Discovering the Internet

Version 3.0

30th January 2012

Contents

Introduction

First of all, the conditions of use. You can mirror this netbook, and print this netbook. You cannot remove this notice, pretend you wrote it, or charge money for it.

This was originally written some years ago (1999!) and was an attempt at producing something like the much more famous "Zen and the Art of the Internet". Today it is somewhat dated in many places, and is probably far too technical for many people to take seriously as an introduction to the Internet. But some might find parts of it interesting from a historical perspective. The intention is not to update the book except to correct obvious mistakes.

The Internet is far older than some people would believe — it is getting on for 30 years old. And of course people new to it get all evangelical about how great "the Information Super-Highway" is, or how democratic "Web 2.0" is. Which of course grates on the nerves of anyone who has been using the Internet for more than a couple of months. As to whether it is "cool" or not, it depends on what you think is cool, but there is also a huge amount of incredibly boring stuff out there, and a moderate amount of useful stuff.

This document will gradually build up to the more interesting services, covering some useful technical information, and some essential, but boring Internet services first. I am one of those boring people who think you should get to grips with the basics, before you should go onto the more interesting bits. If you really don't like that idea, skip the sections that you don't like the look of, or find a better guide to the Internet.

If you are at a University, or a large company, it is most likely that *you* don't pay anything to use the Internet, but your University most certainly will (or the organisation that funds your University). Probably a surprisingly large amount of money, as the Internet connections a University needs tend to be of the very fast (and thus expensive) kind. Those who have their own personal connection to the Internet will definitely know that it costs money to be online. But you do not normally pay any "hidden" charges — once you have paid for your connection, that is it, unless you intentionally buy something on the Internet.

Similarly with your own Internet connection, it is normal these days to just pay a flat charge per month for "as much as you can eat" — providing you are not saddled with one of those ISPs that seem to think that limiting usage is fair play. Usually such limits are very well hidden too.

Some History

This section is probably not that important to an understanding of the modern Internet, but some of you may be interested ...

The Internet has evolved over time, and has incorporated aspects of other computer networks but it began with ARPANET. Many people will be surprised that the Unix operating system had nothing to do with the early history of the ARPANET, although at the time, Unix was only running on very small computers. And of course Windows was nowhere to be seen.

ARPANET

After the Soviet launch of the Sputnik satellite in 1957, the US military set up the Advanced Research Projects Agency to fund research in things sometimes only vaguely related to military matters. Originally, ARPA funded research by individual corporate researchers, although in 1962 it began to fund academic researchers.

One of the original ARPANET engineers has commented that the purpose of the US military was to fund ARPA, whose purpose was to fund research. Over the years, ARPA has funded many projects in computer science research, many of which had a profound effect on the state of the art. None of the projects had such a profound effect as the ARPANET project.

In 1962, the Rand Corporation published a report written by a Paul Baran, entitled "On Distributed Communications Networks" — the first of many. This report recommended the establishment of a communications network with no obvious central control, and where surviving nodes could re-establish communication with each other after the destruction of a number of nodes. He also recommended the establishment of a nationwide public utility to transport computer data, using "packet switching" to establish a "store and forward" network. At least one of his papers was secret, and the others were not widely circulated.

Donald W. Davies (an UK researcher) also did work in this field at roughly the same time, and is credited with the invention of the term "packet switching".

Dr. J.C.R Licklider (or "Lick" as he asked people to call him) was aware of Baran's work through his military contacts — he worked for ARPA from 1962 (as the head of "Information Processing Techniques Office") with an engineering, and physiological psychology background. Lick was interested in how computers (and computer networks) could be used to help people communicate, and how computers could help people think. He and Robert Taylor wrote "In a few years men will be able to communicate more effectively through a machine than face to face". His vision attracted others involved in computer research, and meant that from the start, a computer network was

thought of something allowing people to communicate rather than just computers communicating.

In October 1967, ARPA announced that it was planning a computer network to link together 16 research groups at large US Universities, and research centres, and the competitive tendering began in the summer of 1968. In January 1969, Bolt, Beranek and Newman (BBN) in Cambridge, Massechusets was awarded the contract for establishing the network.

The plan was to deliver four Interface Message Processors (IMPs, which were Honeywell DDP 516 minicomputers) to four centres. The IMP's were the interface between the ARPANET, and each of the centre's main "host" computers. Each centre had it's own responsibility in the project, and different host computers. The details are listed below :-

- University of California, Los Angeles (UCLA). Running the SEX operating system on an SDS Sigma 7, this site was responsible for network measurement.
- Stanford Research Institute (SRI). Running the Genie operating system on an XDS-940, this site was responsible for network information. It was often known as NIC, and was at one time the organisation that assigned network addresses.
- University of California, Santa Barbara (UCSB). Running OS/MVT on an IBM 360/75. This site provided expertise in Culler-Fried interactive mathematics.
- University of Utah. Running the TENEX operating system[1] on a Digital PDP-10. They provided expertise in graphics (in particular, hidden line removal).

From the beginning of the project, things were left a bit loose with the expectation that the research groups would take some of the initiative. The research students involved in the project at all four sites formed an informal "Network Working Group", and started to discuss various technical aspects — even without detailed information from BBN.

Dave Crocker mentions that they were very nervous of offending the "official protocol designers", so when notes started to be written they were published under the title "Request For Comments"[2]. Possibly one of the most important aspects of the early RFC's was the insistence on complete openness — RFC's were allowed to contain almost any subject provided that it had something to do with the network, and they were not held by the NWG as the "official standard". In addition, the NWG encouraged publication of

[1] Before Unix took over as the most important operating system on the Internet, the various operating systems running on Digital PDP-10's (TENEX, TOPS-10, TOPS-20, ITS, WAITS) were enormously important.

[2] One of the things that remains the same today. All standard protocols, suggestions for new protocols, or details of protocols that haven't made it, are published as RFC's. Not all RFC's are dry technical specifications though — such as the RFC that specifies how to use carrier pigeons as a method of computer networking.

unpolished RFC's in the belief that rough ideas are sometimes as useful as fully worked out protocol standards. They also encouraged the free distribution of RFC's (i.e. they weren't kept private) — a practice that continues to this day.

In February 1969, BBN supplied the research groups with some technical details, and the Network Working Group began working on the nuts and bolts of how the network was going to work — both how the IMP-host interface was going to work, and how the simple applications were going to work.

The first IMP was due to be delivered to UCLA on the 1st September, 1969, and the team there expected some extra time to complete the necessary software (1st September is a public holiday in the US, and there were rumours of timing problems at BBN's end that may have delayed delivery). In the end, BBN delivered the IMP on the 30th August 1969, causing a panic with the software writers. BBN delivered the second IMP to SRI at the beginning of October, and by the 21st of November it was possible to demonstrate a telnet-like connection between the two host computers to senior ARPA officials. The net had come "alive". The first two "applications" to work between two host computers on ARPANET, were a terminal connection program (telnet), and something to move files between the two hosts (ftp)[3]. Note the lack of electronic mail (which was first implemented by transferring messages as files using ftp into special areas, before a new protocol was implemented).

After the first four sites were connected, other sites were connected to implement ARPA's original intention of 16 connected research groups. The next 11 include some names that have contributed enormously to the Internet, and they are all listed here — BBN, MIT, RAND Corp, SDC, Harvard, Lincoln Lab, Stanford (the University), University of Illinois, Case Western Reserve University, Carnegie Mellon University, and NASA-AMES.

At this point, BBN came up with a simpler, slower and cheaper version of the IMP — the TIP (or Terminal IMP). The growth of ARPANET continued beyond the original intention.

Date	Number of Hosts
1971	15
January 1973	35
September 1973	40*
1977	111
1983	4,000

* Including a slow link to the UK, and Norway.

[3] Although the details of *telnet* and *ftp* have changed considerably, they still remain today basically the same as the early implementations.

At the First International Conference on Computer Communications, which was held in Washington DC in 1972, delegates from all over the world were treated to a demonstration of the ARPANET. They also discussed the need for a common set of networking protocols, and the Internetwork Working Group was set up. It was also realised that networks such as ARPANET, and similar networks could be inter-connected, and with the use of the same networking protocols, it might be possible to link a number of individual networks into something that could be viewed as just one large network. It was the start of both the name "Internet", and the start of what the Internet is today.

The ARPANET Completion Report pin-pointed the popularity of email as the most surprising service by the pioneers, and the acknowledgements of Guy L. Steele's book "Common Lisp" indicated why. Lisp is a programming language well suited to, and well used by AI researchers, and as it happens many of those AI researchers have a tendency to tinker with the language they work with — by the time Common Lisp was being worked on, there were at least a dozen popular varieties of Lisp in use. Common Lisp was an attempt (and a successful one) at bringing together the varieties of Lisp into one standard — agreeable to the majority (or at least palatable). In his acknowledgements section, Guy suggests that Common Lisp would have been impossible without ARPANET's email facilities. A mailing list was set up, where the issues at stake could be argued about from day to day — in excess of 3,000 messages resulted, varying in size from one line to 20 pages.

ARPANET made possible collaborations between people who were thousands of miles apart.

ARPANET did have one very big disadvantage — it was difficult to get connected to as it required "political connections", and a large amount of money. Due to the difficulties, CSNET was set up by NSF (National Science Foundation) to provide an ARPANET to those who couldn't get connected to the real thing, and it also proved to be very popular. It also extended the community of Internet users to people other than computer scientists.

As ARPANET was being phased out, NSF then funded the NSFNET which served as the main US backbone for the Internet, until the US government disbanded it and allowed commercial Internet providers to fill the place.

The "Poor Man's ARPANET", Unix, UUCP, and USENET

At the same time that ARPANET was being developed, another major factor advance in the state of computer science was being worked on — the Unix operating system[4].

[4]I'm sure that there are computer scientists who would violently disagree with that statement, but even today, the vast majority of operating systems built by operating system researchers are semi-Unix compatible.

Although the details of that development are beyond the scope of this document, there are a few things that are very important.

The early days of the development of Unix were not very auspicious. When AT&T pulled out of the Multics operating system project, the researchers at Bell Labs lost their interactive computing facility. Ken Thompson, Dennis Ritchie, and some of their colleagues began looking for a way to implement their own facility, and started work with an obsolete Digital PDP-7 computer aiming to use some of the ideas they had encountered in the Multics project.

After a couple of years (and moving to a larger Digital PDP-11), Unix had become a viable operating system which was peculiarly suitable to software development. It started to spread through AT&T, although it was not sanctioned by senior management, and improvements (and bug fixes) began to make their way back to Bell Labs. One manager (the head of the Computer Planning Group) did notice Unix, and tried to make it the internal standard. He also pushed it to engineers who were buying minicomputers for telephony purposes, and were planning to write their own operating systems, and set up the Unix Support Group to supply Unix to these groups, and support it. This essentially created two streams of Unix within AT&T — the USG version, and the Research version.

Unix also began to find it's way into computer science departments at Universities — it ran on relatively cheap hardware platforms, and was very cheap itself (AT&T wasn't allowed to sell Unix, and so "gave" it away to educational institutes). It was particularly valued however because it was a small operating system, and was easy to tinker with.

As part of the means to automate maintenance of large numbers of Unix systems within AT&T, Mike Lesk created a suite of programs collectively called UUCP (Unix to Unix copy). This allowed Unix systems to talk to each other using fixed lines, or phone lines, and to distribute information to and from each other. People eventually realised that such a facility could be used for transferring email, and the UUCPnet (as it was sometimes called) began[5].

