Econet_EnumerateMap (SWI &4001E)

Enumerates subnetwork addresses within an AUN site network

On entry

R0 = flags:
all bits reserved (must be 0)
R4 = enumeration reference (0 to start)

On exit

R0 preserved
R1 = net number
R2 = pointer to net name, or 0
R3 = IP subnetwork address
R4 = next enumeration reference, or –1 if no more

Interrupts

Interrupt status is unaltered
Fast interrupts are enabled

Processor mode

Processor is in SVC mode

Re-entrancy

Not defined

Use

This call enumerates subnetwork addresses within an AUN site network. It returns the AUN net names, net numbers and IP addresses of the subnetworks active within an AUN site network, as derived from the Map file located within an AUN gateway.

If R4 is –1 on exit then all subnetworks have been enumerated, and R1 - R3 are undefined. If R4 is –1 on exit from the first call then the calling application is running over a network containing no AUN gateways.

Under native Econet R4 is always returned as –1.

Econet_EnumerateTransmit (SWI &4001F)

Returns the handles of open TxCBs

On entry

R0 = index (1 to start with first transmit block)

On exit

R0 = handle, or 0 if no more transmit blocks

Interrupts

Interrupt status is unaltered
Fast interrupts are enabled

Processor mode

Processor is in SVC mode

Re-entrancy

Not defined

Use

This call returns the handles of open TxCBs. On entry R0 is the index of the TxCB being asked for (1, 2, 3, etc). If the value of R0 is greater than the number of open TxCBs, then the value returned as the handle will be 0, which is an invalid handle.

You should not make this call from an IRQ or event routine as, although it will not fail, the returned information may be inaccurate.

Related SWIs

Econet_StartTransmit (page 2-657), Econet_PollTransmit (page 2-659), Econet_AbandonTransmit (page 2-660)

Related vectors

None

Econet_HardwareAddresses (SWI &40020)

Returns the addresses of the Econet hardware and interrupt control registers

On entry

—

On exit

R0 = address of MC68B54 ADLC
R1 = address of FIQ mask register
R2 = bit mask value to use on the FIQ mask register

Interrupts

Interrupt status is unaltered
Fast interrupts are enabled

Processor mode

Processor is in SVC mode

Re-entrancy

SWI is re-entrant

Use

This call returns the addresses of the Econet hardware and interrupt control registers. It is provided for the internal working of Econet diagnostic software, and is not intended for any other use.

Related SWIs

None

Related vectors

None

108 AUN

Introduction

The AUN software that this chapter describes forms the core component of Acorn's new networking strategy, called *Acorn Universal Networking* (AUN). AUN uses an industry standard method of passing data over a network: a family of protocols called TCP/IP.

