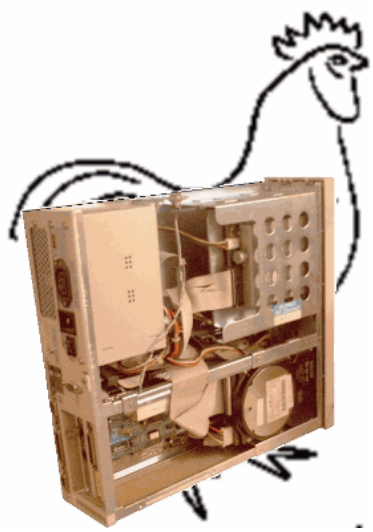
Salvage Server Project Report 2:

Networking Computer Systems Using Ethernet



Produced by the Free Range Salvage Server Project, September 2003
email: fraw@gn.apc.org web: http://www.fraw.org.uk/ssp/

A computer on its own is a useful typewriter, or a games console. But the true value of a computer system doesn't present itself until you connect it to other computers to enable communication and resource sharing. This report looks at the issue of connecting systems together using a 'local area network', or 'LAN'. It explains the concepts behind a LAN, and outlines how to set one up using trash. tech and the Gnu/Linux operating system.

Trash networking means 'Ethernet'

Networking is a complex subject – although it needn't be. For example, a large corporate network might link hundreds or thousands of networked workstations, all communicating and sharing data through a series of large buildings. But if you only want to link two to twenty systems you needn't get bogged down in the minutiae of how networks operate.

There are many different networking standards in use – each internationally agreed by the Institute of Electrical and Electronic Engineers (IEEE). For linking PCs, the easiest option is to use 'Ethernet' (see box, right). There's plenty of old equipment in circulation, and even new equipment isn't prohibitively expensive to obtain.

Using Ethernet requires that each system be fitted with a 'network interface card', or NIC. The NIC allows the PC to communicate with the network. If the PC doesn't have this already, you'll have to fit one. The thing to beware of here is the type of card you can get hold of (whether it's an ISA slot or PCI slot card). Also, the speed the card operates at (see discussion on speed in relation to hubs below). Check the the hardware compatibility data for your Linux distribution to see which NICs are supported.

With the older 'thinnet' system, PCs were connected together in a long cable or 'bus', one to the next. Simple, but the RG-58 cables are expensive to buy (because they're not so much these days), and the BNC connectors are a pain to fit. Thinnet also has problems when there are a lot of machines using the network at once. For this reason we prefer to use 'twisted pair'. Twisted pair cable is easier to get hold of, the connectors are a lot easier to fit. On UTP networks, cabling faults in the system are less likely to bring the whole network down – only one machine

Ethernet Network Types

Ethernet, or the IEEE's '802.3 network standard', has been around for a while. The key factor in network design is the speed of communication, rated in 'mega-bits-per-second' (Mbps). The main Ethernet types you will come across are:

10Base-2 or 'thinnet' – widely used to create the first big network of PCs in commerce, 'thinnet' is a lightweight coax cable (like TV aerial cable) designed to link PCs at 10Mbps (10 'meg'). Unlike modern Ethernet, 10Base-2 connects PCs one to another as a 'bus', with no central hub or control equipment. There's quite a bit of 10Base-2 equipment out there, but whilst cheap to connect two or three computers, you can run into problems as you cable more systems together. For this reason we don't cover 10Base-2 in this report.

10Base-T or 'unshielded twisted pair' (UTP) – the successor to 10Base-2, it also runs at 10Mbps (10 'meg'). It uses cable similar to that used for internal telephones containing four 'pairs' of wires twisted together. 10Base-T also requires a hub, to which each system is connected, to manage traffic on the network. This makes it a little more reliable, and easier to mess around with whilst online, than 10Base-2.