In 1979, three graduate students (Tom Truscott, Jim Ellis, and Steve Bellovin) at Duke University, and the University of North Carolina started to build a computer network to link computer systems at the two Universities. Using UUCP, and two hand-built 300 baud modems (*very* slow in comparison with the speeds today), they built up a very primitive system to allow "news articles" to be shared between the two systems (and very soon three).

Although *very* much smaller than today, USENET in 1979, and 1980 was recognisably the same as the Netnews of today, with newsgroups where articles were intended to be

[5]The email addresses were not particularly friendly though, as to reach someone, you had to know the UUCP name of every intervening system, which led to email addresses such as *cs9h6msm!swanpyr!ukuug!amsvax!uunet* (which is incidentally an invented email address, which may bear some resemblance to my 1987 email address). Sometimes, you would quote several possible email addresses to give several possible paths to your machine. It wasn't an ideal solution.

on a particular subject. And of course, it grew, although not as quickly as the pioneers expected (due to the lack of appropriate modems). It became a lot more popular when the University of California at Berkeley (UCB) joined in, and unofficial links between USENET and the ARPANET were established (although residents on USENET often felt as though they were second-class citizens of the net).

Other Networks

The Internet was not the only computer network around of course, and during the 1970's and 1980's, competing computer networks sprang up all over the West. JANET was the UK academic computer network, and BITNET evolved "because it is time" (it was cheaper than the Internet) are just two examples. All of these different networks talked amongst themselves in different low-level protocols (languages), and the need for moving electronic mail between the networks pushed the development of gateway machines. These computers were connected to two or more different computer networks, and provided a clumsy means of transferring email. For instance, a number of years ago, when I had to send some email to the Internet address "linux-activists-requests@niksula.hut.fi", I had to use "cbs%uk.ac.nsfnet-relay::fi.hut.niksula::linux-activists-request". The art of knowing all of the tricks for transferring email between the various email gateways is luckily now unnecessary.

Due to the open nature of the Internet, and the increasing use of Unix on Internet connected machines, it was possible to use relatively low-cost computers to provide an Internet connection, and to easily write new Internet services. This meant that the growth and size of the Internet began to far outstrip other computer networks, and sites connected to other computer networks began to also connect to the Internet.

Freedom and Censorship

As the Internet has its principle origins in the academic world, it has a tradition of freedom of information with little or no censorship. The academic world thrives on the free exchange of ideas and information, and would be stifled without it. With the Internet now available to almost anyone with a computer, there is increasing concern about the kind of information that can be found, and some are pressuring for some kind of censorship.

There is of course material on the Internet that children should not see, and material that most adults probably don't want to see. It includes far-right propaganda, activist literature (such as extracts from the Anarchist's Cookbook, and bomb-making instructions), recreational drug information, and pornography.

There are reasons for allowing dodgy material on the Internet, as banning it would simply drive it underground and whilst it is visible, it is possible to keep an eye on

what is happening. Where I work, academics track the behaviour, and attitudes of far-right groups by checking out their propaganda. If the Internet is censored, there is the problem of just what to censor — the Internet is international, and standards for acceptable material vary. Do we want an Internet where the only material allowed, is the blandest trash allowed by the most restrictive regimes, or the kind of material that is suitable for young children to view ?

Besides which there are software products designed to work on your machine that will block access to known adult content sites. Some will even block access to sites that contain words in a censored list. Although not perfect[6], they are certainly suitable for preventing the majority of teenagers from looking at sites that their parents think that they should not. Of course, there are a small number who will work out how to get around the restrictions but this is likely to be smaller than most people think.

Electronic Communications

Through the rest of this document, you will come across a variety of different ways to communicate with other people on the Internet. All of these methods involve informal written communication — although it is all written, it has much more in common with normal spoken conversation.

It can be difficult to remember that not everyone on the Internet uses English as their first language, and just because someone communicates with many misspellings, and bad grammar doesn't mean that they are stupid — before you start criticising their English skills, just think how good you would be at communicating in their first language. It is also worth trying to avoid using cultural specific phrases, or slang as they may not be understood by others.

It seems that some people find it easier to drop into insult mode with electronic communications than in ordinary life — or is it that I'm used to polite company? If you are such a person, you should be careful, as if you take it to extremes, the person you are insulting could well complain about what you are doing. Electronic harassment is as illegal as any other kind. At least in civilised parts of the world.

In addition different online communities tend to have different conventions on how to communicate. Some places very strongly disapprove of "top quoting" (where you stick a reply to an electronic mail at the beginning of the email with the remainder being included) and others do not seem to understand why anyone would want to edit down emails and quote at the "appropriate" place. The obvious answer is to use the style appropriate for the community that you are with at the time . . . exactly the same as you would do in real life.

[6] At one point, one service providor prevented any meantion of the placename Scunthorpe, which proved to be a bit embarrassing :-)

Emoticons

In a normal spoken conversation, a good deal of what is said, is communicated with non-verbal signals — facial expressions, body posture, etc. As an aid to this sort of conversation, people use *emoticons*, a table of which follows. You might try tilting your head through 90 degrees to the left, and the symbols might suddenly become a lot more meaningful — even if it does make you look stupid :-)

Emoticon	Description
`:-)`	Standard smiley. Indicates that humour is intended.
`:-(`	Something makes you unhappy.
`'-)`	Winking.
`;-)`	Sardonic incredulity.
`:-x`	A kiss.
`:)`	A shortened standard smiley. For the lazy.
`:-}`	Grin.
`:-]`	Smirk.

As with many good ideas, there are those who take it too far, and a complete list of emoticons would be very long — most of which do not serve any useful purpose.

Flames and Flaming

The old-fashioned phrase (which you will still encounter) for a slanging match in public electronic communication channels, and in particular in Netnews is "flaming". Some people think that there is some kind of art to flaming, but most dislike seeing two or more adults behaving like spoilt children — you often find people saying "take it into email" in response.

Spam

Or "Unsolicited Bulk Email" (UBE).

If you have an email address, it is almost certain that you have received Spam — an email asking you whether you would like to buy something, or some message along those lines. The main thing is that you didn't ask for it. One such message is just mildly irritating, but what if it is 10 per day ? Or 100 ?

The slimy, degenerate scum-balls who send this junk regard this as a valid marketing technique, and prefer it to other similar marketing techniques (such as cold-calling, or

junk FAXes) as it is very cheap — for *them*. However, if you pay for your own Internet connection, *you* have to pay for each spam message that you get. In addition, it will take you time to deal with the spam.

As an example of how unprincipled these cheap-skate, dim-witted spammers (do you get the feeling that I don't like spammers ?) are, they use various tactics to hide their activities. For instance, they like to route their spam messages through innocent third party machines — damaging the reputation of the relevant organisation, and using computer resources that they haven't paid for, and haven't been given permission to use. Even worse, they frequently send out emails with forged reply addresses so that most complaints end up in the wrong place, and if the forged email address is yours, you will end up with *thousands* of complaints.

The most recent trick that spammers use is to infect machines with viruses and use a huge network of machines owned by innocent people to send out spam.

People who run large mail systems (myself included) spend a large amount of time trying to block spam — both spam delivered to their users, and spam directed through their servers to users at other sites. To some extent this has been successful, but spammers adopt new tricks every time an old method is blocked off.

The Internet Provider, Intranets, the Matrix and Cyberspace

Now for the shock — there is no such thing as the Internet. The Internet is really a collection of cooperating networks (or Internet providers) that are all interconnected, and use the same basic networking protocols. For instance, where I work we have a private Internet which is connected to the UK academic Internet by the JANET organisation, and this organisation maintains links to other parts of the Internet (and via these other parts to all of the Internet).

One of the latest buzz-words is that of the "Intranet", which is essentially a private Internet confined to within an organisation to provide services to the people within that organisation. It is not anything new — before the University where I work was connected to the Internet, it had its own private Intranet. It also means using Internet applications and services to provide information privately to the Intranet (or just people within the organisation). Many of the older standards for Internet applications make it plain that private Internets are called "Internets", not "Intranets".

The "Matrix" is an almost historical term which refers to computers and people who can communicate electronically in some manner — usually via electronic mail, but do not have a true Internet connection.

Some people refer to the Internet as cyberspace (the same people tend to stick cyber in front of everything). Although there is such a word, it refers to something that the Internet is not (at least yet). The word was first used by William Gibson in his Cyberpunk

books, and is used to describe a virtual reality interface to a network like the Internet. We've got the network, but not the interface (yet).

The World

Previous versions of this guide used to have a fun looking map here which showed what parts of the world were connected to the Internet and what parts were indirectly connected, and places where there was no connection at all. As you can imagine this has effectively become completely useless as pretty much every country has an Internet connection.

As you can see, apart from small isolated islands in the rest of the world, it is Africa which tends to lack connectivity although the situation is improving very rapidly. The connectivity map for the previous year (1995) showed much larger areas of the world without Internet access, and only a couple of countries in Africa had it. I suspect that by the end of the century, it will be possible to get an Internet connection in any country in the world.

This doesn't mean that everyone in the world will be able to get connected. A peasant farmer in the backwaters of Mongolia is not likely to have a phone, never mind be able to buy a computer so that they can become connected. There are of course people in this country that cannot afford the equipment to get connected, although if they live in a big city they can always visit a CyberCafé (a cybercafé is a public place where you can use computers to use the Internet. As you can guess from the name, you can usually also get coffee and the like). In addition there are people who just don't *want* to be on the Internet.

It is currently fashionable to speak of the information "rich" (those who have access to the Internet), and the information "poor" (those who do not). Various initiatives have been started to make access easier for people — in the UK, the government is connecting every school in the country to the Internet, and the Swedish government is implementing a system that will give every citizen their own email address.

The Speed and Reliability of the Internet

When you are using the Internet (usually with a Web browser), speed very quickly becomes an issue. You will notice that some places are blindingly fast, with pages that show up almost before you can blink. Other pages take *ages* to appear, and may not appear properly at all. It is very difficult to predict just how fast the Internet is likely to be due to the number of factors involved.

Firstly, there is the number of people using "your" Internet connection (unless you have your own personal connection). The more people there are "surfing" the Internet at

your location, the slower it is likely to be. In addition, certain local conditions at your site can make a big difference — servers temporarily overloaded with other work, lots of work travelling on the network, etc.

Of course when you are using the Internet, you are also using a server at another site. The server at this site may have just a few simultaneous users, or thousands. The server may be a very fast machine, or it could be slow. The site could have a fast link and not much going on, or it may have a slow link with many people moaning about how slow the Internet is.

There is also the Internet itself. You could be making use of a well configured, and fast part of it which happens to have relatively few people using it, or you may be using a part that is suffering. And as the Internet grows, parts that were fast with few people last year are likely to be slow with too many people this year.

The diagnosis of "why the Internet is slow this morning" is a complex black art known to relatively few people.

In a similar fashion, faults that occur could be occurring anywhere on the Internet. Sometimes a link to a server is so slow that it appears that the server isn't there. Sometimes the server has been switched off for more disk drives to be plugged in. In either case, you will get some kind of error message.

Due to the size of the Internet, and the number of servers that it is dependent on, it is quite possible that there is never (and never will be) a time when the Internet is *really* working properly. There is always someplace that has a problem, and some days all the problems seem to be where you are looking.