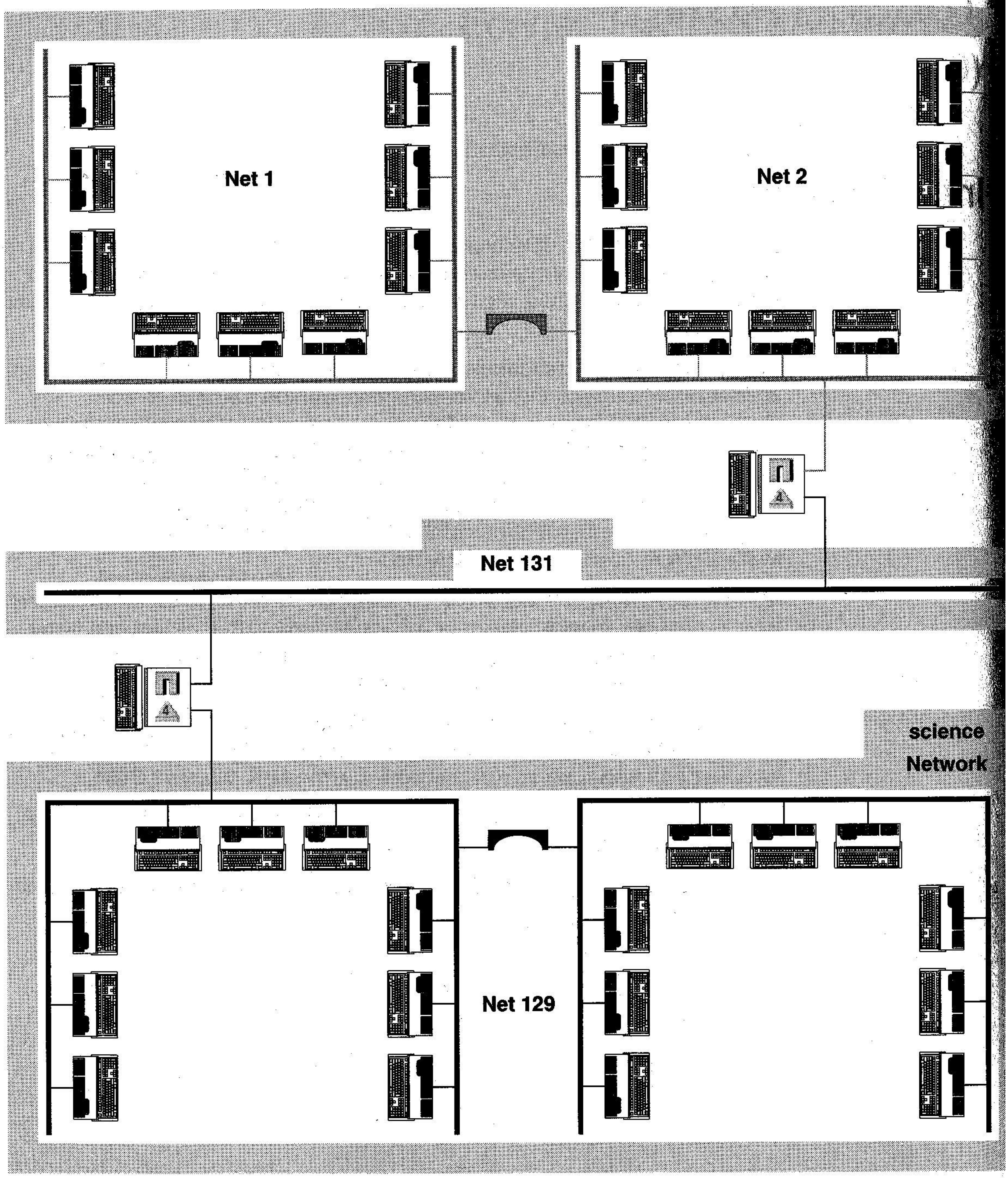
AUN uses the TCP/IP standard in such a way as to retain Econet's existing program interfaces, so your existing network programs should continue to work. Furthermore, AUN's use of the TCP/IP standard supports the concept of Open Systems. Acorn machines – such as Level 4 FileServers – can now co-exist on the same network as other machines that use TCP/IP – such as UNIX workstations and NFS file servers. You can follow this path by using AUN in conjunction with its sister product, the TCP/IP Protocol Suite; this is described in an application note, available from Acorn Customer Services.