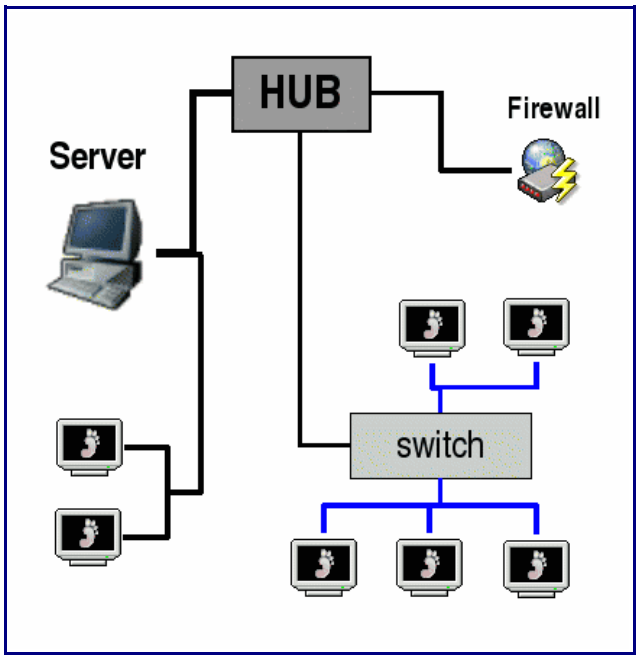
100Base-T or 'fast Ethernet' – the successor to 10Base-T, it runs at 100Mbps (100 'meg'). Whilst identical in general design, it uses higher quality 'Category-5' or 'UTP Cat-5' cables and connectors – old 'Cat-3 cable' can't support the bandwidth demands.

'Gigabit Ethernet' – recently 100Mbps was superseded by 'gigabit' Ethernet running at 1000Mbps. It's still rather expensive, and so is not the sort of thing you're likely to be using in the context of trash tech. systems.

is affected rather than isolating a whole block of clients.

The other component you need is a hub. The hub connects the systems together, one hub port/UTP cable per client. The role of the hub is to manage the movement of data over the network. Hubs are rated according to speed. There are a lot of old 10Base-T '10-meg' hubs around. Most new 100Base-T '100-meg' hubs are dual speed, and will work at the speed of the NIC connected to the hub. If you only have old 10-meg NICs, then a 10-meg hub will be OK. But it's a waste of resources to use a lot of 100-meg NICs (most 100-meg NICs are dual speed too) on a network with a 10-meg hub – upgrade to a 100-meg dual speed hub instead.

A simple Ethernet network



Networks and operating systems

To run a network you need a server. This controls the network, and organises the network services that the clients on the network use. You have two options: use a proprietary system, or use Gnu/Linux. Proprietary systems are usually additional to the operating systems, whilst with Linux all the required software comes as standard with the Linux distribution.

Proprietary network systems, like Microsoft's *Windows NT*/*Windows Server*, or Novell's *Netware*, can be obtained from computer fairs. But to remain legal you not only have to have a valid license for the program – you also have sufficient client licenses to cover the maximum number of systems that you intend to connect to your network. So, as well as hitting you for the software license, they also tax the connection of machines to the network.

Gnu/Linux doesn't have these problems. As a Unix clone, it's designed for networking systems together – in fact, it so desperately wants to network that in the absence of a network it will even network with itself using a 'loopback interface'.

Most of the major Linux distributions (see *Salvage Server Project Report 1*) have built in networking capabilities, but some are better than others. Red Hat is a really good server system, but it's better suited to text-based/command line configuration. SuSE Linux, with it's *YaST* configuration system, is excellent for less experienced Linux users because of its graphical network configuration tools.

In this report we're assuming that you've got a system installed with Gnu/Linux. Rather than discuss the precise details of configuration (as it's slightly different with different distributions) we'll outline the main principles of network configuration. However, specific parts of this process, like configuring network services, will be tackled as separate Salvage Server Project Reports.

More ambitious networks

Networking a few computers together is relatively easy. But problems will arise when you want to get more ambitious, such as when you want to add more servers for specific tasks, or you want all the systems on the network to share an Internet connection (see diagram left). Simple networks consist of just a hub and PCs. But as the network grows you will need to modify the way the network hardware is organised.

Most likely, you'll want to connect to the Internet. This means that you will have to configure some sort of router/firewall system to manage the flow of data from the network into the phone line/broadband connection. This can be done from the server, but it is very complex. For this reason we'd advise people to use a dedicated machine as a firewall. This can be installed with Linux and configured manually. But what's far easier is to use a very old machine (e.g., a '486 DX4 or an early Pentium-1 P60-P100) installed with *Smoothwall*. This has a simple installation and maintenance interface that's easy to use for the less geeky Linux user. Installing/using *Smoothwall* will be the subject of a future Salvage Server Project Report.

