Information Protocol (or RIP) to communicate with other computers that implement RIP – such as other UNIX computers running the `routed` daemon, or RISC OS gateways running `RouteD` – and hence to establish routes. This generates a lot of network traffic. We recommend that you only run `routed` if you have a very complex network on which other computers use RIP. There are no real advantages to using `routed` on a simple network, where it degrades performance unnecessarily; we recommend that you instead use `route` commands, as outlined in the first example above.

See also the *route* (8*c*), *routed* (8*c*) and *rc* (8) manual pages.

Set up an ARP server, if required

When you installed the RISC OS software you may have decided to use a UNIX machine as an ARP server, to map your RISC OS computers' physical addresses to Internet addresses.

Using an existing server

If you've already got a UNIX ARP server running, the most sensible thing to do is to add entries for your RISC OS machines to the database it uses.

You can find the physical address(es) of a RISC OS computer's interface(s) using the relevant `*EnInfo` command; see page 81 of the TCP/IP *Protocol Suite* (*Release* 2) *User Guide* for further details. You may find it helpful to know that the physical address of an Econet interface is:

`00.00.00.00.`*station_number*`.`*net_number*

Setting up a new server

If you need to set up a UNIX ARP server it must *publish* its entries. Use the command:

`arp -f` *filename*

in the relevant `/etc/rc` file.

If you plan to use a RISC *i*X machine as an ARP server, it must be running RISC *i*X version 1.15 or later.

See the *arp* (8*c*) manual pages for more details.

5 Transferring system files to UNIX

The final stage of the installation is to transfer some files to your UNIX computers. It's only now that you've installed enough of the software to do this transfer. Again, we'll refer to these files by their leafnames. (Note that some of these will change when you transfer the file.) The files are supplied on the *Applications* distribution disc as:

Filename	Leafname
`$.unixhost.pcnfs.pcnfsd`	`pcnfsd`
`$.unixhost.pcnfs.pcnfsd/c`	`pcnfsd/c`
`$.unixhost.riscosmail.m/send`	`m/send`
`$.unixhost.riscosmail.m/send/c`	`m/send/c`
`$.unixhost.riscosmail.crontab`	`crontab`
`$.unixhost.termcap`	`termcap`

- The `pcnfsd/c` file is the source to the pcnfsd daemon. You need to have this daemon running on any machine you wish to use as a name server or a print server. You'll have to compile it yourself – we can't supply compiled versions for every different UNIX computer.
- The `pcnfsd` file is a pre-compiled version of `pcnfsd/c`, suitable for running **on RISC *i*X computers only**.
- The `m/send` file is the source to the m.send program. You need to have this on any NFS host on which users of the NFS MailMan have their home directory. Again, you'll have to compile this yourself.
- The `m/send` file is a pre-compiled version of `m/send/c`, suitable for running **on RISC *i*X computers only**.
- The `crontab` file is an example crontab entry to run `m/send` periodically.
- The `termcap` file contains entries to include in a UNIX termcap file. The entries cover all the standard RISC OS screen modes, and are looked up when you use the *Telnet command.

Just as in the last chapter, you'll need to be root to do most of the installation; see *Becoming root* on page 28.

Before transferring the files

Before you transfer the files, you need to create a directory on your UNIX machine. This is used to hold the compiled version of `m.send`:

```
mkdir /usr/lib/riscosmail
```

See the *mkdir* (1) manual page.

Ways of making the transfer

There are four ways you can transfer the files from the distribution discs to a UNIX machine:

- reading the discs directly (RISC *i*X only)
- using NFS, either from the desktop or from the command line
- using Ftp
- using *Tftp.

If you are transferring to a RISC *i*X computer...

All RISC *i*X computers can read ADFS discs, and you may find this is the easiest way to transfer the files. See your RISC *i*X documentation for details of how to do this.

- Transfer `termcap` and `crontab` to somewhere you can temporarily store them – such as the `/tmp` directory. Make sure you don't overwrite the `/etc/termcap` or `/usr/lib/crontab` files.
- Transfer `m/send` rather then `m/send/c` – you'll save yourself the trouble of compiling the program. Put it in the directory `/usr/lib/riscosmail`, and rename it as `m.send`; we'll refer to it by that name from now on.
- Transfer `pcnfsd` – likewise avoiding having to compile `pcnfsd/c`. We recommend that you put this in `/usr/sbin` or (if you have an older version of RISC *i*X which doesn't have this directory) in `/usr/etc`.

