Freely Redistributable Software across the Internet - Current practice and future directions to overcome the bandwidth crisis.

Neil Smith
HENSA Unix,
University of Kent at Canterbury.

Abstract

The Internet offers what must be the best available medium to track, contribute to and distribute free software. Unfortunately, as soon as national boundaries are crossed, the Internet has never really been capable of transmitting such large bodies of software within a reasonable amount of time. This has led to the installation of large dedicated archives whose sole purpose is to build up repositories of this software wherever there is the demand. Initially such archives of software may have been kept up-to-date by manually sending magnetic tapes back and forth. This process was later automated as international network links were established. It is at this point that most archives in operation today have stopped. The bandwidth crisis that is currently making large parts of the Internet virtually useless is just what is needed to spur on the next generation of technologies to make electronic information and software dissemination more effective.

In this paper we take a very brief look at the history of archive sites. We then present in more detail the differences and evolution in traffic flow that we have observed from our FTP archive and through our National World-Wide Web caching service. This is followed by a discussion of a proposed European wide scheme to make FTP archives more effective by co-operative effort and finishes with a preview of software currently under development at HENSA Unix which addresses the problem through a new, more flexible and scalable approach based on a combination of caching and mirroring.

Introduction

For the past six years, HENSA[1] has been operating a large Unix oriented FTP archive for the benefit of the higher education community in the United Kingdom. Since November 1993 we have also been offering a public World-Wide Web Proxy Cache which has recently been adopted as the UK's National Caching Service. Through our involvement with these two services we have been able to see how the use of the network and the characteristics of the traffic have changed. The HENSA Unix FTP archive has always been popular, but we have seen use of the WWW cache simply explode. This is a reflection of the extent to which the Web has become the dominant force on the Internet.

With the United Kingdom's relatively poor international bandwidth[2] the archive's daily FTP mirrors have suffered greatly and we now find that we have to expend a great deal of effort on techniques to make the archive effective again. (Any archive that is unable to maintain an accurate copy of the remote site quickly gains a poor reputation and becomes worse than useless. Users that make use of the archive may not see the latest state of the software, and those that realise there is a problem merely make it worse by using up the resource required while fetching the software they want directly). The fact that FTP is likely to remain the protocol of choice in a number of specific file transmission scenarios means that expending this effort is worthwhile.

The particular problems associated with information distribution on the World-Wide Web have taught us a great deal and it is interesting to consider whether some of the lessons learned on the Web can be applied to FTP archives. In any case users of FTP archives have to co-exist and share bandwdith with users of the Web and so the next generation of software distribution technology cannot be developed in isolation.

History of the Archive

Following attendance at a workshop on scientific software in 1987, Tim Hopkins, collaborated with Eric Grosse, the co-founder of the Netlib software distribution service, to install a Netlib server and associated software archive at the University of Kent.

[1] HENSA Unix, based at the University of Kent at Canterbury, is one half of a JISC, (the Joint Information Systems Committee), funded project supplying freely redistributable software to the higher education community in the United Kingdom.

[2] The UK higher education community currently benefits from a 150Mbps national backbone, but only has 4Mbps to the United States, 4Mbps into Europe and 2Mbps to Scandanavia.

This server provided numerical software, mainly written in Fortran, free of charge to anyone who could access the University via e-mail, thus reducing the need for expensive accesses to the US to obtain the same material. The main difficulty encountered at this time was keeping the archive in step with the ever changing US master archive. The state of the art at the time was to exchange tapes or to send e-mail requests for specific files; the archive was consequently never fully in step.

Following the arrival of an Internet link at UKC in 1992 a mirror of the Uunet archive at ftp.uu.net was introduced. This was the first major enhancement to the type of material the service provided; the emphasis was now on Unix based material though uunet also provided a wealth of general interest information. The availability of the uunet archive changed the perception of the service and opened it up to users from a wide variety of disciplines and interests.

Since that time the material on the archive has been supplemented with mirrors of a large number of remote sites carrying freely redistributable software.[3] The archive also carries a number of home grown packages and sub-archives[4], and is always keen to receive suggestions of other software that would be of use to the community.