Most hubs come as 4-, 5-, 8-, 16- or 24-port units. When you fill all the ports available on the hub you have to get a larger hub, or get a second hub and 'daisy-chain' it. Most hubs have a special port, or a port with a switch beside it, that allow you to plug that port into a port on your main hub. The port then acts as a sort of extension socket for

the main hub.

When hubs send data over the network, they send it across all the cables connected to the hub. On networks with lots of machines, or where you have a special server that works with only a few other machines on the network, this can cause a lot of congestion. In these situations you can create small sub-sections within the network using a 'switch' to manage network congestion.

The switch looks and works like a hub. But it monitors network addresses and will route data for the machines connected to it through its own ports. This allows you to manage traffic between different areas of the network without the need to physically split the network using a router. Only data meant for other machines on the network gets routed back to the hub. This helps control congestion on the network. It also means you can set up small areas on the network where people can work at high capacity with servers, or network resources like printers, without drawing down the capacity of the whole network.

Cabling the network

Cabling the network requires a little thought. Firstly, you're going to need one cable for each machine – with sufficient length to snake around from its location to the hub. You can buy network cables of varying lengths from 1 metre up to 100 metres. But the cheapest option if you need quite a

Making-up an Ethernet cable

Ethernet cables are expensive to buy. If you intend making up a number of them you can save money by buying Ethernet cable on a roll (£25/100m), some RJ45 cable ends (10p each), and some RJ45 crimping pliers (£30) [all prices from Maplin Electronics – http://www.maplin.co.uk].

You could use new cable cut to length off the roll. You can also get very long cables from clearance sales and cut them down to save money on cable. Either way, you'll have to crimp the RJ45 connectors onto one or both ends of the cable. Also, if you regularly plug- and un-plug RJ45 connectors the plastic catch becomes very brittle and breaks off. In these cases you can repair the cable by cutting off the connector and crimping a new one on.

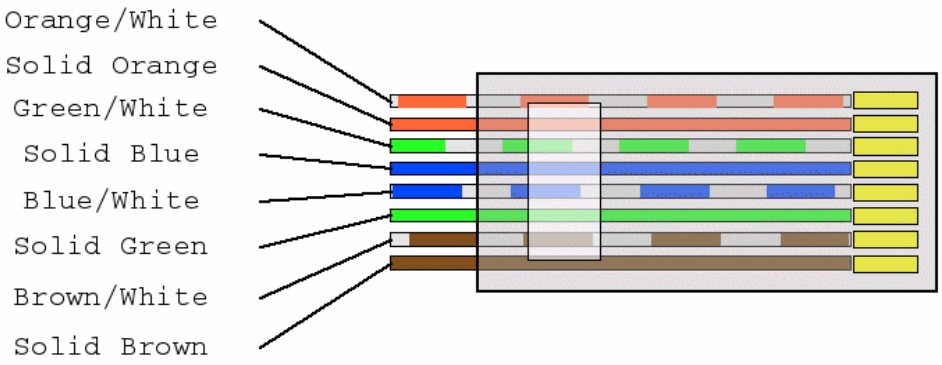
What's really important is that you get the order of the wires right. The network connection uses four of the wires (the other four are intended for a telephone). The diagram to the right shows the order of the wires with the back of the connector facing you, and the metal contacts facing to the right. This assumes that you hold the connector in your right hand and push the wires in with your left – if you hold it the other way around turn the diagram upside down.

The RJ45 connector has eight small channels inside it. Strip about 1cm/½" of the plastic sleeve off the cable using a sharp knife or wire strippers (don't knick the insulation on the wires inside). Each of the wires then feeds down the channel to the end. It helps if all the wires have been cut to the same length.

There's no need to strip the insulation off the wires because the crimping process pierces the insulation to make the connection. When all the wires are in the right order, fit the crimping pliers around the plug and squeeze hard. The pliers are adjusted to stop in the right place, so you shouldn't crush the connector body – unless you've put it in the pliers wrongly.

When the connectors have been crimped on you should check the cable works properly. You can do this by putting it into a network and seeing if you can get data down it. But it's far better if you can get hold of a Ethernet cable tester. You plug both ends of the cable into the tester box, and if all's well, you'll get some lights telling you that the wires have been properly connected. If you don't, it's because you've connected the wires wrongly, or the crimp connector hasn't made a good electrical connection.