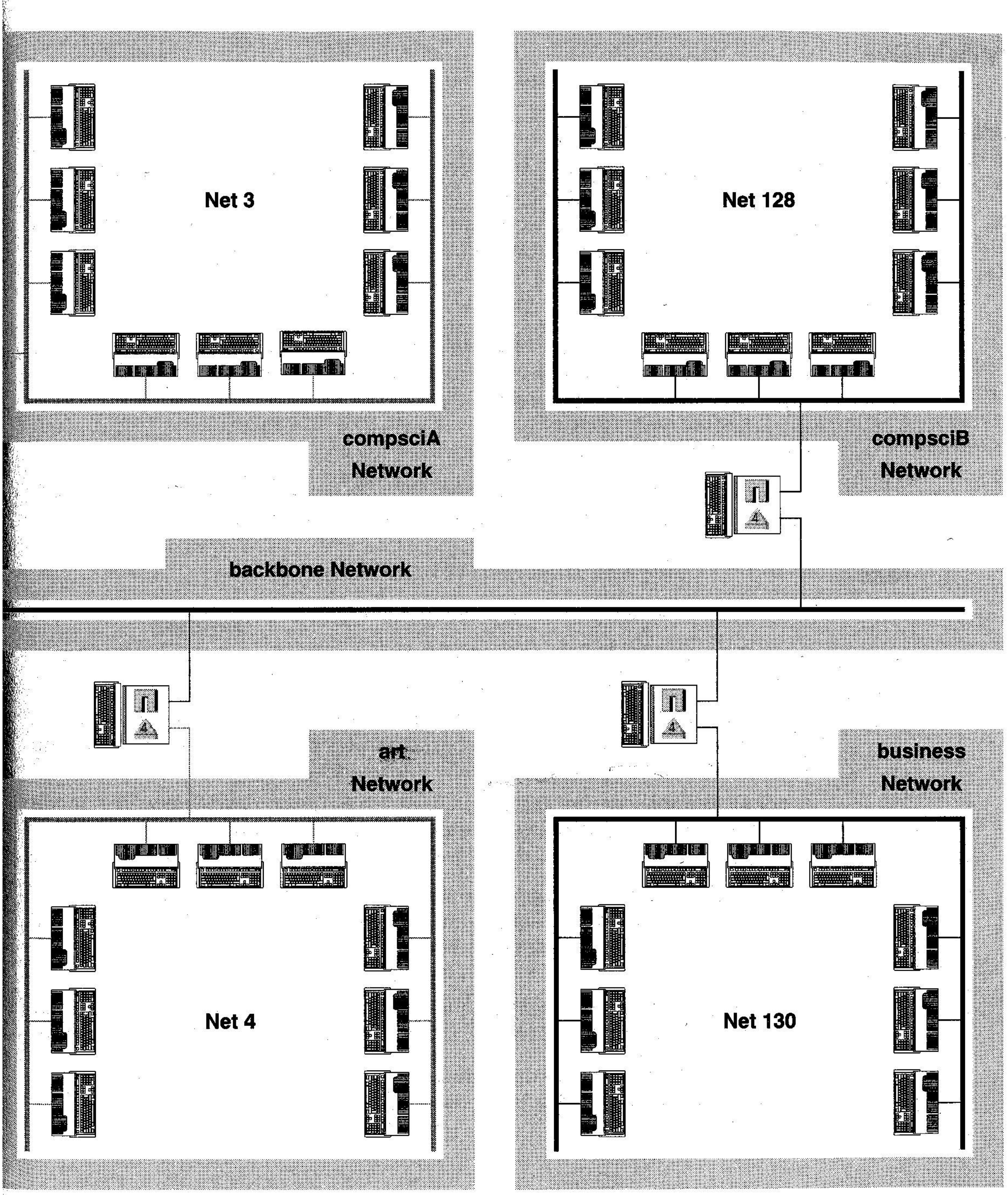
AUN has been designed with an eye to the future, to preserve users' investment as long as possible. In particular, it has been designed so that as new and faster networking technologies become available, developers can easily add support for them by replacing a single hardware-specific module.

For details on using existing Econet networks and AUN networks, refer to the guides supplied with your computer, such as the RISC OS *User Guide*. For details on installing and managing an AUN network, see the AUN *Manager's Guide*.

You should note that networking modules are only loaded if the computer has a network interface fitted.

The rest of this chapter will refer to an example network; this is shown overleaf.


Net 1
Net 2
Net 131
science
Network
Net 129


Net 3
Net 128
compsciA Network
compsciB Network
backbone Network
art Network
business Network
Net 4
Net 130

Overview

AUN concepts

The basic structure of an AUN site network is one of physically distinct networks, typically associated by location and function with a particular room, department or curriculum area. Adjacent networks are interlinked via gateway stations (described below), which pass messages between the two networks.

Networks

A *network* is a physical network of a single type (e.g. Ethernet, Econet). A network is delimited by any *gateway stations* used to connect it to other networks. For more information on gateway stations, see the section below entitled *Stations*.

Network names

Each network must have a unique name. Network names are not seen or handled by users; they are only used to configure the AUN software for a site.

Nets

A *net* is a part of a network that appears to the user as a single entity.

In both Econet and Ethernet, individual segments of a physical network can be linked together by a *bridge*. However, there is a difference between the two:

- Two bridged Econets remain distinct from each other, and so constitute two distinct nets. Hence in an Econet based network there may be several nets: the initial net, and an extra net for every bridge added.

  For an example see the diagram on page 5-168. The compsciA network is made up from nets 1, 2 and 3, which are three Econet segments connected by a bridge.
- Two bridged Ethernets appears to users to be a single Ethernet, and so constitute a single net. Hence in an Ethernet based network there will always be one net; in other words, the net and the network are one and the same thing.

  For an example see the same diagram on page 5-168. The science network and net 129 are identical, and consist of the same two bridged Ethernet segments.

It is important that you grasp the distinction between a net and a network; this chapter will rigorously distinguish between the two.

Net numbers

Each net must have a unique number.

For an Econet the net number must be between 1 and 127.

- If the net is a part of a larger Econet network linked together by bridges, its net number will already be set in the bridge, and the network manager should use the same net number for AUN.
- If the net is not connected to any other Econets (i.e. there aren't any bridges on the net) it will not have a net number assigned to it; under native Econet it will just use the default net number of 0. However, for AUN the network manager must assign it an otherwise unused AUN net number in the permitted range 1 - 127.