Technical Details

The Hardware Layer

Essentially, just what kind of hardware connects you to the Internet does not matter that much (as long as it is compatible with what you are connecting to). At the low-end, a hardware connection could simply consist of a fixed phone line running at a speed of approximately 56 thousand characters per second, moving through various possibilities to high speed ATM, or SDMS connections running at hundreds of millions of characters per second. Apart from the speed (and the reliability[7]), the exact details of the hardware only concern the networking engineers who run the connection.

In addition to traditional fixed Internet links, there are several ways to use a modem attached to a computer — either connected to an ordinary analogue phone line (the kind you are most likely to have in your home), or a digital ISDN line. These methods connect you to the Internet as and when *you* are using it. This is done either using the SLIP, or PPP protocols.

It is impractical to use a phone line connection to the Internet to provide Internet services to others[8], and so I don't regard it as a proper Internet connection. This of course has advantages, as a full Internet connection really requires one or more Internet experts to be around to check on the security of your systems. It is also the only practical way for an individual to use the Internet from home — a proper Internet connection is far too expensive[9].

Both cable modems and ADSL lines offer faster Internet speed, and they are always-on in a similar way to full Internet connections. You could if your ISP allowed it, run your own servers using a cable modem or ADSL line.

[7] There are some places where the link to the Internet is so important to an organisation that it is essential to keep the link available. Even to the extent of keeping a spare link available in case the main one is disabled in some way.

[8] If you are a US reader, you may be suprised to learn that most European phone companies charge for local calls — a typical rate is about 1p per minute.

[9] At least for this year.

Bandwidth

You will often come across this word, which basically means the speed of the connection, or as a vague term referring to how much "speed" an application needs. Some common bandwidths are listed below (bps is bits per second) :-

Line Type	Speed
"slow" modem	14.4Kbps
"fast" modem	56Kbps
ISDN	64-128Kbps
leased line	64Kbps
T1*	1.5Mbps
T3*	45Mbps
SDMS	4-34Mbps

Those connections marked with "*" are usually only available in the US.

Note that a "fast" modem is likely to be quite a bit slower than an ISDN line due to the variable quality of ordinary phone lines. And the quoted speed is partly due to compression of the data going through the modem. If the data is already compressed, the modem will appear to slow down (as the compression becomes less effective, and if the data is very suitable for high levels of compression (such as pure text), the modem will appear to speed up (as the reverse happens).

Routers

You may come across routers being mentioned as being critical to the operation of the Internet, and they are. When your machine sends something on the Internet, it doesn't know where the destination machine is, but routers do. Or at least they know a router that does. Your computer is told to use a particular Internet address whenever it doesn't know where to send some information, and the router at this address will deal with forwarding it.

If you use a Unix system (some other systems also have something similar), you can use the *traceroute* program to discover how packets sent to a particular machine bounce from one router to the next, and so on until they reach the final destination. It can be quite a surprise to see just how many machines are involved — typically 10 to 20 "hops". If something goes wrong with the routing, it may take an infinite number of hops to reach a destination, but the networking software decides that enough is enough after about 30 hops.

Due to the size of the Internet at present, routers are pieces of equipment that are under enormous amounts of strain. In addition to being expected to work *very* quickly, some of the larger routers are expected to hold information on an *enormous* number of routes. As with any piece of equipment that is working to its limit (or beyond them), routers do sometimes either fail, or work incorrectly. This results in some of the problems on the Internet.

Most local area networks use a networking technology called *Ethernet*, which works by sending a "packet" of information to every machine on the network[10] (although in normal operation, only the destination machine will listen to that packet). This is called a *broadcast* network, and as you can imagine, if you have a large network with a large number of machines spread over a large area it can be quite inefficient. The way of dealing with this inefficiency (and to make networking several different buildings, or floors in a building), the whole network is split into a number of separate physical networks with routers joining the networks together. If the router is told that a packet should be on a different network, it will transmit it to that network on behalf of the machine that is trying to talk to the machine on the different network.

The machine itself has to know that a particular machine is on a different network, and so packets that it wants to send to such machines have to be sent through a particular router. The most common way of doing this, is to tell machines that addresses that don't match a particular set of addresses are remote, and all traffic for such remote addresses should be sent via a particular router — the *default route*.

Firewalls

In the past (and even when the first version of this guide was published), only very organisations, or paranoid organisations would have a firewall installed. But with "always on" connectivity (such as ADSL, or Cable Modem) becoming more widely available, and the immense increase in computer break-ins, firewalls are becoming much more common. Even to the extent that normal home users are getting them — and home users with "always on" connections definitely should have them.

There is a wide variety of different devices which can be called a firewall, varying from dedicated computers to shield an entire organisation down to software packages that can be installed on the average computer at home. Although they vary widely in sophistication and speed, their aim is the same — to block network traffic that should not be coming in (or going out), and to permit traffic that should be allowed out.

[10] Actually, modern Ethernet hardware doesn't always work in quite this way.

IP Addresses

At the lowest level of communication[11], the Internet uses the Internet Protocol to send packets of information from machine to machine (and from machine to lots of machines). Each machine on the Internet has an IP address so that other machines that wish to send packets to it can do so. In the "old" days, a machine had one IP address allocated to it, and it always used that address, but today there is a certain class of machines which are told what IP address to use every time they are turned on, and this address may change from day to day — such as machines connected to the Internet via a modem connected to a commercial Internet Service Provider.

An IP address consists of four octets (8-bit numbers — from 0 to 255), part of which is a network address, a subnet number, and the a host address. The network number is either issued by your Internet Service Provider[12], or is obtained from one the top-level organisations that allocate network numbers to ISP's, or very large organisations. In the past, it was possible for smaller organisations to obtain their own IP addresses without going through an ISP — in fact it was common for organisations to get an IP network number even if they did not have an Internet connection, but planned to get connected in the future.

With the introduction of *Classless Inter-Domain Routing* (CIDR), an organisation can get an appropriate network number — i.e. one that allows for an appropriate number of connected machines. CIDR also introduced a new notation for indicating a network number; one that indicates how large the number is. For instance, the University of Portsmouth was allocated *148.197/16* as a network number. The *16* indicates that the network number is 16-bits long — two octets (i.e. every machine in the University of Portsmouth has an IP address starting with *148.197*). You may hear people mentioning class-based network numbers — class A, B, and C. This was the scheme for allocating network numbers in the "old days". Class A addresses (CDIR: *x/8*) allowed up to 16,387,064 machines; class B addresses (CIDR: *x/16*) allowed up to 64,516 machines; and class C addresses (CIDR: *x/24*) allowed up to 254 machines. This wasn't flexible enough — many organisations were allocated class B networks, although they didn't need quite that many, which is why CIDR was introduced.

The *subnet* part of an IP address is just an organisation's way of lengthening the network number. This is done so that an organisation can split it's network up into lots of little networks (subnets), which may be to increase network efficiency, or more commonly because the organisation needs to link together (with routers), physically separate networks. The organisation chooses how much it increases the network number by — the larger the subnet part, the more subnets it can have, but the fewer machines it can have

[11] It isn't *quite* the lowest level of communication, but the lower levels are pretty much tied to the hardware used.

[12] Actually, most domestic machines obtain an IP *address* from their ISP. It is only organisations with a pernament Internet connection that get a network number.

on each subnet.

IP isn't a totally static standard, although it doesn't change that frequently — the last change to version 4, was done to increase the maximum number of machines from *255* to the current maximum. The next change is to version 6 (version 5 was allocated to another protocol by mistake, and is avoided to prevent confusion) and some test networks are already using it. IP version 6 (sometimes called IPng) allows for a *much* larger number of IP addresses as the IPv6 address is 16 octets long — several thousand for everyone who lives on this planet, and has a number of interesting new features.

The Next Level

You will hear that the Internet uses *TCP/IP* as it's networking protocol, which is true, but not the whole truth. The *IP* part of *TCP/IP* is explained to a certain extent in the previous section (well ...I do tell you what it is). The *TCP* part of *TCP/IP* is one of a number of different protocols which are used on top of *IP* for various different purposes. The following table shows a list of the protocols specified in the file */etc/protocols* on my machine (don't worry, you won't need to remember them, or understand them) :-

Name	**Number**	**Comment**
ICMP	1	Internet Control Message Protocol
IGMP	2	Internet Group Multicast Protocol
GGP	3	Gateway-Gateway Protocol
TCP	6	Transmission Control Protocol
PUP	12	PARC universal packet protocol
UDP	17	User Datagram Protocol
IDP	22	Unknown
RAW	255	Raw IP interface.

The most commonly used protocols are *TCP*, *UDP*, and *ICMP* — at least at the level of Internet services that you are likely to use directly. Some of the others are used to keep the infra-structure of the Internet ticking over, whilst the rest remain as the after effects of obscure and forgotten experiments. Just because my computer just list the ones above, doesn't mean that others don't exist.

Most of the services that you use will either use *TCP*, or *UDP* to communicate. *TCP* is usually used when a reliable but relatively slow communication channel is needed, whilst *UDP* is used when a fast, but sometime unreliable service is needed to send (or receive) small chunks of information.

Ports

All Internet services have an associated port number to connect to when making an Internet connection, which simplifies the decision of which service daemon[13] to run at the other end. If you have access to a Unix system, you can browse through the file */etc/services*, which contains a list. I am not going to include such a list here — my Unix system lists well over a hundred, and there is no reason why thousands more cannot be accommodated.

Most of the services listed in this file are chiefly for testing, or are the remainder of services that are no longer used.

The world-wide-web *usually* uses port number 80, and mail uses port number 23.

Host Names and Domain Names

IP addresses work quite well for the computers themselves, but do not work quite as well for us humans. In the earliest days, each machine was given a name such as *MIT-AI*, and a list of the names and IP addresses was kept in a file called *HOSTS.TXT*[14]. This worked quite well whilst there were just a few hundred machines on the Internet, but rapidly proved unworkable as the Internet grew. A distributed database was organised where each organisation managed its own IP addresses and names, and that organisation's part of the database was used to look up IP addresses from the names.

Each machine on the Internet has its own Fully Qualified Domain Name (FQDN for short) which looks like *ranger.hum.port.ac.uk.* (the last "." is quite important). The name tells you a few things about the machine as it is very carefully structured. The least important part is at the left side — *ranger* which is simply the name of the machine, which usually has little or no meaning, and is called the *host name*.

Some host names do have particular meanings, such as *www.hum.port.ac.uk*, which is the name of the Faculty of Humanities Web server, and is another name for a machine called *hobbit.hs.port.ac.uk*. Giving services names like this, allows a service to have the same address for years without the user being aware that the machine providing that service may have changed any number of times. For instance, the Faculty of Humanities Web server at *www.hum.port.ac.uk* has recently been moved from *hobbit.hs.port.ac.uk*, to

[13] They are called daemons under the Unix operating system, and the name works as well as any other. Interestingly, ignorant christian fundamentalists take exception to the use of this word, mistaking it for demons. According to my information, demons work for the devil, and daemons work for God — if you believe in such things.

[14] It is still possible to download the latest version of this file from the relevant FTP server, but I suspect that it is *very* out of date.

warlock.hs.port.ac.uk, and back again (this isn't *quite* as stupid as it sounds) without the users being aware of the changes[15].