Finally, take note – if you are making cables for a 100-meg network, make sure that all the cables and connectors are rated 'Category 5' or 'UTP CAT 5'. This cable is of higher quality than the older 10Mbps 'Cat-3' cable.



few cables is to buy some crimps and RJ-45 connectors and make your own (see box below).

There are restrictions on cabling-up Ethernet networks:

- Cables should be a minimum of 1 metre long, and a maximum of 100 metres long – in general, the longer the cable runs the more power your hub will have to pump into the network to keep things running.
- For 100-meg networks, the cable and the connectors must be rated as 'Category-5' or 'UTP Cat-5' – the old 'Cat-3' cable/connectors are not of sufficient quality to reliably transfer data, especially when the network is busy.
- Avoid a large number of tight coils if gathering up cable as this can impede the flow of data (the cable forms a coil which dampens the flow of current).
- Avoid running the cables along side mains wiring or electrical ring mains because the Ethernet cables may induce electrical noise that can interfere with radios, stereos and other sensitive equipment connected to that circuit.
- If daisy chaining hubs and switches, don't string out more than three hubs/switches in a row – use another port on the main hub to increase capacity instead.

The other problem you may have is with electrical noise. Electric motors, large transformers, power supplies and switchgear/power relays create magnetic fields that can induce or dampen the current flow in the cable. With excessive electrical noise you may find that data transfer is slowed, or is impossible. In these situations you should use the more expensive 'shielded twisted pair' (STP) cable to cut-down the level of interference.

If the cables are installed, and left in-situ, then they should pose no problem. Problems will occur where cables are regularly plugged, unplugged, and moved around. The small clip on the RJ-45 connector is very brittle and can easily be broken off after a long period of repeated unplugging. In these situations just cut off the connector and crimp another on. The metal in the wires within the UTP cable is also very soft – in order to make them flexible. However, repeated bending into tight curves, or stretching the length of the cable, can lead to deformation and eventual breaking of the wire. In these situations, after checking to be sure that the cable is faulty, you'll have to junk the cable and get a new one.

The important thing when cabling any network is to leave slack in all the cables, including the power cable for the hubs/switches. This will ensure that if caught, the cables are not stretched and/or pulled out of their sockets. It's also important to plug the hub power supply into a power socket that's not likely to be unplugged or turned off. If necessary, label the power supply and any switches to ensure that they are not turned-off/removed by accident.

'Thin client' networks

As servers and networks become more powerful, rather than having a very powerful workstations, system engineers run software on the server and have a less powerful terminal to allow the user to access their programs. These systems are called 'thin clients'.

In the trash tech. world, thin clients are also used. Old, less powerful computers can be configured as thin clients, accessing programs on the server at higher speed than if they were run locally. Whilst there's nothing wrong in theory with this model, from an engineering point of view such a model has a 'single point of failure'. If your server goes down, then nothing happens anywhere on the network.

For this reason, especially where there may not be immediate support to repair/reconfigure the server, thin clients should be used with caution. In our view, it's better to have some old machines doing basic functions, like word processing or accessing the Internet, rather than trading off the additional speed in return for a less secure system.

IP numbering and dynamic numbering (DHCP)

Networks use the same type of numbering as the Internet – the 'Transfer Control Protocol/Internet Protocol' (TCP/IP) system. IP numbers are made of 4 bytes – 32 binary digits. To make this more human-friendly these are presented as four decimal numbers – e.g. 192.168.67.1.

To ensure that the numbering of the LAN doesn't conflict with the Internet, there are 'reserved numbers' that should be used to number your network. The number depends upon the type network you are creating:

- 'Class A' networks – numbers 10.0.0.0 to 10.255.255.255 – 16.7 million possible numbers, for use on large networks.
- 'Class B' network – numbers 172.16.0.0 to 17.31.255.255 – 1 million possible numbers, for use on medium sized networks.
- 'Class C' networks – numbers 192.168.0.0 to 192.168.255.255 – 65,536 possible numbers, for use on smaller networks.

Small networks are 'Class C'. IP addresses are used in blocks of 256 numbers. Therefore you would select 192.168.1.X or 102.168.2.X, etc. The 'X' refers to the 'subnet' of numbers, running from 0 to 255 (256 numbers in total). Each subnet has a 'network address' (the server)

Glossary

Daemon – a daemon is a program that runs on a server and provides a network service.