For types of net other than Econet (e.g. Ethernet) the net number must be in the range 128 - 252. If such a net is the **only** net on the site (i.e. the whole AUN network consists of a single non-Econet net, such as Ethernet), the network administrator need not set up a net number. It will use net number 128 by default, but – since it is the local net for all stations – users can also refer to it as net 0, in line with Econet convention.

Net numbers 0, 253, 254 and 255 are reserved.

Stations

A station is a computer connected to a net. There are two types of AUN stations.

Client stations

A *client station* has a **single** AUN-configured network interface with which it is connected to a net.

Client stations will form the vast majority of stations in each net, and are typically used as personal workstations.

Gateway stations

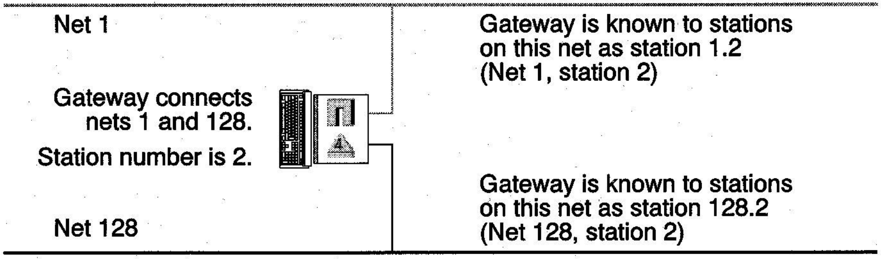
A *gateway station* has **two** AUN-configured network interfaces with which it is connected to a net in each adjacent AUN network. It relays messages between these two networks via the interfaces. The networks may be of different physical types (e.g. Ethernet and Econet). There may only be a single gateway between any two networks.

Station numbers

Each station must have a number, which must be between 2 and 254. Station numbers 0, 1 and 255 are reserved.

A station number must be unique on the net(s) to which the station is connected.

A gateway will have the same station number on both connected nets:


Net 1
Gateway is known to stations on this net as station 1.2 (Net 1, station 2)
Gateway connects nets 1 and 128.
Station number is 2.
Gateway is known to stations on this net as station 128.2 (Net 128, station 2)
Net 128

A gateway station's number must therefore be unused by any other station on either net.

echnical details

rotocols

AUN uses the UDP, IP, ARP, RevARP and RIP protocols from the TCP/IP family:

- The transport protocol is User Datagram Protocol (UDP), enhanced by a proprietary handshake mechanism designed to support the semantics of Econet SWI calls. This is not a straightforward port of the four-way handshake mechanism used by native Econet, but is rather a two-way handshake protocol overlaid with a timeout and retransmission mechanism better suited to the characteristics of IP traffic.

  TCP itself is not used, as it is a stream oriented protocol unsuited to supporting an Econet-like data delivery service.
- The network protocol is Internet Protocol (IP).
- Address Resolution Protocol (ARP) is used to map IP addresses into physical network addresses.
- Reverse Address Resolution Protocol (RevARP) is used by client stations to request their own IP addresses from gateway stations.
- Routing Information Protocol (RIP) is used to pass routing table information between stations.