If you are transferring to a UNIX computer...

For other UNIX computers, you'll have to use either NFS, Ftp, or the *Tftp command. See the notes below and the TCP/IP *Protocol Suite* (*Release* 2) *User Guide* for details of how to do this.

- Transfer `termcap`, `pcnfsd/c`, `m/send/c` and `crontab` to somewhere you can temporarily store them – such as the `/tmp` directory. (The leafnames of `m/send/c` and of `pcnfsd/c` are translated to `m.send.c` and `pcnfsd.c` respectively, which are the names we'll use from now on.) Make sure you don't overwrite the `/etc/termcap` or `/usr/lib/crontab` files.

Notes

You may have problems transferring these files if you haven't yet put the host name of the destination UNIX computer into your RISC OS `hosts` file. You'll have to do one of the following:

- use the UNIX computer's Internet address rather than its host name
- add its host name and Internet address to your RISC OS `hosts` file before you access it.

If you're using NFS you must use the user name `nobody` – normally this doesn't have a password set. This gives you access to a UNIX computer, but as a totally unprivileged user. You may have a problem finding a directory that you can write to; we suggest you try `/tmp` and the home directories for the users `guest` and `demo` (if they exist). As a last resort, log on to the computer as `root` and set the permissions on a directory so you can write to it.

Once you've transferred, compiled and run `pcnfsd` on one computer, you'll be able to use it as a name server, so you can then use other user names with NFS.

If you're using Ftp or *Tftp you can transfer the file using any valid user name and password.

Once you've transferred the files

Now that you've transferred the files you need to:

- Set up pcnfsd by compiling `pcnfsd.c`, ensuring that it has the correct permissions and ownership, and by editing one of your startup files (such as `/etc/rc.local`) to run the pcnfsd daemon.
- Set up the mail system by compiling `m.send.c` and saving it in `/usr/lib/riscosmail`, ensuring they have the correct permissions and ownership, and by adding the contents of the `crontab` file you transferred to the UNIX `/usr/lib/crontab` file.
- Set up *Telnet by adding the contents of the `termcap` file you transferred to the UNIX `/etc/termcap` file.

You may then wish to transfer the UNIX hosts file to your RISC OS computers, if you chose to do this earlier (see *Installing the software on* RISC OS on page 17).

Finally you'll need to repeat the changes for all your other UNIX computers, and to re-boot them.

See below for full descriptions.

Setting up pcnfsd

Background

Running pcnfsd makes a machine act as a *name server*, that can be used by the NFS Filer to authenticate a username/password pair before making a mount. The mount need not necessarily be on the name server.

Running pcnfsd also turns a machine into a *print server*, to which the RISC OS Printers application can submit files as NFS print jobs. Pcnfsd places each submitted file in a subdirectory of its own spool directory (which by default is `/usr/spool/pcnfs`), creating the directories if necessary. It then uses the UNIX lpr command to print the files. If you need more detail than this, see *More about printing with pcnfsd* on page 37.

Compiling pcnfsd

If you are using a RISC *i*X machine, and you transferred `pcnfsd` rather than `pcnfsd.c`, you can ignore this section – read on from *Checking permissions and ownership* below.

You must use the UNIX `cc` command to compile `pcnfsd.c`, leaving you with code for the pcnfsd daemon that will run on your UNIX computer. You should put the compiled code with any other similar daemons that your computer uses. The example below assumes that the source was in `/tmp`, and that the code is being placed in `/usr/etc`; you may use different directories:

```
cc /tmp/pcnfsd.c -o /usr/etc/pcnfsd
```

If you don't want to use the default spool directory (`/usr/spool/pcnfs` – see above) you can specify a different one when compiling:

```
cc -DSPOOLDIR=spool_dir /tmp/pcnfsd.c -o /usr/etc/pcnfsd
```

Some versions of UNIX may give you an error message from the compilation, such as:

```
"/usr/include/sys/tty.h", line 12: syntax error at or
      near variable name "queue_t"
```

If this happens to you, load `pcnfsd.c` into a text editor and find the line:

```
/*#include <sys/stream.h>*/
```

Remove the comments, thus:

```
#include <sys/stream.h>
```

Then compile your edited version.