DHCP – *Dynamic Host Configuration Protocol* – this daemon automatically allocates IP numbers to clients that logon to the network using DHCP. It's provided by the dhcpd daemon.

DNS – *Domain Name Server* – this is a daemon that turns IP numbers into domain names and vice-versa. It's provided by the Bind daemon.

Domain name – the name given the the network, implemented by DNS.

Firewall – firewalls monitor the flow of traffic between networks in order to restrict undesirable connections. Usually used as a security measure between LANs and open Internet connections.

FTP – *File Transfer Protocol* – a daemon that moves file over a network from one machine to another. It's provided by the ftpd daemon.

Gateway – a machine that translates network communications between different networks, for example between a LAN and the Internet.

Host name – the name given to a particular machine on a network. This may be set on the machine, or allocated via DHCP.

Hub – a hardware device that links NICs on a UTP network.

IP number/address – a 4 byte/32-bit number that identifies a particular NIC on the network.

LAN – *Local Area Network* – a network of machines that access a common network server.

LPD – *Line Printer Daemon* – a daemon that provides printing services on a network.

Netmask – a number used in conjunction with an IP number to control the selection of the NIC on a particular machine.

NFS – *Networked File System* – a protocol that allows the files on the server to be accessible to other machines over the network.

NIC – *Network Interface Card* – a card inserted into a machine that allows communication with the LAN.

rlogin – a daemon that allows execution of instructions on one machine sent from another. It's provided by the rlogind daemon.

Router – a hardware device that translates IP numbers to route data between different subnets on a LAN.

Samba – a daemon (one of a pair) that replicates the SMB system that forms the basis of networking within Microsoft programs. It's provided by the smbd and nmbd daemons.

Subnet – a block of IP numbers.

SSH – *Secure Shell* – a daemon that allows execution of instructions on one machine from another, but the session is encrypted for greater security. It's provided by the sshd daemon.

Switch – like a hub, it organises communication between machines on a network, but it restricts the forwarding of data to only those machines that are connected to it.

Telnet – a daemon that allows execution of instructions on one machine from another. It's provided by the telnetd daemon.

UTP – *Unshielded Twisted Pair* – the cable used for Ethernet networks.

and a 'broadcast address' (used to contact all clients). Usually, you also have a 'gateway' address where the subnet access other networks or the Internet. So in any subnet there are a possible 252 IP numbers that we can use to connect client machines to the network.

Along with the IP number, there is also a 'netmask'. This is another 4 byte binary number that is used to control how the IP number is interpreted. On any network, the netmask can be used to 'mask' the selection of IP numbers by the NICs. This allows the block of 256 number to be split into smaller networks of 2, 6, 14, 30, 62 or 126 numbers. To do this you have to set up gateways or routers to manage communications between the subnets, and for this reason it's rather complex. Unless you've a good reason to split into smaller subnets, small networks nearly all use the entire block of 256 numbers. Therefore the netmask that you use will be 255.255.255.0.

When planning a network you have to decide the numbering of the server and any clients. By default (although you can change them) the main network addresses are:

- Server – 192.168.X.1.
- Gateway – 192.168.X.254.
- Broadcast – 192.168.X.255.

The client machines can have fixed IP addresses. If you use fixed addresses you have to set these individually for each client, and ensure that IP addresses do not conflict with each other. On a small network of a few machines this is not a problem. But as you increase the number of machines it can increase the amount of work if you ever need to reconfigure the network numbering scheme. For this reason larger networks allocate numbers dynamically using the Dynamic Host Configuration Protocol (DHCP) service (the 'dhcpd daemon').

DHCP works as part of the server system. It listens to requests from machines as they boot up, and allocates them an IP number to use. The number allocated are fixed

as a range in the configuration file for the DHCP program. For example, if the addresses 192.168.X.100 to 192.168.X.199 are set as the range, it means that DHCP can log on up to 100 machines to the network. DHCP is also useful where people may bring their own computers to your network. Most operating systems automatically configure the use of DHCP so that the system can be logged on to any network that it is connected to.

Domain names and DNS

The Internet uses names to identify different machines. The same can be done on a LAN to allow people to access the functions of a LAN more easily. Also, on a LAN that uses DHCP, often it's easier to use names than numbers because you won't always know what the network numbering scheme is.