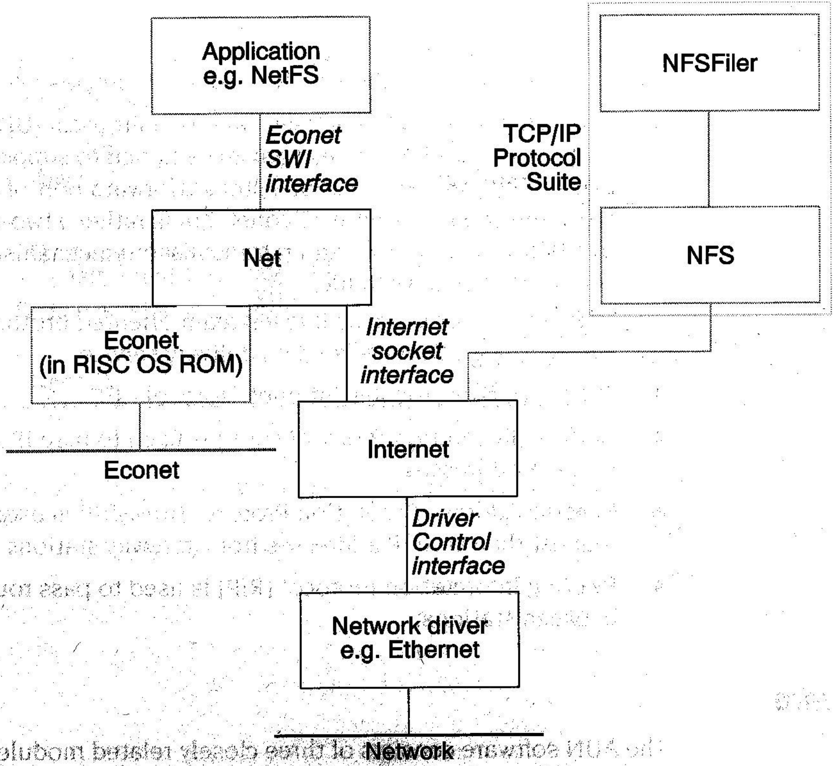
oftware

The AUN software consists of three closely related modules:

- The **Net** module implements the two-way acknowledgement handshake, and presents an Econet-like service to applications via Econet SWI calls. It also implements the RIP function.
- The **Internet** module implements UDP, IP, ARP and RevARP protocols, and exports an industry standard (Berkeley socket) interface to other RISC OS software such as the TCP/IP Protocol Suite.
- The **device driver** module enables the AUN software to communicate with a particular network interface. Each type of network interface needs its own device driver. There are no device drivers supplied in RISC OS 3.5; they are instead normally supplied with network interfaces, either in ROM or on disc.

The software in detail

The following diagram illustrates the relationship between the modules in AUN:



There is a particularly close connection between the Net module and the Econet module. The Net module learns which nets may be accessed via a directly connected Econet, and which nets need to be accessed via IP (ie nets that do not use Econet, or nets using Econet that can only be reached via a gateway). The Net module intercepts SWI calls to Econet from higher-level applications such as NetFS, NetPrint and Broadcast Loader, and – by examining the destination net number – determines whether to route the calls to the Econet module for traffic over native Econet, or to the Internet module for traffic over IP.

If the AUN station does not have an Econet interface fitted then the Econet software module will not be present, and so all traffic will be via the Internet module and IP protocol.

The Internet socket interface – used by the Net module in AUN – remains exposed for parallel use by other applications. Hence other protocols running over IP, such as NFS, can run at the same time as AUN. However, in RISC OS 3.5 the interface is only a partial implementation, supporting only those protocols needed by current Acorn products, and may change in extent or be removed in future versions of RISC OS. If you wish to program using the Internet socket interface, you should use the Internet module from the latest release of the TCP/IP Protocol Suite, which provides a full implementation. For more information, please contact Acorn Customer Services.

Since device drivers are not a part of RISC OS itself, we don't document the Driver Control Interface here. It is subject to change as the range of Acorn networking products is expanded and updated. Should you wish to program using it (say to implement a new network interface), again you should contact Acorn Customer Services.

Addresses in Econet and AUN

Under native Econet, users and programs uniquely identify each station with two one-byte numbers, thus:

net.station

Under AUN, users and programs use exactly the same scheme, to preserve compatibility with native Econet. However, the underlying Internet protocols used by AUN use four-byte numbers to identify stations. The AUN software therefore needs to translate each two-byte address passed by a user or program into a four-byte IP address. The AUN interpretation of each of the four bytes is:

site.network.net.station

The bottom two bytes (*net.station*) are the same two bytes as are seen by users and programs. The *network* byte is used to provide additional routing information to the underlying IP software only, so that it can route data to the correct destination network. The *site* byte is currently unused and always has a value of one.

Technically speaking, an AUN IP address is a Class A IP address, with a netmask of &FFFF0000.

For example, the AUN interpretation of a command – in the normal IP emphasis – to:

'send data to host 1.3.129.16'

is actually:

'send data to station 129.16... (which is located in network number 3)'

or, more meaningfully:

'send data to station 129.16... (which is located in the science network)'.

The difference between the addressing used by native Econet and the IP address used by AUN is summarised by the table below:

Network	Bytes	Form	Examples
Native Econet address	2	*net.station*	3.2 8.103 129.12
AUN IP address	4	1.*network.net.station*	1.1.3.2 1.4.8.103 1.3.129.12

AUN IP address configuration

How a gateway station finds its full IP address

When a gateway station starts up, it reads its station number from CMOS RAM. (This number is set by the SetStation command supplied with the AUN software.)