The remaining part of the FQDN is the *domain name*, and is subdivided by the "."'s, and is arranged with the top level at the right, and the bottom level at the left. For instance, using the example used at the start of the previous paragraph, "." at the extreme right is the root domain into which all other domains fit. The *uk* to the left of this indicates that the machine is in Great Britain, and the *ac* indicates that the machine is in an academic institution. The *port* shows that the machine belongs to the University of Portsmouth (*portsmouth* could be used instead), and the *hum* means that the machine is in the Humanities domain. Each domain ("."*, uk*, *ac*, *port*, and *hum*) has its own domain name server which is used by the rest of the Internet to find out about machines, or domains with its domain.

Most top-level domains use the ISO country code, but as the Internet originated in the US, many US Internet sites use one of *edu* (the US equivalent of *ac*), *com* (for commercial organisations — increasingly this is used for international commercial organisations), and *mil* (for the US Military), although there are moves to try and get them to move into the *us* domain to make them look a little more like the rest of the world. In addition, Great Britain uses *uk* as its top-level domain name, for the simple reason that the JANET network did[16], and changing to use the ISO country code (*gb*) would be unnecessarily complex.

The *Domain Name System* translates FQDN's to IP addresses, and also IP addresses into FQDN's, and despite its distributed nature works out surprisingly well. This is especially surprising when you consider the number of machines that the DNS is dependent on, and the fact that distributed databases were a new idea when the DNS was first envisaged. You will often hear the DNS called *BIND*, as the most common piece of software implementing a DNS server is the *Berkeley Internet Name Daemon*.

The DNS does not just translate numbers into names (and back again). It can also provide other information about a domain, or a host in a domain — the "Responsible Person" for a domain, where mail for a host should be sent, etc.

[15]It has since been moved onto *va5-3.iso.port.ac.uk*, and the users *still* didn't know about the change.

[16]JANET also did everything the right-way round. For instance, the NRS name for Hobbit, would have been *UK.AC.PORT.HUM.HOBBIT* (in fact that would be the short-form NRS name, the long-form NRS name would be *UK.AC.PORTSMOUTH.HUMANITIES.HOBBIT*, which is a bit more readable, if a lot more difficult to type.

Some Basic Services

Although these services are not especially exciting, they are the descendants of the earliest Internet services, and are still useful. In some ways, they work better than later services which perform a similar function — for instance *ftp* works better for transferring large files across slow Internet links than *http* (which is the lowest level of the World-Wide-Web).

Ping

This service is so useless for normal Internet users that it is almost not worth mentioning — except that you may hear of it. It is mostly used for testing a new machine connected to the Internet, so that the person who set that machine up can be sure that the most basic part of the networking is working properly.

With Unix (this will also work with Windows 95), you enter a simple command such as the following at the command prompt :-

```
ping 148.197.160.1
```

This tells Unix to try and send a "ping packet" to the named machine — using either a machine's IP address or it's hostname. If the "ping packet" reaches the remote machine, the local machine will tell you that it is "alive", and if not, that it is "unreachable". Not exactly much fun.

You may also hear about something that happened quite recently. Microsoft's Windows 95 has an illegal implementation of *TCP/IP* that allows illegal packets to be sent to remote machines — in particular the ping command can be used for this. Some machines were vulnerable to such illegal packets, and could crash as a result — not something you would like to happen to your mail server.

Terminal Sessions (Telnet)

When the Internet was first introduced, they managed to leave out the ability to connect a terminal to a computer across the Internet, but this was very quickly rectified with the *telnet* protocol (which is also the name of a Unix program). Although we have moved on since the days when a simple terminal connection to a computer was the ultimate man-machine interface, such an interface can still be quite useful — for instance, a large super-computer may only offer a simple terminal based interface to its facilities, but someone who can do something in half an hour on a super-computer instead of three days on a fast PC will be quite happy to live with a "primitive" user-interface.

A sample telnet session is shown below. Bear in mind that it is a session from a Unix machine to another Unix machine, and so may look quite different to a telnet session on your own machine.

```
mike@warlock - ~ % telnet hobbit.hs.port.ac.uk
Trying 148.197.160.117...
Connected to hobbit.hs.port.ac.uk.
Escape character is '^]'.

Red Hat Linux release 3.0.3 (Rubens)
Kernel 2.0.0 on a i586

hobbit.hs.port.ac.uk login: mike
Password:
Last login: Mon Aug 19 19:46:44 on tty1
mike@hobbit - ~ %
Connection closed by foreign host.
```

The bold phrases in the example above were what I typed — although the password I typed to login to Hobbit is not shown here :-) The first line is me instructing my workstation to start a telnet session to Hobbit (ignore the "mike@warlock - ~ %" — this is some additional information for my use). The second line is shown by my workstation, and lets me know that it is trying to make a connection to a machine at a particular IP address, and the third line is my workstation telling me that a connection has successfully been made. The machine called Hobbit is responsible for the messages from lines 6 onwards (excluding that last line) — you can see that it asked me for a user name (I responded with "mike"), and a password (which I supplied correctly), and then showed me a typically customised Unix prompt. At this point, I logged out (by pressing Control-D), and my workstation took over, and told me that the "foreign host" wasn't interested anymore.

File Transfer Protocol (FTP)

Probably just as important as making terminal connections to a remote computer, is transferring files to or from a remote computer. FTP works in a similar manner to telnet in that you will have to supply a user name and a password before transferring files. The following shows an FTP session similar to the telnet session shown above :-

```
mike@earthquake - ~ % ftp hobbit.hs.port.ac.uk
Connected to hobbit.hs.port.ac.uk.
220 hobbit.hs.port.ac.uk FTP server (Version wu-2.4.2-academ[BETA-9](1) Thu Feb 29 15:50
Name (hobbit:mike): mike
331 Password required for mike.
Password:
230 User mike logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 1569
-rw-r--r--   1 mike     mike           52 Oct 14  1995 CMOS.ROM
drwx------   2 mike     mike         1024 Mar 18 22:02 Calendar
drwxr-xr-x  28 mike     mike         1024 Jun  1 18:38 Progs
drwxr-xr-x   2 mike     mike         1024 Jan 26  1996 Test
-rw-r--r--   1 mike     mike         1908 Mar 24 01:51 URLs
-rw-r--r--   1 mike     mike         2597 Dec  6  1995 URLs.blank
-rw-r--r--   1 mike     mike         5202 Aug 10 20:56 banner.ps
-rw-r--r--   1 mike     mike           26 Mar 25  1993 bashrc
drwx--x--x   2 mike     mike         1024 Nov 12  1995 bin
-rw-r--r--   1 mike     mike       168284 Aug  1 20:46 lyxtmpb22193
drwx------   2 mike     mike         1024 Mar 14 23:40 nsmail
drwxr-xr-x   5 mike     mike         1024 Jun 26 20:13 public_html
-rw-r--r--   1 mike     mike        18598 Aug  1 23:32 raquel1.lyx
drwxr-xr-x   2 mike     mike         1024 Jan  4  1996 scenes
drwxr-xr-x   2 mike     mike         1024 Jul 21  1995 src
-rw-r--r--   1 mike     mike        10166 Aug  7 00:07 unix.lyx~
-rw-r--r--   1 mike     mike         1699 Jun 27 23:01 unnamed.fig
-rw-r--r--   1 mike     mike          869 Oct 17  1995 wt.xpm
226 Transfer complete.
ftp> cd mail
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 7
```

```
drwx------   2 mike     mike         1024 Feb 28 01:25 .
drwxr-xr-x  36 mike     mike         4096 Aug 20 20:47 ..
-rw-------   1 mike     mike            0 Aug 25  1995 saved-messages
-rw-------   1 mike     mike         1195 Feb 28 01:25 sent-mail
226 Transfer complete.
ftp> get sent-mail
200 PORT command successful.
150 Opening BINARY mode data connection for sent-mail (1195 bytes).
226 Transfer complete.
1195 bytes received in 0.0228 secs (51 Kbytes/sec)
ftp> quit
221 Goodbye.
mike@earthquake - ~ %
```

All of the text displayed in bold in the session shown above are commands typed by me. The first line is me telling my workstation that I want to make an FTP session to the machine Hobbit. The second line is a comment that my workstation is attempting a connection to Hobbit. The lines from there up until I type "mike", are Hobbit announcing what it is, and asking for a user name. I also have to enter a password. The text from the password line until you see "ftp>" is Hobbit announcing that I have successfully logged in. The "ftp>" text is a command prompt — I am being asked for a command to tell ftp what to do, and it is not Hobbit I am talking to, but my workstation. Once I have told my workstation what to do, it will talk to Hobbit to transfer files, or whatever.

The first thing I do in the session above is to type "ls" to see a list of files on Hobbit. The list of files that is shown is shown in full detail — which I didn't really need, and usually isn't necessary. The name of each file is shown at the end of each line. I then type "cd mail" to change into the "mail" directory, "ls" to list the files in that directory, followed by "get sent-mail" which fetches that file. Finally I type "quit" to end the FTP session.

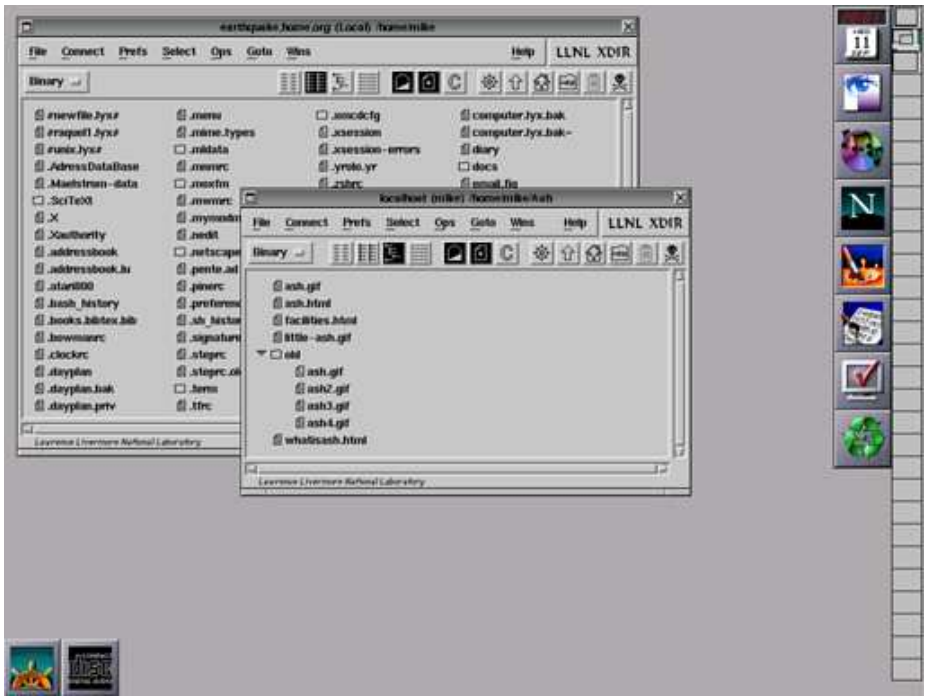
There are many more FTP commands which can be used, although some of them are very rarely used, such as the "tenex" command (to fetch files from a 36-bit TENEX system — almost all of which are unfortunately on the scrap heap), and the "account" command, which allows you to enter a supplementary password for additional security (and something that I have *never* used). Some of the more useful commands are listed in the table below (optional parameters are shown surrounded with "{}") :-

Command	**Parameter(s)**	**Comment**
ascii	N/A	Transfer the file in ASCII (text) mode.
bell	N/A	Ring the bell after a file transfer is complete.
binary	N/A	Transfer the file in binary mode. For *everything*, which isn't
		a plain text file.
cd	directory	Change into a directory on the remote machine.
cdup	N/A	Change into a directory above the current one on the remote
		machine.
chmod	mode filename	Changes the permissions on a file on the remote machine.
delete	filename	Deletes a file on the remote machine.
dir/ls	{name}	Lists the current directory, or the directory/file specified.
get	name {name}	Gets a file from the remote machine. And optionally name it
		on the local machine.
hash	N/A	Prints a hash mark ("#") for every 1024-byte block transferred.
lcd	{directory}	Changes directory on the *local* machine
mget	wildcard	Gets multiple files from the remote machine.
mkdir	directory	Creates a directory on the remote machine.
mput	wildcard	Puts multiple files from the local machine onto the remote
		machine.
open	name {port}	Opens a new connection to a remote machine. The port number
		is almost never used.
prompt	N/A	Toggle interactive mode — asking questions for getting
		or putting multiple files.
put	name {name}	Puts a file from the local machine onto the remote one,
		optionally naming it.
pwd		Prints the name of the directory on the remote machine.
rename	from to	Renames a file on the remote machine.
rmdir	name	Removes a directory on the remote machine. Usually the
		directory must be empty.