Once you've successfully compiled `pcnfsd.c` you can remove the source code, which you no longer need:

```
rm /tmp/pcnfsd.c
```

Checking permissions and ownership

You must make sure that `pcnfsd` is owned by root, and has the permission 'rwxr–xr–x'. To check this use the command:

```
ls -l filename
```

(where *filename* is the full UNIX pathname of `pcnfsd`). To set the ownership to root, use the command:

```
chown root filename
```

and to set the permission use the command:

```
chmod 755 filename
```

See the *ls* (1), *chown* (8) and *chmod* (1) manual pages.

Editing your startup file

Finally you need to edit one of the startup files that your computer reads at boot time, such as `/etc/rc.local` – see the *rc* (8) manual page. Using existing entries as a guide, add an entry to start the pcnfsd daemon. You must start it sometime **after** you start the portmap daemon.

If you wish pcnfsd to use a different spooling directory to the one set at compile time (`/usr/spool/pcnfs` by default), you can specify the new spool directory using pcnfsd's '`-s spooldir`' option.

More about printing with pcnfsd

This section gives more technical information that you may need if you have problems printing with pcnfsd. In the vast majority of cases you can ignore it, and go on to read *Setting up the mail system* on page 39.

When pcnfsd first gets run, it creates a spool directory *spooldir* on the print server. By default, this is `/usr/spool/pcnfs`. If this is unsuccessful, you will get the error 'pc-nfsd: invalid spool directory *spooldir*'.

When a print job is initialised from a RISC OS client named *printclient*, pcnfsd:

1. Sets the spool directory's permission to `rwxrwxrwx`.
2. Creates a subdirectory named *printclient*. If this is unsuccessful, you will get the error 'pc-nfsd: unable to create spool directory *spooldir/printclient*'.
3. Sets the subdirectory's permission to `rwxrwxrwx`.

The RISC OS client then spools the file to be printed into this subdirectory, and tells pcnfsd to start the print job. Then, pcnfsd:

1. Checks the file exists, contains data, and is not already being printed.
2. Forks, and sets the user and group ids of the child process to those of the user requesting the print.
3. Submits the print job from the child process by execing lpr with the –r and –s options.

   This puts on the printer queue a symbolic link to the spooled file (rather than a copy of it), hence saving disc usage. Both the link and the spooled file are removed when the print job finishes; should this not happen, see *Undeleted print spool files* below.

Incidentally, the liberal access permissions on the spool directories are necessary for printing using the –s option.

Undeleted print spool files

You may find that files you send from RISC OS to a print server do not get deleted after they have been printed, but instead remain in the directory structure beneath pcnfsd's spool directory. The most likely explanation is that the version of lpd on the print server is one that – for security reasons – does not support lpr's –s option. To work around this you will have to edit the source for pcnfsd, removing the lines that specify the –s option, so:

```
execlp("/usr/ucb/lpr",
       "lpr",
       "-s",
       "-r",
```

becomes:

```
execlp("/usr/ucb/lpr",
       "lpr",
       "-r",
```

Then recompile the source, and re-boot your print server so that it runs the new version of pcnfsd.

Note that in making this change you will double the disc space used for spooling RISC OS print jobs: as well as the copy beneath pcnfsd's spool directory, while the file is printing there will be another copy in the printer's spool directory.

Setting up the mail system

Background

The `m.send` program transfers mail between users' private mail queues set up by the MailMan application, and the UNIX mail system. RISC OS users can hence send and receive mail via an NFS host.

The program reads `/usr/lib/riscosmail/m.list` to find who on the computer is using NFS MailMan. It then transfers their incoming mail from `/usr/spool/mail` to their `Mail/Intray` files, and pipes any outgoing mail from their `Mail/Pending` directories into the UNIX `/usr/lib/sendmail` program.

Compiling m.send

If you are using a RISC *iX* machine, and you transferred `m.send` rather than `m.send.c`, you can ignore this section – read on from *Checking permissions and ownership* below.