To find the site, network and net numbers of both its interfaces, the gateway station looks at its AUN Map file and Configure file.

The Map file

The Map file tells the gateway station the IP address of each net on the site. As an example, let's look at the Map file for the site illustrated on page 5-168:

```
| Example: Large site network containing 5 dept networks linked via backbone

compsciA        1 2 3                   | old compblock econet
compsciB        128                     | compblock Ethernet
science         129                     | science Ethernet
art             4                       | art room econet
business        130                     | business studies ethernet
backbone        131                     | backbone ethernet
```

The gateway station converts each network name to a network number in the order they're read; the first network has the number 1, the second is number 2, and so on. Adding in the net numbers to the example above, the following full IP addresses apply to the example network. (The site number defaults to 1, and the *station* field is read by each individual station from its configured value in CMOS RAM):

Network name	Network number	Net number	Returned IP address
compsciA	1	1 2 3	1.1.1.*station* 1.1.2.*station* 1.1.3.*station*
compsciB	2	128	1.2.128.*station*
science	3	129	1.3.129.*station*
art	4	4	1.4.4.*station*
business	5	130	1.5.130.*station*
backbone	6	131	1.6.131.*station*

The Configure file

The Configure file tells the gateway station its own position in the site: specifically, which network is connected to which interface. For example:

```
| Example1:
|       network compsciA is Econet;
|       network backbone is Ethernet.

Econet          is compsciA
Slot 0          is backbone
```

This tells the gateway that its Econet interface is connected to the compsciA network, and its Ethernet interface (in slot 0) is connected to the backbone network. What it does not tell the gateway is whether the Econet interface is connected to net 1, 2 or 3. The gateway station resolves this by reading the correct net number (in this case 2) from an Econet bridge on its own net. Thus, if the station number were 7, the two interfaces' IP addresses would be:

1.1.2.7	for the Econet interface
1.6.131.7	for the Ethernet interface

Note that an Ethernet network must always consist of a single net, and so the gateway does not have to resolve the same ambiguities as for Econet.

How a client station finds its full IP address

Like a gateway station, an AUN client station reads its station number from CMOS RAM at start-up time.

However, at this stage it does not know its site, network and net numbers; instead, it finds these out from a gateway station connected to its local network.

To do so the client station broadcasts a RevARP message requesting its IP address The gateway receives this broadcast on the interface that is connected to the client's network, and returns that interface's IP address, first setting the station number to zero:

site.network.net.0

Because the gateway station's interface and the client station are on the same network, the returned site and network numbers are therefore the same as those of the client station. The net numbers will also be the same, unless the client station and the gateway station are on different nets within the same network (which can only be the case if they are separated by Econet bridges).

The client station takes the returned address and substitutes its own station number. It also determines if it is connected to a bridged Econet; if so, it replaces the returned net number – which may be incorrect – with the correct net number, read from an Econet bridge on its own net.

Default addresses

If a client station does not get a response to its request for its full IP address, this means that no gateway computer is present and so the local network is isolated. This being the case, then:

- If the station is connected to an Econet it will use native Econet rather than the Internet protocols used by AUN.
- If the station is connected to any other network it adopts a default IP address of 1.0.128.*station*, giving a user address of 128.*station*.

When/if a gateway computer subsequently comes 'on-line' it will immediately send a message to the other stations on the previously isolated network, so they may then complete their address and routing configuration, and get access to all other networks in the AUN system.

Consequently while a network is isolated all its stations may communicate between themselves; stations don't 'hang' awaiting a response from a gateway. You may later start up a gateway station to bring the isolated network into your site's AUN network. However, since this is likely to change 'on the fly' all the addresses of that network's stations, you must take care only to do this when there are no users active on the network.

Application program interface

The application program interface, or API, is the same as the RISC OS 3 (version 3.10) Econet SWI interface, with certain usage qualifications described below. For full details, refer to the RISC OS 3 *Programmer's Reference Manual*.

Existing user applications which access Econet do not require functional modification at the network interface in order to run over an AUN network.

The AUN module intercepts SWI calls to Econet from user software. It treats the calls differently according to how it can access the destination station:

- If the destination station can be accessed directly via Econet, AUN passes the SWI calls to the resident Econet handler. This avoids unnecessary IP protocol overheads for a localised Econet-only transaction.
- Otherwise the destination station must be accessed via IP. AUN maps the SWI calls into calls to the Internet module, having first expanded the two-byte *net.station* destination address into a four-byte *site.network.net.station* IP address.