Some of the commands in the above table mention wildcards, which allow you to specify a pattern to match the files that you want to do something with. For instance if you want to fetch all of the files in the current directory, you would use the command "mget *". For precision, the ftp wild cards are compatible with globbing documented in the *csh* manual page, but basically, "*" matches any number of characters, and "?" matches any one character.

Of course you don't have to use a plain FTP system like the one detailed above — you probably have access to a graphical one, such as Rapid Filer, WSFTP, or something similar. Just below is shown a view of a graphical FTP program that you are not likely to encounter unless you are using a Unix machine, but does illustrate how most of the

graphical browsers work :-



Although the program illustrated above consists of two separate windows, and the programs that you are likely to use will have just one window (split into two), they do work very roughly in the same way. One window shows a list of local files, whilst the other lists files on the remote system, and you can simply drag files across between windows to transfer them.

Electronic Mail

Electronic mail is older than the Internet, and the standards for Internet mail build on what happened before[17]. The main standard for Internet mail on which others are built, is the Simple Mail Transfer Protocol (SMTP). Before this standard was implemented and available, electronic mail was transferred between systems using the file transfer protocol mentioned in the previous section — an inefficient way of doing things.

SMTP was developed to support the transfer of mail across the Internet, and it is quite possible to send email "in the raw" by directly using SMTP. Sending email this way is either used by people who run email systems to diagnose problems, or by unscrupulous

[17] As an examply the mail headers (you will probably encounter the 'From' and 'Subject' mail headers regularly, but there are more) for Internet mail, and Coloured book mail are suprising similar.

people who are faking email (i.e. trying to send email that looks as though it came from someone or somewhere other than where it really comes from). In addition to being possible to fake, email can also be easily intercepted, and read.

There are solutions to this problem of insecurity — verifiable encrypted signatures for determining whether an email message is genuine[18], and robust encryption of entire messages for keeping your email private[19]. If you have a real need to, you should investigate using encryption (PGP, or Pretty Good Privacy, is a good place to start), but it is far easier to remember to avoid trusting email too much.

Before the establishment of a standard for transferring "multi-media"[20] mail, email was limited to pure text — and 7-bit ASCII text at that (in other words, pure US English text with no allowance for foreign language characters, including the "£" sign). The Internet standard for "multi-media" mail, is MIME, and was designed to be infinitely flexible — so flexible that it is also used for indicating the kind of things being transferred across the World-Wide-Web.

Mailing Lists

As electronic mail became more widely used, it was quickly discovered that not only was it a convenient way to communicate, but that it could also be used to communicate with groups of people with interests in common. Initially, the groups were maintained manually, but software was soon written to automate the process, and the number of mailing lists exploded.

Today, although there are similar means (and in some ways better, or at least easier to use) of communicating with like-minded groups of people (such as NetNews, which is covered in the next chapter), there are still an uncountable number of mailing lists distributed across the Internet. Some mailing lists are private, and you can only join when you are invited, whilst others are public and can be joined by anyone.

When you find the address of a mailing list that you want to join, you should also find some details of how to join the list. This usually involves sending some sort of command in an email message to an email address (*not* the same address as the list itself). One of the major packages for managing mailing lists is called *Majordomo*, and works with commands similar to the following :-

[18] Just copying someone's digital signature won't work — they are different for every message.

[19] Warning, some encryption methods are *not* private — one (the Clipper chip), has a built-in weakness that allows the US Government to spy on electronic conversations, and many don't believe that it will just be governments spying on such conversations. Besides if you want something to be private, do you want your government spying it out ?

[20] Multi-media in this case means any kind of computer data other than pure ASCII — including latin1 text (Western European languages), more complex standards of text (Russian, Chinese, etc.), pictures, movies, etc.

Command	Description
subscribe *list-name*	Subscribes to the list of the relevant name.
unsubscribe *list-name*	Unsubscribes from a list.
help	Sends a list of commands with explanations.

Abuse of Email

The ease with which email can be sent, and the "unreality" of it, can delude some people into sending messages with contents that could be viewed as abusive — excessively abusive email is illegal[21]. Always think before you send an email message — would you say to someone's face what is in your message ?

If you receive abusive email yourself, you can send email complaining about it to the postmaster of the same site — for instance, if the email came from *nasty.person@port.ac.uk*, you should complain to *postmaster@port.ac.uk*. It would be a good idea to include a copy of the offensive message in your complaint. You should also keep a copy of the original message, as there is some extra information with that message that an expert may be interested in.

If you receive abusive email from your boss (which is apparently an increasing problem), you could *try* complaining to your boss, his/her boss, or the postmaster. As to whether any action will be taken is a different matter. At the very least, it does tell you that your boss is an incompetent who really should not be in the job.

Although nowhere near as bad, it can be wrong to send files via email — not small files (for example documents up to 20 pages), but large ones (a complete book). It is unlikely that you will be caught doing this[22], but it can clog up email systems.

Directory Services (X.500 and LDAP)

There is one big problem with email — how do you find out someone's address ?

At present, there really isn't much of a solution as if an organisation publicises email addresses at all (and some don't — either because it will cost someone time to do so, or in an attempt to discourage the spammers), it will probably do so in some non-standard way. Although you can try looking at an organisation's web page (if they have one), there isn't always an answer there.

There is an answer though — directory services[23].

[21] At least in the Uk. Where are you reading this from ? Answers on a postcard please :-)

[22] Unless you do something increadibly stupid such as sending a large file to all your students. That *will* be noticed.

[23] Which are not used *just* to help people find email addresses.

In the OSI network model, X.500 provides directory services, including email addresses. Unfortunately, X.500 isn't really an Internet service, and although there are X.500 directory servers on the net, there are not that many of them. Part of the problem is that X.500 software is very complex to install, and is dependent on another piece of software that is probably even more complex to install.

An alternative known as LDAP (light-weight directory services), which was intended as a cut-down version of X.500 is available. With numerous companies and organisations flocking to the LDAP banner, it looks as though LDAP will be the service of choice for providing information such as email addresses.

Although there are LDAP servers available now, and you can certainly use them (usually through the Web), there are not enough to be certain of finding the email address you want. Just wait a few years :-)

More Services

USENET News

USENET News (which is sometimes called NetNews) is a badly named service[24], that provides thousands of discussion groups, where you can read messages from others, *respond* to messages from others, or *post* totally new messages of your own. Some of the newsgroups are moderated which means that messages are checked whether they are appropriate or not before appearing in the group[25], whilst others are a free for all[26].

Each newsgroup has a name which indicates roughly what the subject the group is about, and where in the hierarchy of groups it is. For instance, *comp.os.linux.announce* is a newsgroup to do with computers (*comp*), operating systems (*os*), the Linux operating system (*linux*), and announcements relating to Linux (*announce*). Not all (or even most) of the newsgroups are about computers — they vary from discussions on keeping cats as pets (*rec.pets.cats*), to a group dedicated to the works of David Eddings (*alt.fan.eddings*).

Before getting involved in some of the discussions going on, you should read the articles in the newsgroup *news.announce.newusers* and *news.groups.questions*, which contain some useful information on how the newsgroups work. In addition, if you are going to ask a question, check with a service like Dejanews (*http://www.dejanews.com/*) to see if your question has been answered before, or check the FAQ for the group.

FAQs or Frequently Asked Questions

Many of the more serious Netnews groups have regularly posted documents called FAQs, which contain useful tips on the Netnews group in question, other Internet resources associated with the Netnews group, and elementary information on the subject that the group discusses.

[24]It is after all, more about discussions than news, although there are some news groups which do have news articles — some of which you will have to pay for (or more specifically, your service providor will have to pay for).

[25]This work is done by a person — how they find the time, I just don't know.

[26]Not *quite* a free for all. You should keep to the "rules" for the group, or you will annoy the others in the group.

These FAQs tend to be a very good source of basic information on something, and usually contain references to more in depth information that the experts believe are the best. It is always worth hunting down any FAQs relevant to what you are interested in.

Many FAQs can be found in the newsgroup *news.answers* (which makes for an interesting read on a rainy afternoon), and there are some Web sites that archive the FAQs.

Recently, FAQs have been showing up as information sources on subjects that have nothing to do with Netnews groups. FAQs have become such a useful source of information, and such a widely understood method of presenting information that they have become a sort of standard.

Campus-Wide Information Systems and Distributed Information Systems

During the 1980's, a number of Universities started to implement, and use Internet services for providing information to their own students and staff, which became known as Campus-Wide Information Services. Initially all of these services contained boring information (when lectures were scheduled, etc), but began to acquire more interesting information (such as MIT's guide to computers and health, or the notorious MIT lock-picking guide).

As the information became more interesting, some of the CWIS's began to provide information to users across the Internet, and the software that was used to implement them was also made available, and in some cases began to acquire features to be used across the network.

Most of these early distributed information systems were based around simple text menus, and simple text files — very few supported the use of graphics, or could be expanded to support graphics. They were mostly incompatible with each other, either requiring multiple "client" software packages, or multiple telnet sessions to obtain the information in each one.

Gopher

Gopher was one of the early distributed information systems that for various reasons took off, and became *very* popular for a while before the World-Wide Web became popular. Gopher clients look like a basic text menu which you can use to select further menus, or files to view.

Most of what is in "GopherSpace" are plain text files that are nowhere near as fancy as what can be found on the World-Wide Web.

Although Gopher is pretty much an obsolete service, many of the things that it innovated such as the standard for the presentation of multi-media material (using the MIME standard for pictures, sound, movies, etc.) have been absorbed by the World-Wide-Web. Even the first search engines (such as Veronica) were written to index GopherSpace.

Internet Relay Chat

IRC is a world-wide real-time chat service subdivided into channels which should be orientated towards a particular subject. Anything typed by anyone "tuned" to a channel, appears on the screen of everyone else when they press the *Return* key.

Although in my experience, most of what goes on in IRC channels tends to be a bit frivolous or even juvenile, it has been used for more serious purposes. For instance, private channels (only open to those with an invitation) are frequently used by geographically dispersed development teams to co-ordinate projects, and IRC was widely used when an earthquake in California knocked out the phone system to inform relatives that people were ok.