Read the comments at the start of the `m.send.c` source, and make any necessary changes; then compile it using the UNIX cc command. The example below assumes that you transferred the source to the `/tmp` directory:

```
cc /tmp/m.send.c -o /usr/lib/riscosmail/m.send
```

Just as with `pcnfsd.c`, you may get some errors when compiling, and will need to uncomment a line to include an extra header. This line is clearly indicated in the source.

Checking permissions and ownership

You must make sure that `/usr/lib/riscosmail` and `m.send` are owned by root, and have the permission 'rwxr–xr–x'. To check this use the command:

```
ls -l filename
```

(where *filename* is the full UNIX pathname of either `/usr/lib/riscosmail` or `m.send`). To set the ownership to root, use the command:

```
chown root filename
```

and to set the permission use the command:

```
chmod 755 filename
```

See the *ls* (1), *chown* (8) and *chmod* (1) manual pages.

Setting up each user

For each user:

- Choose a UNIX host for them to use, which can act as an NFS server.
- Give them an account and a home directory (which we refer to as `~user`) on the UNIX host.
- Create the file `~user/Mail/Intray` and the directory `~user/Mail/Pending` (remembering UNIX is case-sensitive).
- Add their user name to the file `/usr/lib/riscosmail/m.list`, which is a list of all the users on the UNIX host using the NFS MailMan. There should be one user to each line.

Editing /usr/lib/crontab

Finally you need to edit the UNIX `/usr/lib/crontab` file to include the contents of the `crontab` file you transferred. This periodically runs `m.send`, which transfers mail between users' private mail queues and the UNIX mail system. The numbers specify the minutes past the hour when the command is run. So if you wanted `m.send` to be run every ten minutes, you could change the line to read:

```
0,10,20,30,40,50 * * * * root /usr/lib/riscosmail/m.send
```

See also the UNIX *crontab* (5) manual page.

Removing lock files

The mail system creates temporary files in each user's `~user/Mail` directory. These are:

File	Use
`~user/Mail/M_ACTIVE`	A lock file created when the m.send program or the MailMan application is active (i.e. reading or writing) in the directory.
`~user/Mail/M_CTMP`	A temporary work file created by the MailMan application to construct outgoing mail messages.

`M_ACTIVE` prevents simultaneous access to files within the `~user/Mail` directory by m.send and MailMan. If, because of a system crash or some other failure, this file is not removed by m.send or MailMan, you should remove it yourself, as its presence may block further mail traffic.

`M_CTMP` will not cause such problems, and anyway is automatically removed each time MailMan starts.

NFS file extensions

By default, when saving files of type Text from RISC OS to NFS, no extension is added to their name. If you change this behaviour by editing the `extensions` file, the supplied versions of `m.send` will no longer work. This is because certain files created by MailMan will no longer have the names expected by `m.send`: these files are *~user*`/Mail/M_ACTIVE`, and *~user*`/Mail/Pending/00` to `09`.

To fix this, you will either need to stop using extensions when saving Text files from RISC OS to NFS, or you will have to appropriately modify `m.send.c`, and recompile it. We recommend the former.

Setting up *Telnet

Next you need to edit the UNIX `/etc/termcap` file to include the contents of the `termcap` file you transferred. This provides termcap entries for when you use `*Telnet` from the RISC OS command line. You will get the best performance if you put the entries after the ones your UNIX computer uses, but before any other entries you do not normally use. See the UNIX *termcap* (5) manual page.

Transferring the hosts file from UNIX

If your UNIX network has a lot of hosts, and consequently a large `/etc/hosts` file, you may have put off adding their entries to your RISC OS `hosts` file (as we advised you earlier in the chapter *Installing the software on* RISC OS). The last thing you might want to do is to use either NFS, Ftp or *Tftp to copy a UNIX `/etc/hosts` file to RISC OS, so you can easily merge it with your RISC OS `hosts` file – using Edit, for example.

By the way, you can now use any valid user name with NFS, because you should have installed `pcnfsd` on a computer, making it a name server.

Transferring files to other UNIX computers

Now make the same changes to all your other UNIX computers. You may find it easier to use your UNIX network to copy the files you've just set up, rather than re-install from RISC OS. Just remember if you're using different models of computer that files such as `/etc/termcap` may vary, and also that you will need to recompile both `pcnfsd/c` and `m.send.c`.