We have made little attempt to build a World-Wide Web based interface around the material in the archive. Maintaining a large archive is difficult. When the majority of the material contained in that archive has the potential to change on a daily basis, building a useful Web infrastructure that requires little maintenance is virtually impossible. Instead the archive has chosen to simply provide the mirrored material "as is" and any requests to the archive via the HTTP protocol are simply the result of people following links to the top level of the mirrors and then navigating from there as if it were FTP.

The sense in which the archive has been influenced by the World-Wide Web is in terms of its public caching proxy. Taken over a few weeks this service now accounts for a significantly larger number of accesses and volume of data transferred than the FTP archive throughout its entire six year history.

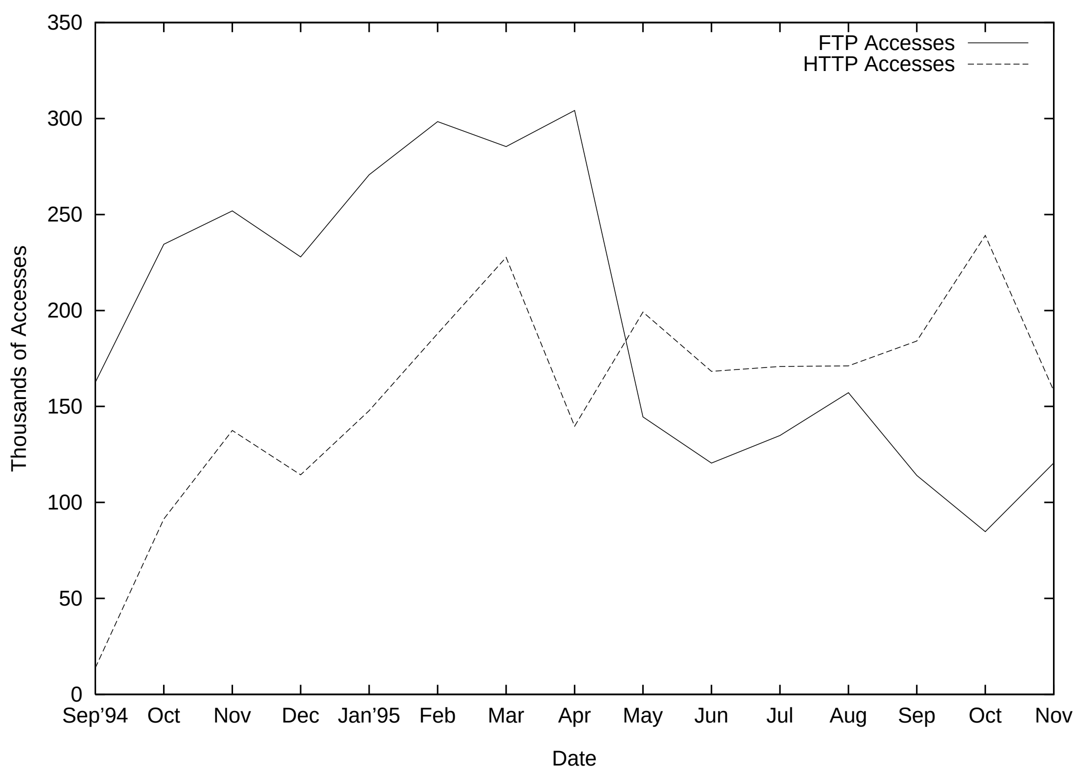

Figure 1: Accesses to the HENSA Unix Archive by protocol

[3] For example Linux, FreeBSD, Statlib, Xv, Xanim, Glimpse, Chimera, Perl CPAN, Hyper G, Lynx, Netscape, Java, and Apache.
[4] For example the ups debugger, the Dyslexia Archive, the Parallel Computing Archive and the Combined Log File Handling System.