When you start an IRC client, it makes a connection to an IRC server (which is hopefully quite close). The IRC servers all talk to each other so that they appear to an IRC client to be one server. In the earliest days of IRC, all of the IRC servers were connected together, but recently some have split up to form separate IRC networks. These include EFFnet (the original), and Undernet.

Although the press tends to give the impression that most of the illegal or tacky things that go on with the help of the Internet are to do with the World-Wide-Web, in fact much of it takes place with the help of IRC.

Instant Messaging Services

Distributed File Systems

When you save files to a floppy disk, or to a hard disk you are making use of a *filesystem*, which determines how your files are stored onto the disk, where they are stored on the disk, and how they are named. When you are using a Microsoft operating system, you usually have access to only two different filesystems — the one for accessing your hard disk, and the one for accessing your CD-ROM drive.

When you use a networked computer, you also have access to drive letters that correspond to a distributed file system, where all of the files are stored on a file server, and many connected computers can make use of the same file system.

Although most network file servers are only semi-compatible with the Internet, a few filesystems have been written with Internet compatibility in mind.

NFS

Or Network File System.

NFS was first developed by Sun Microsystems to be used with their Unix workstations, but as they made the specification of how it works easily available, it spread quickly to other Unix systems. It has now become so wide-spread on Unix systems, that you are unlikely to encounter a Unix system which is not able to make use of NFS[27].

Although NFS is designed to be used over local fast connections, it can be (and is) used across the Internet. Some Internet archives allow others to NFS-mount their archives across the Internet (such as *Sunsite Northern Europe,* which is a *large* anonymous FTP archive with nearly 300Gbytes of files available).

The Andrew Filesystem

Or AFS for short.

AFS was developed by MIT at much the same time as they developed X-Windows specifically to make it easy to copy files across the Internet. Although not as wide-spread as NFS (NFS comes "free" with many Unix system, and you have to pay extra for AFS), many large American Universities do use it, and it is quite possible that you may find it mentioned.

"Streaming" Audio

Audio (i.e. music, etc) tends to make very large files when stored on a computer — for instance an audio CD-ROM can be copied onto a computer hard disk, but will take up about 650Mb. If you want to use the Web to distribute music, you *can* use simple audio files but these will take a long time to download (and start playing).

Compressing audio files does make things a bit easier — the files are smaller, and so quicker to start playing, but it still takes some time to start.

If an audio player can start playing the audio file when just a small part of the file has been downloaded, it begins to take on some of the aspects of the more common

[27] Although sometimes you will have to hand over a large chunk of the green folding stuff to get NFS — some Unix vendors like to charge extra for functionality that should come with the base operating system.

streaming audio players. These automatically adjust to the speed of the Internet, and will "skip" parts of the audio file if they start to get behind.

With streaming audio players, it is possible (and has been done) to create "Internet Radio Stations" where you click on a Web page link, and a few seconds later music (or voice) starts playing.

Internet Telephony

With a small enhancement, streaming audio players can "stream" in both directions so that you can speak into a microphone to send sound to someone else. Sound familiar ? Of course it does — just like a telephone, although it isn't quite the same.

However if a company equips its staff with multi-media PC's, and the necessary software, they can use the Internet to make phone calls amongst themselves. In a very large international company this can save very significant piles of cash. Unfortunately, companies doing this can slow down the Internet to the point where streaming audio just doesn't stream.

"Streaming" Video

In the same way that audio can be streamed, so can video (i.e. moving pictures with sound), although the streaming video takes a much larger amount of space and Internet bandwidth. With a very fast local area network, and very fast machines video can be quite large and impressive. More normally however, you get a tiny little window presenting a jerky and grainy film.

Videoconferencing

In the same way that streaming audio can be turned into a two-way system allowing communication, so can streaming video. Often with more than two people in a session, this is called video conferencing.

The same limitations that apply to streaming video also apply to video conferencing.

Peer-to-Peer Networking

The World-Wide-Web

Since the Mosaic browser first appeared, the World-Wide-Web (or WWW for short) has exhibited enormous growth. It is become so large that many novices tend to assume that the Internet and the Web are one and the same thing. Hopefully by now, you should be aware that this most definitely is not the case.

However this does not mean that the Web is not the most important user-level Internet service, with an absolutely mind-boggling amount of information available to those who can use the Web.

The first Web browsers and servers were developed at CERN in March 1989 by Tim Berners-Lee to aid in distributing information to the Physicists there, and originally used machines running the NextStep operating system (which is closely related to Unix). The Mosaic browser (written at NCSA), and NCSA's Web server were not available until a few years later, and it was these that started the Web.

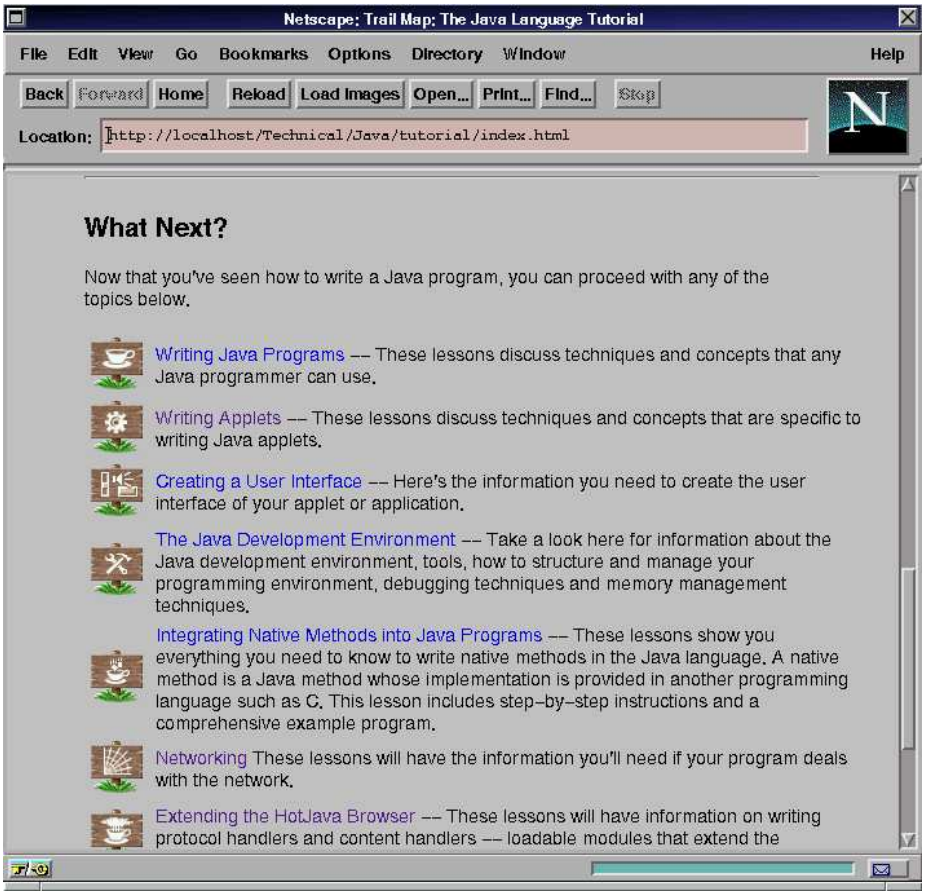
There are two halves to the Web — one half called the browser which is what you use to view what is on the Web, and the other called the server which is used to hold the Web (actually a tiny part of it).

The Browsers

The browser provides a more or less user-friendly interface to the Web to the user, and is usually capable of displaying a wide range of different kinds of information to the user. Probably the best known Web browser is Netscape, but probably more important is Mosaic[28]. Both of these are graphical Web browsers, but there are Web browsers for those who do not have access to a modern computer, or have no wish or need to see the pretty pictures — the blind have as much right and ability to use the Internet as sighted people.

[28]Due to the historical impact of Mosaic. Before Mosaic, the World-Wide-Web was just another distributed information system, and Mosaic gave the user a friendly tool for accessing the Web. Incidentally, Netscape is a descendant of Mosaic. The original programmers who wrote Mosaic, left NCSA to form a company to exploit their software. The original Mosaic (and updated versions) is still available from NCSA.

The screenshot above is what Lynx looks like when it is running, and the screenshot below is Netscape showing a page.



Although the browser is an essential item in the Web, it also stands alone as it can display information from other information sources — *Gopher* servers, *FTP* servers, or even files on your local hard disk.

The latest, most up-to-date browsers (especially Netscape) have become much more than just Web browsers — they allow you to read your email, read Netnews, . . . in fact they have become generic Internet access applications, although this does mean that they need quite powerful machines to work well.

HTML

HTML "expands" to "Hypertext Markup Language", and is the most common format of data served by Web servers. When displayed with a graphical browser such as Netscape, you will see text information, pictures, and icons. When displayed with a text browser (or read by computer to a blind person), you will of course lose the pictures and icons, but the text will still show up — at least if the Web page is properly written.

It is based on "SGML", which explains some of the peculiarities of the format — such as the lack of font controls, and layout which tends to infuriate those who are more used to conventional publishing. The earlier versions of HTML were particularly bad for this, but the latest versions include such things as tables, some font control, etc. . .

Although there is a standard for what HTML is (and what it can do), the two main competing Web browsers — Netscape's Communicator, and Microsoft's Internet Explorer each have their own enhancements to HTML. So there may be Web pages that are only properly visible with one or other of these browsers. Not many however, as most sensible Web authors tend to stick to the standard.

In addition to elements to specify the formatting of a document, HTML also has some features which allow forms to be created. This allows someone to write a Web document which with the help of readers allows him or her to collect some data.

URL's

Every Web page that you can reach has a unique URL, or Uniform Resource Locator. Basically a page's URL is it's address — telling a browser how to retrieve a file, and from where a file is located. An example of a URL is *http://www.iso.port.ac.uk/~mike/index.html* (which incidentally is my own unimpressive home page).

The first part of the URL (the bit up to the ":") is a specifier of how to fetch a page. The more common access methods are listed below :-

Prefix	Description
http	The "Hypertext Transfer Protocol". The most common method.
gopher	Instructs the browser to fetch a document from a Gopher server.
ftp	Use the file transfer protocol to collect a document.
mailto	Used to tell a browser to start sending an email message.
news	To pick something up from a Netnews server.

The second part of the URL (the bit between the double "/" and the first single "/" — in the example above, it is *www.hs.port.ac.uk*) is the hostname of the Web (or some other) server to talk to. The "address" in other words.

The final part of the URL (from the first single "/" until the end of the URL) is the part that indicates which document you want from the relevant Web server. This looks very much like a Unix filename, and frequently is (the pathname part of the example URL is *~mike/index.html*, which indicates a file (*index.html*) at the top of a person's Web documents collection.

Link Collections

Link collections are exactly what they sound like — Web sites whose purpose is to contain well ordered collections of links to other Web sites. There are many different kind of link collections; in fact almost every personal home page seems to have a very limited collection of links (which to be honest aren't usually up to much). Professional collections such as *Yahoo* (*http://www.yahoo.com/*) are very useful for browsing the Internet.

Search Engines

When you start to use the Web seriously for finding information that you need for some reason or another, you need to be able to locate it quickly. To hunt randomly around the Internet until you find what you need would take days or weeks. You need for something to search out what you are looking for. You need a search engine.