The maximum amount of data which can be passed in a single transmission SWI via IP is 8192 bytes.

When transmitting to a station via IP, transmission SWI calls will return only the error values Status_NetError and Status_NotListening in the event of failure. Over raw Econet other Econet-specific error values may be returned.

Constraints on the use of Econet SWI calls over AUN

Immediate operations

In general the Immediate mechanism is considered to be Econet specific. The only Immediate operation supported by AUN over IP is Econet_MachinePeek. All other Immediate SWI calls return Status_NotListening, unless the destination station is accessible via a directly connected Econet.

Transmission strategy

An application's choice of values for the Count and Delay parameters it passes to transmission SWIs may make assumptions about the actual physical characteristics of Econet. For example some Econet utility programs set the Count to 0 in Immediate operations, relying on the fact that the return of a scout acknowledge frame in response to a valid scout frame will always be effectively instantaneous. However, over an AUN IP network this assumption is invalid; the functional equivalent of the scout acknowledge may arrive 'sometime', or even 'never'.

Consequently AUN uses a retransmission strategy more suitable to the nature of IP traffic, whilst retaining the existing retransmission strategy for transmissions to a directly connected Econet. The retransmission strategy for AUN over IP is as follows:

For ordinary data, AUN employs a two-way handshake. A receiving station will return a positive acknowledgement if it has successfully received a data frame into an open receive block, or else a reject message if there is currently no open receive block, or some other detectable reception error has occurred.

```
If Count > 1
    The maximum elapsed timeout period in seconds (T) requested by the
    application is computed as:
    $T = (\text{Count} \times \text{Delay}) / 100$.

    On receipt of reject messages, the sender will retransmit the data frame 10
    times after 1 centisecond timeouts, then:
    If $T < 5$
        $T \times 10$ retransmissions will occur, each after 10 centisecond timeouts;
    Else
        If the destination station is not on the same network as the sender
            exactly 50 retransmissions will occur, each after $(T \times 100) / 50$
            centisecond timeouts;
        Else
            If the retry delay < 25 centiseconds
                exactly 50 retransmissions will occur;
            Else
                $(T \times 4)$ retransmissions will occur, each after a 25 centisecond
                timeout.
        (This provides some optimisation for simultaneous loading of software
        from a local file server, whilst protecting against excessive overload at
        gateway stations caused by rapid retransmission.)

    If no response is received at all then:
        If $T < 5$
            1 retransmission will occur, after a 5 second timeout;
        Else
            $T / 5$ retransmissions will occur, each after 5 second timeouts.
Else
    The sender will transmit exactly once. The transmission status will not change
    until a positive acknowledgement or a reject message has been received, or a 5
    second timeout has elapsed.
```

For an Immediate operation (i.e. Econet_MachinePeek), a SWI call with Count = 0 or Count = 1 always results in a Status_NotListening return; no actual network transmission is made. In other cases the sender transmits an Immediate message exactly once, changing transmission status only when a response has been received or a 5 second timeout has elapsed.

Bridge protocol

Use of the Econet Bridge protocol by a RISC OS net utility program to identify valid net numbers does not work over non-Econet networks within an AUN system, as no actual Econet bridges are present to respond. However, cycling through the range of net numbers in a sequence of calls to Econet_ReadTransportType can provide this information without involving any network transactions; the call returns R2 = 0 if the given net number is not currently accessible from the local station.

Note that this constraint does not affect use of the Bridge protocol onto a directly connected Econet system.

Meaning of net 0

In AUN, a station may be connected to both an Econet and an Ethernet at the same time. This means that the assumption that Net 0 means the local network is no longer safe, as the AUN software could not, in this case, distinguish the two connected networks with certainty. Hence applications running over AUN should strive to supply an actual net number with every transmission SWI call.

You should note that the actual net number of a connected Econet may in fact be 0, if there are no bridges present; however the net number of an Ethernet in a correctly configured AUN network can never be 0, so no clash will occur. If a net number of 0 is supplied to a transmission SWI, AUN maps it to the net number of a directly connected net, with Econet taking priority over Ethernet if both are connected.

Local broadcasts

If a station is connected to both Econet and Ethernet, transmit SWI requests for a local broadcast – as issued by Broadcast Loader– are directed to the Econet only.

Data delivery

As with Econet, AUN over IP cannot guarantee that a message apparently correctly received and acknowledged by a receiving station will not be retransmitted if the acknowledgement is lost in transit. Applications using AUN should therefore ensure that they can detect whether a transmission has been repeated. This is usually done by adding a sequence number or bit to transmissions.