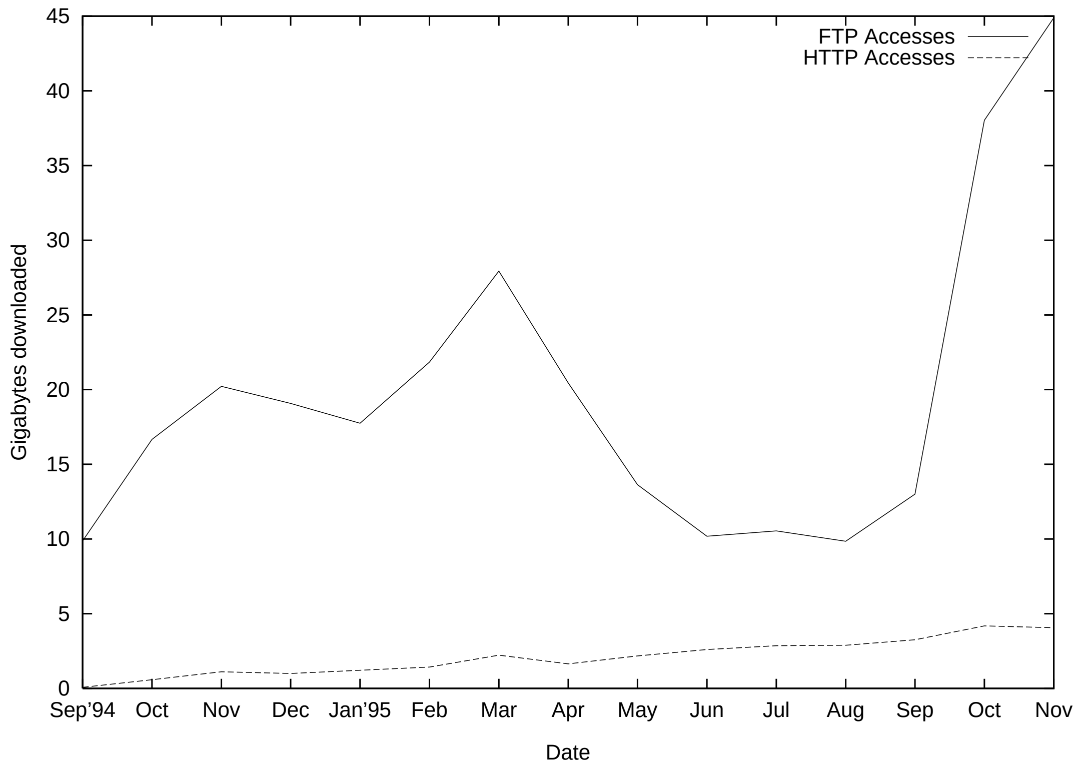

Figure 2: Volume of data downloaded from the HENSA Unix Archive by protocol

Traffic Flow Analysis

The way in which the Internet is used is by no means static. Over the past couple of years the distribution of protocols used has changed dramatically. Not long ago electronic mail and FTP transfers accounted for the majority of Internet traffic, Gopher started to catch on, but before it could attain any real grip the Web arrived. Now it is Web traffic that accounts for the majority of all traffic on the Internet.

However, while figure 1 suggests that the use of FTP is diminishing, figure 2 demonstrates that the volume of traffic handled by FTP for software retrieval remains high. Over the last year the number of FTP accesses to the archive has varied in the usual seasonal manner, but with the new academic year it has not picked up as expected. The conclusion that can be drawn from this evidence is that FTP continues to be used for the transfer of large independent bodies of information while the Web is now the medium of choice for other material.

The volume of data retrieved from the archive by FTP has followed the normal patterns. We believe that part of the particularly large increase in the volume since September 1995 can be attributed to several recent software releases; FreeBSD 2.1.0[5] and Netscape version 2 being two examples. Of course, other factors, such as more users coming online with the new academic year, will also have had some effect. Retrieval of material from the archive by HTTP can be seen to be growing steadily but this is almost certainly due to the general growth of the Internet population rather than a move towards using HTTP for software retrievals.

This evidence shows a small movement away from FTP, but for archive access it is fair to say that FTP is still the protocol of choice. By itself this is not a problem. The real problem can be seen by examining figure 3 and the growth of the HENSA proxy. The growth of this service is fairly representative of the growth of the use of the Internet as a whole. It is the competition for the shared bandwidth introduced by this explosive growth that is starting to make software distribution a significantly more difficult problem that it used to be.