In the early days of the Web, many search engines sprung up to help in this task. They were normally run by individuals, or organisations interested in the methods of searching very large distributed databases, and some of them are still around today — mostly having made a transformation from an academic project into a commercial organisation.

Today, most search engines are run by commercial organisations who try and make money with advertisements on their Web pages. A few charge for their searches.

When you go to the starting page of a search engine such as *AltaVista* (*http://altavista.digital.com*), you will normally find an empty space to fill in a word or phrase that you are searching

for, and a button to instruct the search engine to start its work. The word or phrase that you use should be as specific as possible to avoid encountering many Web pages containing information that you are not interested in.

You can usually use much more complex searches than just hunting for a specific word or phrase. Just look for a "help button" on the search page for more information.

Search engines work by having two parts — a "robot" that crawls around the Web collecting Web pages to summarise and URL's to look at for other Web pages, and the search engine itself that searches through the information that the robot finds.

Due to the size of the Web, and the server-friendly behaviour of robots, it takes many weeks of continuous "travelling" for a robot to build up a database of the contents of all the Web pages. This database tends to be *extremely* large.

Because of the size of the database gathered by the robot, when you search for information using search engine, you will be using an enormous amount of computer power at the other end. If you do use *AltaVista*, you will be using a cluster of Digital[29] AlphaServers that has pushed the forefront of database server clusters.

Java and JavaScript

Java and JavaScript (which was previously called LiveScript) are both programming languages that a suitable browser can run. They both allow the Web authors to enhance their Web pages in an infinite numbers of ways.

JavaScript is the simpler of the two, and is capable of less than Java, but does allow significant enhancements to a Web page. Although in the end, a JavaScript enhanced Web page is still a Web page.

Java itself is *much* more powerful, which allows Web pages to be enhanced to a much greater degree, although it is quite a bit more complex to use than JavaScript. Java was originally intended to be a stand-alone programming language designed to be capable of implementing applications — for instance Sun's HotJava browser is itself written in Java, and although it tends to be slower than Netscape is pretty much just as capable (at least at presenting Web pages).

ActiveX

ActiveX is Microsoft's answer to Java, and is intended to allow software components written using standard software tools to enhance Web pages. These software compon-

[29] *AltaVista* is run by Digital, as a demonstration of power of their machines, a public service, and a revenue source (through advertising, although I doubt it brings in enough money to pay for the computer equipment). Digital has been purchased by Compaq, or D-Compaq as some of Digital's engineers sometimes accidentally call it.

ents could be downloaded from the Web automatically, or could be any software component residing on the hard disk of the users computer.

Unfortunately ActiveX controls suffer from two major disadvantages :-

- Components will only be written to work with Microsoft operating systems running on Intel chips. This is currently the majority of user computers, but goes against the idea that a browser should be available on every platform, and every Web page should be visible in every browser.

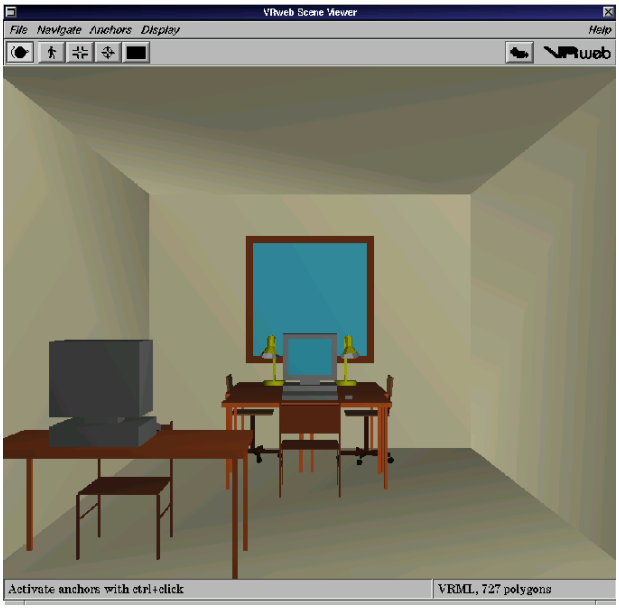
- More seriously, ActiveX controls can do anything they wish to the computer they run on — including erasing everything on your hard disk, or passing computer data from your computer onto an unknown third party. Microsoft's answer to this is to say that ActiveX controls should only be installed from "trusted" suppliers, and the decision as to whom should be trusted is left to the user[30].

The first problem is comparatively minor, but the second basically means that any browser supporting ActiveX components is a *totally* unacceptable risk in any network environment that has, or may have confidential information.

VRML

Which is short for "Virtual Reality Modelling Language". This is a standard for the representation of 3-dimensional "spaces", complete with moving objects, and Web links that a browser navigates in a visual sense. More than just a way of putting 3-dimensional pictures on your screen, it allows you to "fly" through a virtual landscape. Although VRML could be used as a general standard for "virtual reality", or for 3d modelling, it is designed to be integrated into Web browsers.

[30]Many users are likely to ignore any warnings, and install any ActiveX component. After all, users are not expected to be security experts.



Modern PC's are quite capable of displaying VRML scenes, although you may find that it takes a very high end PC (or a Unix workstation) to be able to display a scene with full textured objects (most VRML browsers/plugins allow you to change the quality of the display, and a scene with textures is the highest quality) and still let you fly around the display.

The Servers

A browser is one half of the story of the Web, but it does something at the other end to send it information — the server.

Servers essentially do little more than listen at an Internet port for requests from browsers, and return the information that they request[31]. As a comparison, a browser such as Netscape needs a *fast* 486 machine, or a Pentium to run well, whilst a server can run quite well on an old 386 machine.

Obviously, a server that is expected to serve requests from many browsers simultaneously will need to be a great deal faster than an old 386, and in extreme cases, a "server" will in fact be a large farm of high-end Unix machines made to look like one Web server.

Despite the best efforts of Microsoft, most of the Web servers on the Web are running on machines running the Unix operating system, and more alarmingly from the point of view for those people who want to sell us tools to run the Web, the most popular piece

[31]There is a little more to it than that, such as maintaining security, etc.

of software that implements a Web server (according to NetCraft) is Apache. Apache is a piece of software written by a large group of volunteers, and is free to use.

You may hear of the term *virtual servers*. Originally, Web server software expected to sit on one machine and provide one Web server on that machine. To run more than one Web server on one machine, it was necessary to use a non-standard port number for the second server (which is why you can sometimes come across URL's such as *http://somewhere.com:8080/*). Because the business of providing Web servers for others needed the facility, Web server software server software acquired the ability to run several servers with different names on the same machine.

"Secure" Servers

When your browser and a server talk to each other, it is normally easy[32] for someone to "spy" on the conversation to see what is going on. Not exactly what you want to happen when you are letting someone know your credit card details for a purchase on the Net.

Some servers (and most browsers) support a similar standard for transferring data to standard servers, except that the data travelling between the server and the browser is encrypted making it much more difficult[33] to collect your credit card details (or any other sensitive data). When you are talking to a secure server, the URL prefix changes from *http* to *https*. If you are in Europe, you are limited to 56-bit encryption which is considerably weaker than the US versions of the same software which uses 128-bit encryption — basically, 128-bits allows the use of longer numbers to encrypt the data making it much harder to decode the messages by brute force.

If you are buying products over the Internet you should ensure that you are talking to a secure server.

Web Caches

When Web browsers were first produced, they couldn't be used behind firewalls. This omission was quickly corrected so that people at large institutions that used firewalls could use the Web. A browser that is configured to work behind a firewall, is configured to send all the requests to servers on the Web through a proxy gateway.

At some point, somebody realised that much of the data going through a proxy gateway was the same — for instance, in a large organisation it is likely that someone has looked

[32]Well, it's easy for someone who knows how to run *tcpdump*, which whilst is intended as a tool for diagnosing network problems also works quite well as a snooping tool.

[33]But not impossible. Especially if you are not resident in the US, as encrypted services in the states are much more secure than the ones used internationally — the US classes encryption technologies as armements, and restricts their export.

at Yahoo's[34] main page in the half an hour before you try. By setting aside some disk space, and making some modifications to the way that the proxy gateway worked, the first caching proxy gateway was up and running.

The idea quickly caught on, as it makes sense, does tend to speed up browsing the Web, and reduces the amount of traffic on an Internet link[35].

Many large organisations run their own Web cache servers, and other organisations run Web cache servers for use by Web caches. A hierarchy of Web cache servers allows more documents to be cached, and a larger population to populate the cache — one of the problems with a Web cache is that if you are the only one who looks at a certain type of Web page, it doesn't improve things that much for you at first.

After having read some technically inaccurate, and wildly misleading articles[36] on the Web, I have decided to add something to this section to make it plain that these articles are totally wrong.

Firstly, caches are not intended to censor information. In fact the Web caching software that I use does not mention any possibility for this (although I dare say it is possible), and although I work in an academic environment where censorship is likely to be looked upon with disapproval, I suspect that most cache administrators would rather not bother. Having spoken to a cache administrator who does have to bother (he runs the Web cache for a school), I'm convinced that it is a major pain.

Secondly, caches do not provide "stale" information — they specifically do not cache dynamic information (such as the output of cgi scripts), or information that is marked as changing frequently.

Finally whilst cache software packages usually do provide some form of logging, it can't be used as an accurate track of what people are looking at as it just details what machine looked at what Web page. Which isn't enough to say that a particular person looked at a particular Web page. Besides which, some Web cache software packages have an option to mask out the machine name.

CGI

CGI stands for "Common Gateway Interface", and is a way for attaching Unix programs to the Web (according to rumour it also works for Microsoft programs, although why anyone would want to is beyond me) for enhancing the functionality of Web pages (such as putting counters on pages, etc), or for linking Web pages to some other kind

[34] A large, and well organised link collection. Have a look at *http://www.yahoo.com/*.

[35] Although this reduction in traffic is more than made up with by the increase in use of the Web.

[36] One article that appeared in the Independant newspaper was so over the top, and written by someone who both should have known better, and may have had a hidden agenda, that it grossly insulted the networking staff at UK Universities.

of information (such as providing a Web interface for looking at the contents of some database).

Whenever you fill in a Web form and "submit" the information back to the Web server, it is almost certain that you are sending the information back to some sort of CGI script. Even when all that CGI script does is to email someone the details of the form you filled in.

The Future & Other Bits

One way to look at what will happen in the future on the Internet is to take a look at what capabilities are available on high-end Unix workstations. A Unix workstation costs from £5,000 upwards. Some of the more extreme examples exceed £500,000. This expense buys you far superior performance to what a normal desktop PC is capable of *now*, although in 5 years time the standard PC could well match the workstation of today.

Many such workstations are dedicated to extreme applications which require such power (such as data-mining, scientific visualisation, molecular modelling, etc...), but a few are used by researchers to experiment with building future Internet applications. As an example, the first graphical Web browsers were available on NextStep workstations, and SGI workstations have had built-in video conferencing facilities for years.

One thing is quite obvious about just what is going to happen to the Internet over the coming years. It is going to become bigger — with both people who will use the Internet, and with services for them to use. It also going to become more commercial as more and more people realise that they can make money on the Internet. This has already happened to some extent with a large number of Web sites being funded with advertising revenue.

Virtual Communities

Some people criticise the Internet as something that encourages people to talk to computers, and neglect communication with other people. This is of course complete rubbish, as nobody talks to their computer (except when they swear at it when it crashes). When using the Internet, they are communicating *through* the computer to other people elsewhere on the Internet. In a very real sense, the various means of communication on the Internet (mailing lists, NetNews groups, IRC, Web-based chat lines, etc) each form virtual communities of people with interests in common.