Printer server protocol interface

The printer server protocol interface was omitted from the RISC OS 3 *Programmer's Reference Manual*. It is as follows.

NetPrint status protocol

Status enquiry packet

To request the current state of a printer server the client sends an 8 byte status enquiry packet to port &9F:

Byte	**Meaning**
1 - 6	printer name, padded with spaces
7	reason code (1 ⇒ status request, 6 ⇒ name request)
8	reserved (must be zero)

Status request

If the reason code is 1 (status request) the printer server should check the printer name. The check should be case insensitive, but with accents significant, preferably using Territory_Collate (see page 3-834):

- If the name matches the name of a printer connected to the server (eg 'PScrpt'), the server should send its status.
- If the name matches the string 'PRINT' or 'SPOOL', the server should send the status of the user's default printer. (With Acorn's !Spooler software, this is the most recently used printer, or the first listed printer if none has yet been used).
- If the name matches neither of the above cases, the server should not reply.

The status reply, if any, must be sent to port &9E:

Byte	**Meaning**
1	status: 0 ⇒ Ready, 1 ⇒ Busy, 2 ⇒ Jammed, 6 ⇒ Offline; all other values reserved
2	station number for Busy status, or 0
3	net number for Busy status, or 0

If the server is Busy, the second and third byte of the status packet are the st and net number with which it is busy. If the server is Busy with no particu station, or if the status is not Busy, these bytes should both be set to ze

Using the name 'PRINT' is deprecated because it makes it difficult fo server that supports multiple logical printers. Wherever possible yo the printer's name.

Name request

If the status enquiry reason code is 6 (name request) then the client is asking the printer server for its name. The name sent by the client is 'PRINT' or 'SPOOL', but it is not necessary to check this. The server must reply to port &9E:

Byte	Meaning
1 - 6	printer name, padded with spaces

If the printer server supports multiple logical printers it may send multiple replies with different names. If the client discards duplicate replies then it should take account of the name in the packet as well as the station and net numbers.

Flag bytes

For all status packets the flag byte currently has no meaning. Clients should send a flag byte of zero, and servers should send back the flag byte that they received from the client.

NetPrint printing protocol

Finding the status before printing

Before starting to print, the client should ideally send a status enquiry to the server to ensure it is ready (see above).

Establishing the connection

The connection is then established using packets where only the flag byte is relevant. It has this meaning:

Bits	Meaning
0	sequence bit
1, 2	modes
3 - 6	task id
7	reserved (must be zero)

The client first sends a packet to port &D1 on the server, with either zero or one byte in it (the contents of which don't matter), and with the flag byte's sequence bit clear, its mode bits set to 2_01, and its task id bits set to either 2_0000 or 2_1000 depending upon the version of NetPrint.

- If the task id is 2_0000, then the client will only send data in &50 byte blocks.
- If the task id is 2_1000, then the client code is both asking for the allocation of a task id by the server, and trying to establish if the server can accept large blocks of data (up to the size returned by SWI Econet_PacketSize) or only small ones (up to &50 bytes).

If the server is unwilling to accept the print it doesn't send a reply. If it is willing then it sends back a single byte of any value to port &D1:

- If the client's task id was 2_0000 (ie small packets only), the server's flag byte is the same as that it received from the client.
- If the client's task id was 2_1000 (ie request for large packets and task id), the server uses the flag byte to respond to the request.

  If the server isn't willing to assign task ids – and hence accept more than one connection from a single client – it sends back the client's (illegal) task id of 2_1000; otherwise it sends back a task id chosen from the ranges 2_0001 to 2_0111, or 2_1001 to 2_1111.

  If the server can accept large blocks of data it sets the mode bits to 2_10, else it sets them to 2_01.

The connection is now established. The final flag byte sent by the server is the one that will be used when sending the data.

Sending the data

The client then sends the data in blocks, the size of which can vary from zero bytes up to the maximum established by the connect protocol. The flag byte for each block is the same as that negotiated when connecting (see above), save that the sequence bit is toggled for each block.

Each time the server receives a block and is ready to accept another, it must acknowledge the received block with a one byte packet. The packet's flag byte must match that received from the client; its byte of data must be zero.

Closing the connection

When the client wants to close the connection, it sends a data packet with the mode bits set to 2_11. The data for this last packet must be terminated by an &03.

Port claiming

NetPrint claims ports &D1 and &9E with Econet_ClaimPort. A printer server should claim port &9F.