Each network can be given a domain name. For example, 'mynetwork.lan'. The purpose of this is to provide a general name to which host names – the names that identify specific machines or services – can be added. For example, if we ran an Intranet to provide a local web service, this could be called 'www.mynetwork.lan'. Note that the '.lan' part is not necessary – but it's very useful yo identify services that form part of the local area network from those of other connected networks or the Internet.

This is implemented via a 'domain name server' (DNS). The DNS system, called Bind, is provided by a daemon, called named, that runs on the network server, or some other server than can be contacted from the LAN. It receives DNS requests from the network for:

- *Forward resolution* – turning names into IP numbers.
- *Reverse resolution* – turning IP numbers into names.

To create a DNS server you have to create some files. First you need to edit the files that control the operation of Bind. Then you have to create 'zone files' that provide the details required to provide DNS services. The 'forward zone' files provide the information the hosts that are identified with a particular domain, and the numbers that they respond to. The 'reverse zone' files list the IP numbers, within the local subnet, that identify particular machines.

Principal Local Network Services

Name	*Function*	*Purpose*
DHCP	Dynamic Host Configuration Protocol	Allocates IP addresses dynamically to clients.
DNS	Domain Name Server/System	Provides domain names to identify local services on the server.
LPD	Line Printer Daemon	Provides a centralised printing system so that clients can access a single printer.
NFS	Networked File System	Exports the file system of the server, or another machine, so that other machines on the network can access the files as if they were located on the machines own hard disk.
Samba	*Windows* SMB/NMB networking	Allows *Windows*-based clients to access the network.
FTP	File Transfer Protocol	Allows the movement of files over the network, usually between the client and the server.
Telnet	Remote login	Allows command line login to execute instructions on one machine from another.
rlogin	Remote login	Same as Telnet, but less secure.
SSH	Remote login, with encryption	Like Telnet, but encrypted for improved security.
HTTP	Hypertext Transfer Protocol	A web server, allowing the development of Internet/Intranet systems over the LAN.
SMTP	Simple Mail Transfer Protocol	Delivers email over the local network.
POP	Post Office Protocol	Allows retrieval of email over the network.
IMAP	Internet Message Access Protocol	Allows access to email stored on a server, rather than downloading, as you do with POP.

DNS can be quite complex to set up, and the process varies from distribution to distribution. For this reason it will be covered in greater detail in another Salvage Server Project Report.

Sharing printers and files

Linux systems, even clients, operate printers via a 'queue'. Print jobs are received and queued. The printer daemon continually monitors the queue, and when it finds a job waiting, it passes it on to the printer. This means that to create a networked printer you create a print queue on the server, and then set up printer queues on the clients that point or 'forward' to the queue on the server.

The information that you have to supply to the client is the name of the print queue and the type of printer. The name is usually in the form 'printer@server.lan'. You then set the printer information locally in order to ensure that the format of the data sent to the queue is correct for that type of printer. You then give a name to the remote queue in order to differentiate it from other printers that you might configure on the client You can configure a printer locally, connected directly to the client, in addition to one or more networked printers.

The other main use of a network is to share files. For Linux machines, the simplest way of sharing files is a Networked file System (NFS). NFS reads information from a file called 'exports' (most distributions give you a graphical program to configure this file) that specifies which directories within the local file system will be made available over the network. Each line of the file also allows control over which machines have access, and whether that access is read/write or read-only.

At the client end, NFS requires that you create a directory to form the mount point of the exported directory. The file that controls the mounting of file systems, 'fstab', is then modified to mount the networked directories when the client machine boots up.

If you want to connect *Windows*-based clients to the network you have to configure the Samba service. This is similar to NFS. However, the results can vary depending upon which version of *Windows* you are using. Sometimes problems with the *Windows* registry can block full network access. Also, the Samba system requires complex configuration in order to establish *Windows* 'shares' on the server, and to connect networked printers.

The other option for file transfer is File Transfer Protocol (FTP). This is a service that allows the movement of files between one machine and another using an FTP program. There are two forms of FTP:

- User-based FTP, which allows file to be moved to a specific user account, and which is password protected for security.
- Anonymous FTP, which allows anyone to download/upload files.

How this is configured depends upon the FTP daemon you use. If you use *wu-ftpd*, just activating the service enables user-based FTP, but not anonymous FTP. If you use *pure-ftp*, that if configured to automatically provide anonymous FTP, but must be configured for user-based FTP. Configuration details are usually provided with the documentation that accompanies the daemon program.