In the real world, it can be difficult to find people with interests in common with you — particularly if the interests are unusual, or you live in a remote location with few people around. On the Internet, it is *always* possible to find someone with interests in common with you.

Virtual Worlds

If you combine IRC, the Web, and VRML into a service that offers high-speed virtual-reality in which you can meet other people[37], and interact you will have a "virtual world" service. You could also use a standard for such services to talk to each other — so that you could move from one virtual world to the next (maybe by "walking", or "taking a bus").

When connecting to a virtual world, you create an "avatar", or on-line version of yourself. Your avatar may not be anything like yourself (either in appearance, or behaviour), and is what other residents of the virtual world will see you as. Virtual worlds could be like the real world to some extent with buildings, and "normal" scenery, or they could be totally imaginary environments with no resemblance to reality at all.

Today most people will see VRML objects, and explore Virtual Worlds on a flat screen, which although can be quite exciting is hardly true "virtual reality". A new phrase has sprung up to mean what virtual reality used to mean — immersive virtual reality, where the user can't see anything that the computer doesn't generate. This can of course be extended to mean the feedback from all human senses are generated by computer — sight, sound, smell, and touch. Some equipment to do *some* of this is available today, although most of it tends to be very expensive (we're talking hundreds of thousands of pounds) and is normally only used when it is essential — such as flight simulators.

Intelligent Agents

When you start exploring the Web, you will realise just how vast it is, and most of it is of no interest to you at all. Finding what you want on the Web is very difficult — even with the help of search engines, and link collections. It would be nice if someone or something could do all the donkey work for you, and just let you look at the pages that you really *need* to see. This will probably take some of the fun out of exploring the Web, but those of us who use it more for serious use will find such things very useful.

Intelligent Agents are the name given to a new kind of software that is designed to hunt out the information that you want. They do this by browsing the Web whilst you are doing other things, and they build up a page of links to the pages that they think you are likely to be interested in. As time goes by, you train your agent (it spies on what you really look at) to become better at hunting out the pages you really want.

[37] Which may not be human. Although real artificial intelligence is still a long way off, some IRC robots have managed to be convincing enough that they have been mistaken for people.

Internet Access Devices

Currently if you want to be able to use the Internet from your home, you need a quite powerful desktop computer (approximately £1,000), a modem (approximately £100), and a subscription with an Internet provider (about £10 a month, plus the phone bills). That's a fair wedge for someone who might not be sure whether they want to make much use of the Internet, and that price won't buy you something that is particularly good at doing the more advanced things that you can do with the Internet.

A new kind of computer which is designed to be mostly a machine for accessing the Internet will be available in the near future. Designed to be more like a video recorder, or a games console; it will attach to your TV, and be much easier to use than normal computers.

As it is likely to be sold as an "Internet access device" rather than a computer, it is likely to attract the kind of people who are not interested in computers, but may be interested in the Internet. After all such people have almost certainly heard about the Internet, and some of them will be curious about just what it is. With more people on the Internet (and people who are less interested in the technology itself), the flavour of the Internet may change so that you are not expected to know about the technology as much. Hopefully it won't change the Internet community's fondness for freedom and privacy.

In addition, the equipment needed to get onto the Internet is getting smaller. With a small palmtop, and a cell phone modem it is possible to get connected with just two handheld boxes. Within a year of me writing this, it will be possible to get connected with just a small palmtop smaller than a typical paperback.

Microsoft's Plans

Microsoft wants to control the Internet. It wants you to use Microsoft software to browse the Internet, and it wants service providers to supply the services using Microsoft operating systems, and Microsoft server software. And it badly wants to become a service provider itself. Does it want to do this out of altruism ? No of course not. It wants to make even more money.

Not in itself a bad thing, but it wants to do more. It wants to restrict the ability of competitors to compete in the same way that it has managed in the desktop operating systems market, and the applications market. It wants to be the only choice.

In a memo leaked from Microsoft, one of the ways that it plans to do this is to produce Internet software that it is very difficult for competitors to communicate with. In the past (and the present), all the important Internet services have been documented in open standards documents that are available to all (the RFC's). Microsoft plans to introduce

new standards that aren't published and are deliberately obfuscated to make it difficult for people to reverse engineer the protocol being used.

It has already done this to some extent with the protocols used for providing file and print networking with it's Windows operating systems, but thanks to a very clever bunch of engineers, it is possible to provide these services using the Samba (*http://www.samba.org/*) software package. In fact Samba works so well, that it far outperforms Microsoft's flagship operating system (Windows NT) at providing these services.

In addition, Microsoft has a package called FrontPage to help users write their own web pages[38]. This works well with Microsoft's web servers, and to be fair, they have released extensions so that FrontPage will work with other services. However, their extensions for other services are astonishingly badly written (or well written if you consider that they might have a hidden agenda) — the instructions are abysmal, the software doesn't work as advertised, and doesn't integrate with the servers in the correct way. It *is* possible to use these extensions with software like Apache, but it's such a difficult process that most will just give up (me included).

Will it succeed ? Hopefully not.

CyberWar and CyberTerrorism

I have mentioned Internet security in passing in this document, but not in any great depth. One of the things that *will* happen in the future is that organisations will use the Internet to attack their enemies — and not just companies. To some extent this is already happening — there is already some indication that there is quite a bit of industrial espionage going on over the Internet, and the American military employs "tiger teams" to attack their own systems to find security problems.

In a world where we are *all* (whether we use the Internet ourselves or not) becoming increasingly dependent on the Internet, it becomes more and more likely that a large amount of disruption in the real world could be caused by attacking certain parts of the Internet. For example a heavy, and sustained attack on British Telecomm's Internet servers could result in a high cost to that company — both in terms of the time it takes to repair any damage to systems that may have been caused, the time it takes to protect themselves, embarrassment caused by changing their Web servers, and the damage caused by the leaking of sensitive information. An unscrupulous competitor of BT would certainly like to see something like this happen, and if it thought that it could hire a group to do this without it being traced back to them might do so[39].

[38] I've known a number of people who *used* to use FrontPage, but moved onto other packages when they gained enough experience to learn just how limited FrontPage is.

[39] Let me make it plain that I don't believe that BT or any of its' competitors is indulging in such activities, or would do so in the future.

The military would certainly be interested in such activities — both to protect themselves, and to cause "confusion" to their enemies. The US military has acknowledged that it is developing both defence and attack techniques in this new kind of warfare. And of course, it is a great way for terrorists to work — it is far cheaper than conventional means.

Final Word

The Internet is a fascinating place, full of interesting information and with quite a few interesting people to communicate with, and will undoubtedly dramatically change the way that we work and play (for some of us, this is already true). It will certainly be far easier to contact people who have the same interests as ourselves — no matter how obscure those interests may be.

Take Internet shopping for example — it is possible today to buy products and services on the Internet[40], and have them delivered direct to your home. Although it may not be immediately obvious that this is better than mail order, it certainly is — after all, the Internet is truly international, and gives you a wider choice in finding a perfect bargain. It certainly makes sense for those of us who do not like shopping, or who find it difficult to get to the shops for whatever reason.

Hopefully, this document has given you a better understanding of what the Internet is.

[40] Although it would be advisible to find out whether you are using a secure method whilst buying.

Appendix A: Tracing Spammers

When looking at an email message sent by a spammer, you should realise that the email address that it appears to come from (that contained in the "From" header, and the more difficult to see envelope sender) is almost certainly forged. In fact spammers have been known to be sued by the Internet Service Providers that are behind the forged from addresses. To trace a spammer you have to look at the headers that indicate where the message was received (it will have been received at several locations, if not many).

Normally, when you are reading an email message, most of the mail headers (which are normally very boring, even to those who know what they are saying) are hidden from view. You will have to find the command that shows the headers properly, and you will see something like the following (actually the following is faked) :-

```
Received: from alpha9.pw.pling.co.uk (alpha9.pw.port.ac.uk) [10.97.24.14]) by
          hobbit.pw.pling.co.uk (8.9.3/8.9.3) with ESMTP id FAA21154 for
          <\mike@hobbit.pw.pling.co.uk>; Fri, 12 Feb 1999 06:11:53 GMT
Received: from inn.victim.net (s23.victim.net [10.21.23.9]) by alpha9.pw.pling.co.uk
          (8.7.1/8.7.1) with ESMTP id FAA22164 for <mike.meredith@pling.co.uk>;
          Fri, 12 Feb 1999 06:11:52  GMT
Received: from mail.isp.net (m3.isp.net [10.92.1.3]) by inn.victim.net
          (8.9.1/8.9.1) with ESMTP id TAA22690 for <mike.meredith@pling.co.uk>;
          Thu, 11 Feb 1999 10:53:11 -0800 (PST)
From:     51432@aol.com
Received: from mail.isp.net (slip87-65.nw.green.isp2.net [10.9.65.87]) by
          mail.isp.net (8.8.8/980514) with SMTP id EAA08827; Fri, 12 Feb 1999
          05:35:54 (GMT)
Message-ID: <199902120337.EAA08827@mail.isp.net>
To:       friend@everywhere.com
Date:     Thu, 11 Feb 1999 18:49:23 GMT
Subject:  MAKE MONEY !!!
```

First of all, ignore the "From" address — it's worthless (and this spammer could well get sued as AOL is into sueing spammers for such fake From addresses). Also ignore the "To" address as it is also worthless. Look at the "Received" headers, as they are added by the mail transport software, and are not under control of the spammer[41].

The first "Received" header (from the bottom up) indicates that the message was first received by a server called "mail.isp.net", from another machine called "slip87-65.nw.green.isp2.net".

[41] Actually the spammer can insert fake "Received" headers, but these will appear towards the bottom of the headers, and he or she can't alter what the earlier ones will say. And one will give the game away as to whose customer he is.

In this case, the server "mail.isp.net" is an innocent victim (like "inn.victim.net" in the "Received" header further up). The "slip87-65.nw.green.isp2.net" address is the address of the spammer's machine, and you should probably complain to the spammer's ISP to get his or her account cancelled.

An example of a suitable complaint follows. Note that the innocent victims are also sent an email message pointing out that they are being used as a relay site.

```
To: abuse@nw.green.isp2.net, abuse@green.isp2.net, abuse@isp2.net,
    postmaster@mail.isp.net, postmaster@isp.net,
    postmaster@inn.victim.net, postmaster@victim.net
Subject: SPAM Complaint

isp2.net:

One of your customers appears to be sending me UBE which I do not
wish to be receiving. Could you prevent him or her from doing so
again. In case it is not your customer that is spamming me, could
you please forward my complaint to the spammer's ISP, and prevent
your system from relaying such mail.

isp.net, victim.net:

It appears that a spammer is using your facilities to bounce spam
onto me (and probably many others). Could you consider securing your
server so that it cannot be used in this way ?

Spam message follows:-
<Include the full spam email *including* all headers>
```

The addresses on the first line of the "To" field are obtained from the spammer's machine name — just send email to "abuse" at every possible email address making up the spammer's machine name. The other lines of the "To" field are made up in a similar manner, except that the "postmaster" address is used instead — whilst ISP's do have an "abuse" address, some sites may only have a "postmaster" address.

Note that you should remain polite, as ranting just gives a bad impression and it is possible that you have mis-identified the spammer's ISP.

Index