[5] Installation is so simple I only had to go through the procedure three times :-)

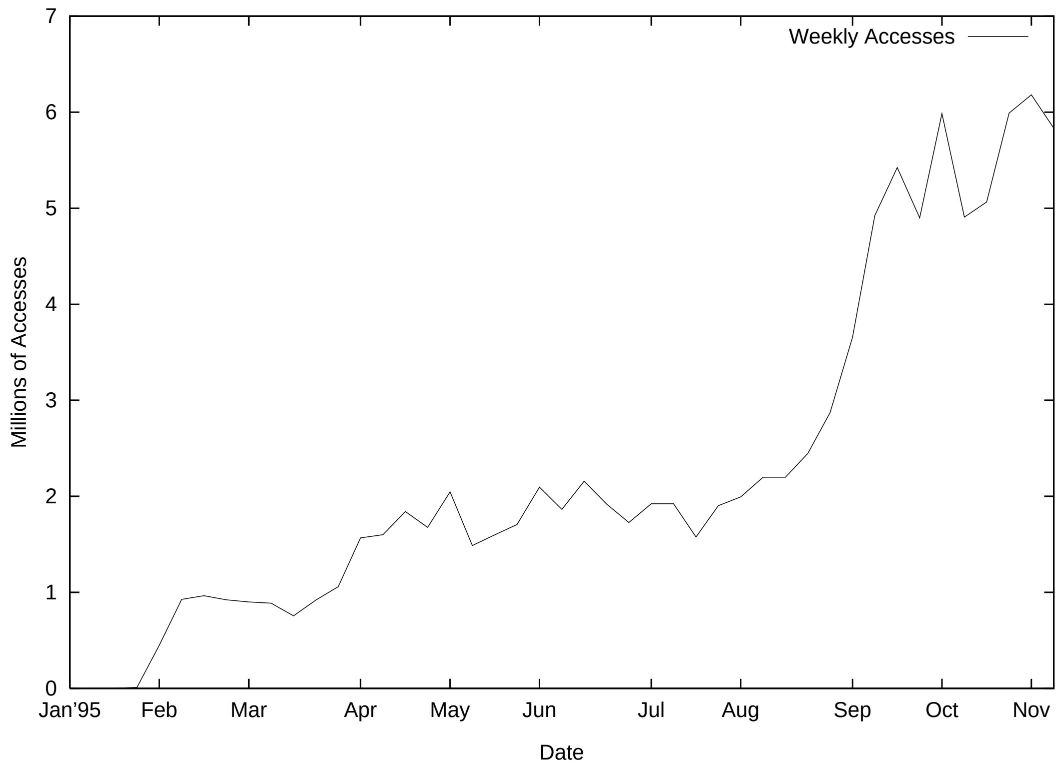

Figure 3: Accesses to the HENSA Unix World-Wide Web Proxy

Current mirroring strategies

The currently implemented strategy to overcome this bandwidth crisis is simple mirroring. This involves the retrieving site (the client) building up a list of the files that it currently has, and another list of the files available on the publishing site (the server). These lists are then compared and the client performs the necessary actions to make its filestore mimic that on the server.

This mirroring can occur at several levels. For example the Parallel Computing archive, based at HENSA Unix, is mirroring 79 packages from 59 different sites. This archive is collecting together resources from all over the world. In turn this collection is then mirrored in Australia, France and Japan to provide fast local access to users in those countries.

Unfortunately, assuming mirrors happen on a daily basis (it is not always the case that they are scheduled so frequently) this scheme can lead to software on the archives being up to 24 hours out of date for every hop through the chain of mirrors. Congested networks and very busy servers can reduce the effectiveness of mirroring dramatically and result in even longer periods between successful updates.

To counteract this some of the more popular sites operate special mirror accounts to ensure that the mirror process can always log on, (FreeBSD for example), and others announce the availability of software to mirrors in advance of it being available to the general public, (Netscape for example). These measures are not always enough as the lack of bandwidth often results in lost connections.

There comes a point where the period between the release of new software and its arrival on a local mirror is so long that users, who would otherwise make use of a local archive, determine that the material is too out of date and resort to retrieving it directly from the publishing site. This only compounds the problem.