FTP is useful because it's more efficient at moving huge files than NFS. So if users back-up their data to the server, FTP can provide a quick and easy alternative to NFS. FTP is also useful for Windows machines that don't have access to the network using Samba. Even though the Windows machine is not properly logged onto the network, it still has access to the basic TCP/IP services. This means that the machine can move data on and off its hard disk to the server using FTP.

Intranets and mail

An Intranet, or local web server, is very simple to set up. All you do is install the Apache web server, enable the service, and straight away you should be able to access the test page. All you need do then is replace the test pages with your own web site in order to run your Intranet.

Of course, operating a proper Intranet requires a lot more thought and editing of configuration files. In particular, if you want to use local search engines, or other web based tools, you will have to enable these individually and edit the configuration files as required. But for a simple network, for example where you only require very basic information hosting, just enabling the web server daemon should work OK.

Configuring email is a lot harder. By configuring email the clients on the network can mail each other – which is often the easiest way for users to share information and files. Depending on which mail transport agent (MTA) you use you will have to configure user accounts on the server as well as the MTA. *Sendmail* is the most complex to configure – mainly because it's such a complex programs. Others, like *Exim* or *Postfix*, are easier to configure, but not all Linux distributions install and configure them as standard as part of the installation process.

For uses to check their mail you have to configure a mail delivery agent (MDA). The simplest MDA is the Post Office Protocol (POP). This is provided as part of the 'IMAP' package with most distribution. All you do is install POP and activate the daemon. Then users can check their mail.

The Internet Message Access Protocol (IMAP) is also simple to configure. But unlike POP, IMAP allows access to messages stored on the server – so you only download a particular message when you need to read it. However, the trade-off with IMAP is that all the accumulating email can clog up a small hard drive if you use an older machine as a network server.

Networking benefits

This report is just a quick run through of what a network is, and the steps that you need to go through to set one up. But it's also important to understand the benefits of setting up a network.

First an foremost, networking improves the abilities of a computer to store and exchange information. For example, why back-up data to CD? It is easier and faster to back-up files to another machine over a network. If you have more than one machine in use, a network also enables those machines to share resources – like the same Internet connection, or an expensive, high quality laser printer.

The other benefit of a network is that you can run Internet services over your local network. This is useful for training people to use these services without the need to connect to the Internet. It's also useful if you want to develop web sites, along with complex server-side functions, because you can run the system over the network and perfect its operation before uploading it to a live web server.

More than anything, setting up a network, and enabling different services on the network, is an excellent way that a person can improve their knowledge of how computer systems work. This is because networking requires the interaction with hardware, as well as software, in order to get things working. You don't have to learn programming to do this (although it might be useful). But the requirement to interact with the system at the command line, as well as using graphical tools, improve the skills to interact with the computer. Whilst at the same time the need to understand how various service work improves a person's use of networking services generally.

So, there are many benefits of setting up a network. At the general, it allows you to back-up your laptop in case it gets stolen. At the more complex level, the process of setting up a network will improve the skills of those using it.

Further Information

Salvage Server Project:

To keep up-to-date go to the web site:

Salvage Server Project
http://www.fraw.org.uk/salvageserver/

Useful books:

Red Hat Linux Survival Guide (Red Hat Press):
Excellent and brief guide to setting up simple servers using Red Hat Linux. Covers every aspect from NIC installation to configuring services. Cost around £30/US$40.

Upgrading and Repairing PCs (Linux Edition) (QUE):
As good as all those other book on repairing and upgrading PCs, and setting up networks, but this one specifically looks at the hardware issues in relation to the use of Gnu/Linux. Costs around £44/US$60.

Networking for Dummies (IDG Books):
Feeling stupid and insecure? – try this. Whilst completely useless from the Linux point of view, it provides some useful details on configuring Windows machines to access networks. Cost around £19/US$20.

Linux for Dummies (IDG Books):
Still feeling stupid and insecure? – try this. Whilst good on basics of client installation, it doesn't do much on servers. Cost around £24/US$25

The Salvage Server Project has been developed by the Free Range Network to promote the use of redundant IT equipment as a resource for community and grass roots campaigning organisations. This report has been produced to support the work of the project, and is made freely available to encourage the objectives of the project.