Re-booting your UNIX computers

Now re-boot your UNIX computers, to make sure that they start all the necessary daemons. You've now set up everything you need to so you can use the TCP/IP Protocol Suite (Release 2).

Part 2 – Appendixes

Appendix A: Using the serial port

This appendix gives a summary of how to connect equipment to the serial port on a RISC OS computer. You will find this information particularly useful if you wish to use the Serial protocol module with the VT220 terminal emulator.

Cables

The serial port supports two different types of equipment: *Data Terminal Equipment*, or DTE (for example printers and computers) and *Data Communication Equipment*, or DCE (for example modems). It follows that you will need different cables depending on the equipment you wish to connect.

When you make or purchase cables you must ensure that they use screened cabling.

DSR and CTS

RISC OS computers use the DSR (Data Set Ready) signal to determine whether to transmit data, rather than the CTS (Clear To Send) signal more commonly used. When you connect other equipment to a RISC OS computer's serial port consult the instructions its manufacturer has supplied, but note that:

- **at your RISC OS computer's end of the cable only** connections to the CTS signal should instead be made to the DSR signal.

The following sections give some guidelines which will work with most hardware. Some or all of the connections shown with dotted lines may be necessary, and you must first check the manuals for the hardware you are connecting.

Pin assignment

This diagram shows the assignment of the pins on a serial port plug that is to be connected to a RISC OS computer, viewed from the side that is to be soldered:

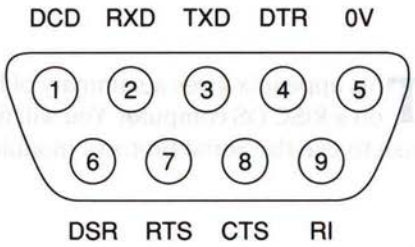


Figure 5.1 Archimedes serial port pin assignment: view from rear of computer, or plug from side to be soldered to cable

The pin assignment of 9-pin serial ports on other equipment is often the same as above.

Connecting two RISC OS computers

The RISC OS computers function as DTEs. Make the following connections:

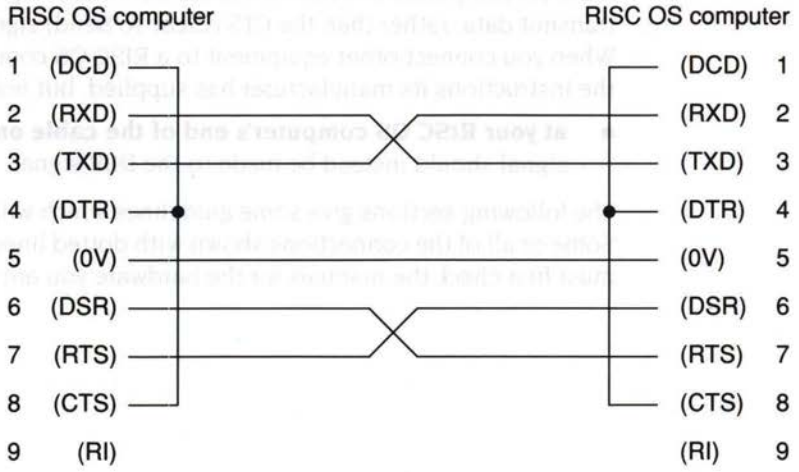


Figure 5.2 Serial cable to connect two RISC OS computers

Connecting to another computer

In this case both your RISC OS computer and the other computer function as DTEs. Make the following connections:

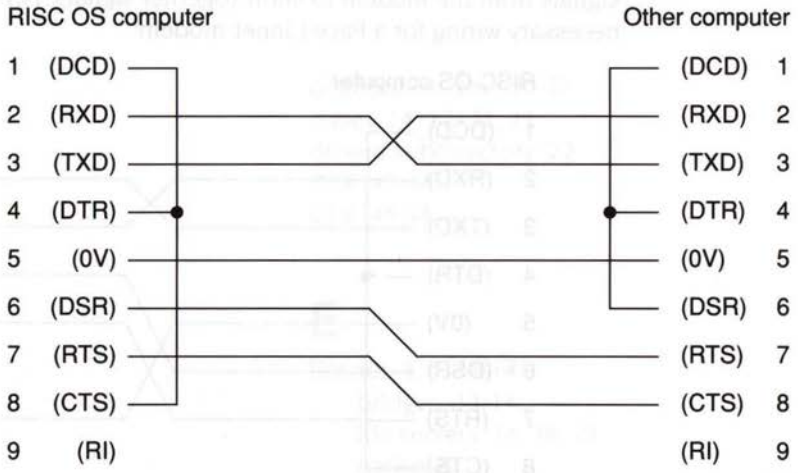


Figure 5.3 Serial cable to connect a RISC OS computer to another computer

Connecting to a modem

In this case your RISC OS computer functions as a DTE and the modem as a DCE. There is more variation between the connections needed for modems than there is for other types of hardware, and you should read your modem's manual:

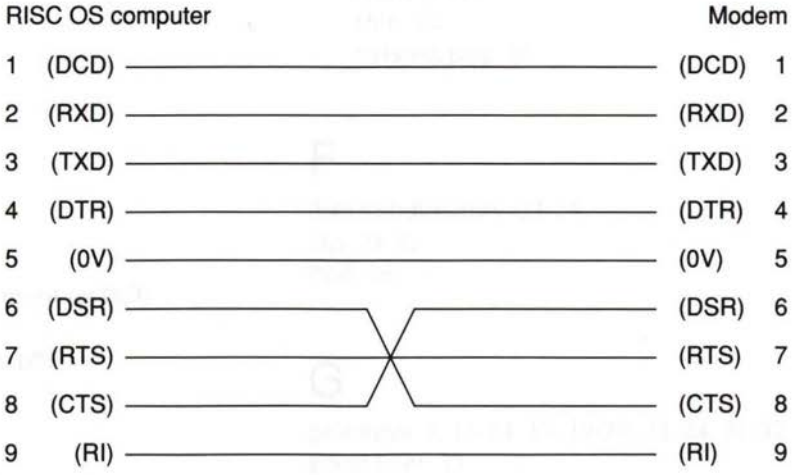


Figure 5.4 Serial cable to connect a RISC OS computer to a modem

A specific example of connecting to a modem

The above diagram showed connections for a typical modem. Different connections may be necessary for actual modems, especially the 'loopback' of signals from the modem to short together signals. For example, this is the necessary wiring for a Pace Linnet modem:

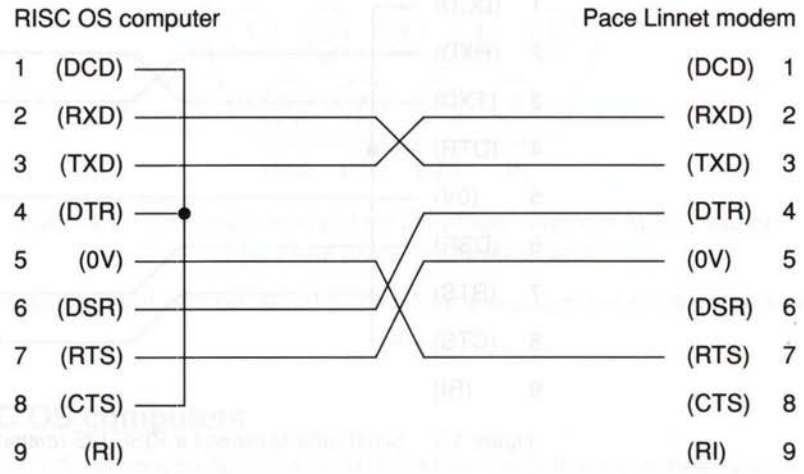


Figure 5.5 Serial cable to connect a RISC OS *computer to a Pace Linnet modem*

Index

H

I

L

M

N

P

R

S

T

Y

Reader's Comment Form

TCP/IP *Protocol Suite (Release 2) Installation Guide*

We would greatly appreciate your comments about this Guide, which will be taken into account for the next issue:

Did you find the information you wanted?

Do you like the way the information is presented?

General comments:

If there is not enough room for your comments, please continue overleaf

How would you classify your experience with computers?

☐ **First-time user** ☐ **Used computers before** ☐ **Experienced User** ☐ **Programmer**

Cut out (or photocopy) and post to:

Dept RC, Technical Publications
Acorn Computers Limited
Acorn House
Vision Park
Histon
Cambridge CB4 4AE

Your name and address:

This information will only be used to get in touch with you in case we wish to explore your comments further