Proposed Co-operative Scheme

One approach to solving the problem of out of date mirrors, and thereby persuading more people that their use is effective and reducing the overall waste of bandwdith, is to move away from the model of independent national archives. Instead we would like to propose a scheme where archives that are relatively well connected to each other, that is, there tends to be some bandwidth between

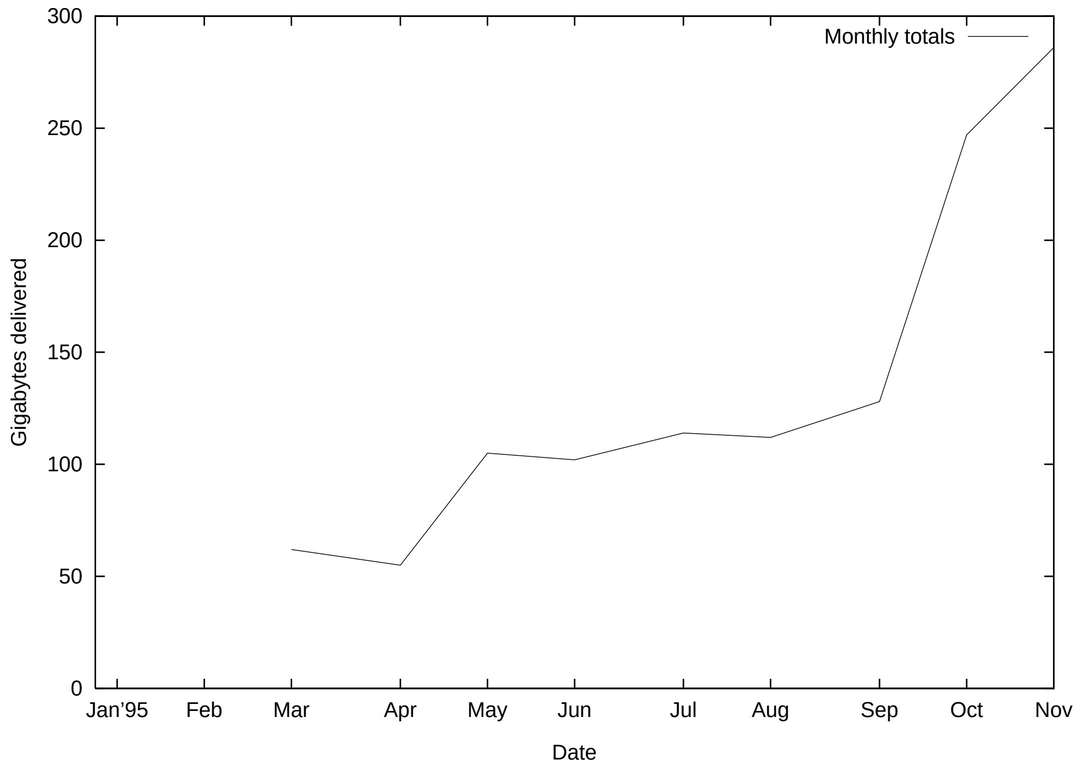

Figure 4: Volume of data delivered by the HENSA Unix World-Wide Web Proxy

them, co-operate to make the best use of their combined resources. Typically this will mean an archive in a region with a good connection to the remote site to be mirrored performing the first mirror operation. The other archives with poorer connections would then mirror this intermediate copy.

If we take the UK and Scandanavian countries as an example, we see that within the UK there are at least three large archives that mirror material from the United States[6]. When they perform their mirrors they are competing for a very congested 4Mbps worth of bandwidth. A significant proportion of the material that these archives hold is already available on archives in the Scandanavian countries. These countries have access to a relatively lightly loaded 34Mbps link to the US, with the 2Mbps link between the UK and these countries also being relatively lightly loaded. It would make sense for the archives in the UK to mirror their material from Scandanavia.

If this scheme were naively put into place we would suffer from the fact that our material is now an extra hop away from the originating source and is therefore that much more out of date. Rather than providing a better service, this may be considered by some users as worse. In order to ensure that the mirror in the UK is at least as up-to-date as it used to be it is necessary to put into a place a scheme whereby the clients of an intermediate mirror can perform their updates as soon as the intermediate server has finished. A simple signaling mechanism, for example an electronic mail gateway, or a status file somewhere in the anonymous FTP tree, could be used to communicate between servers and clients. Provided there is sufficient bandwidth between all parties a complete hierarchy of mirrors should still be able to complete a copy in one night.

If it is found that the mirrors do take more than the available time to complete, the next logical extension would be to have the two mirroring servers co-operate at the file level rather than site level. Rather than the final client only starting once the intermediate client has finished, the two could co-ordinate their work so that the final client would mirror each individual file as soon as it has been completely retrieved by the intermediate client.

At the moment this scheme is merely a proposition. Interest in co-operative mirroring has been expressed by Luleå University in Sweden and HENSA Unix in the UK. This solution makes most sense in areas short of international bandwdith, and it is hoped that other participants in

[6] Admittedly there is some overlap amongst the material that they mirror and if it were possible to put rivalry and funding issues aside the result would be co-operation rather than competition.

Europe will express an interest.

Web Caching

In terms of bandwidth, the World-Wide Web is a real "Killer App". The combination of a very popular system that promises multimedia rich documents in an interactive environment with no automatic replication of information or distribution of load will bring any bandwidth bound communications system to a dead stop, or at least close enough to be indistinguishable from dead. This fact was clearly apparent in 1993 and led to some thought going into how things could be made better.

Having provided an FTP mirror service for a number of years our first thoughts were towards providing a similar service for the Web. At this point there were only a few sites that people were really interested in, but documents on the Web have some fundamental characteristics that make them much more difficult to mirror that software packages. This work is more thoroughly discussed in [1].

- In general, documents on the Web are not independent of their location. They contain links to other documents on the same server, or documents on completely different servers. Any document mirrored to a local server would have to have all the appropriate links rewritten. This means grubbing around in the HTML and making a decision for each link; a very expensive process.
- A significant proportion of the documents on the Web do not actually exist in any sense that can be easily replicated. Documents generated on the fly as a result of a CGI script execution could only be mirrored through the mirroring of that CGI script and its whole support environment. Doing this for a popular service, such as InfoSeek, is a technically daunting prospect, and would be, financially, very expensive.
- Web documents tend to be small and numerous. HTTP is an effective protocol to use when retrieving documents interactively. When fetching large numbers of documents automatically, the overheads involved become significant and result in a very inefficient mirroring process.

Fortunately, at about the same time, Lagoon, a prototypical Web cache became available and was quickly installed at HENSA Unix as an experimental service. It proved extremely popular, although suffered a little as a result of a lack of real proxy support in clients, and unrefined protocol implementations. The inclusion of proxy support in version 3 of the CERN server provided a much more stable platform and HENSA Unix started using this server in the first quarter of 1994. The load continued to ramp up and at the beginning of 1995 another switch was made from the relatively inefficient CERN server to the dedicated Netscape Proxy Server. This was accompanied by a move to dedicated hardware, initially in the form of a Silicon Graphics Challenge S server and then a dual processor Challenge DM. Over the summer of 1995, and based on the expertise built up over two years, HENSA Unix was charged with running the UK's National Web Cache. In order to serve this load an addition five Challenge S servers were acquired. These will be placed at different locations around the UK with a great deal of pressure being placed on institutions to make use of them.

FTP Caching

Caching proxies were introduced to the Web and developed to provide quality services very quickly. The nature of the Web makes their use very effective and they are seen as a key technology in the fight for bandwidth. The nature of FTP is different to the Web, but they do have to co-exist and it is for this reason that at HENSA Unix we are now investigating the feasibility of an FTP caching system.

FTP caching is supported by both the CERN Server and the Netscape Proxy Server but it is only in a very naive form. Their HTTP caching allows for the use of complex heuristic rules to establish a strategy for determining when documents become out of date. Such an opportunity does not exist for FTP. Instead any file retrieved by FTP is simply kept for a specific period of time before a fresh version is pulled. A dedicated FTP cache with extra knowledge of the FTP protocol and an understanding of the organisation of the resources available via FTP on the Internet could do a lot better.

The problems that make FTP mirrors less and less of an attractive solution are the general increase of information on the Internet and the wider and wider dispersion of that information. As more information becomes available from more publishers, maintaining a complete mirror of all that information becomes an enormous task. A single caching proxy that could fetch material from any publisher would require no human management. Users would not have to concern themselves over which mirror site to visit to find a particular piece of software, they would always use their local FTP cache.

A comprehensive FTP cache could make use of the same kinds of heuristic rules as used in HTTP caches in an attempt to make educated guesses about the frequency at which material is updated. The equivalent of an If-Modified-Since HTTP request could be made by connecting to the publishing site and looking at the modtime on the file in question. In addition, based on past experience, a profile of each site could be constructed giving hints as to the frequency at which the contents of particular parts of a remote archive change. A record of the popularity of each remote archive and each part of that archive could be kept with a view towards prefetching the most popular material once it has changed.

A USENET News aware cache would be able to scan appropriate newsgroups looking for announcements or mentions of software available by FTP. Often such announcements are followed by a flurry of activity and retrievals. Having the software prefetched during a period of relative network idleness would ensure that it is available to satisfy the following days demand.

As an FTP cache encounters more and more servers it should be able to determine whether any of these servers are in fact storing duplicate information. In this case one is likely to be a mirror of another or several are mirrors of a common third party. This information can then be used to save space in the cache (only one copy of duplicate files needs to be kept) and can also be used in the process of resolving a user's request that cannot be immediately satisfied from cached data. For example, if a user in the UK requested a file from the ftp.uu.net archive the cache may have built up the knowledge that this archive is mirrored at HENSA Unix and Imperial College in the United Kingdom and at various other sites around Europe. The cache would try to satisfy the request by making use of one of these sites in preference to the publishing site.

Originally the choice of the most favourable site to retrieve from would be made on the basis of the difference in domain names. However this could be superceded by the cache's experience of the responsiveness of different sites measured over several retrievals at different times during the day.

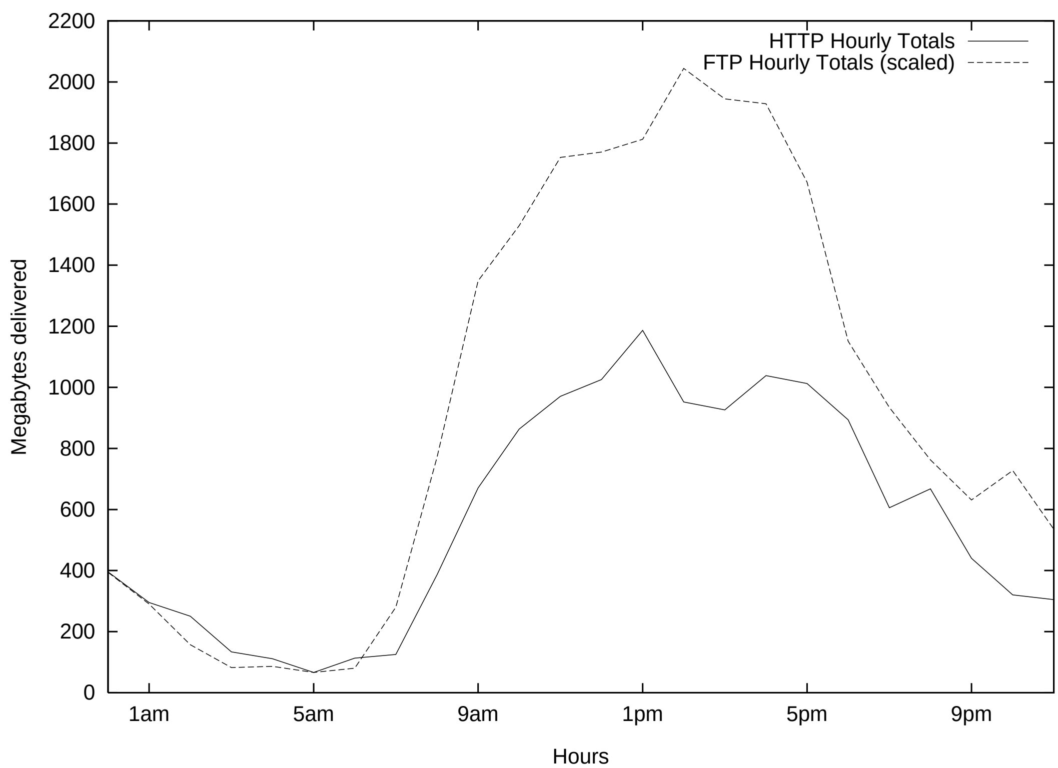

Figure 5: Pattern of data delivery to clients through the day comparing HTTP and FTP

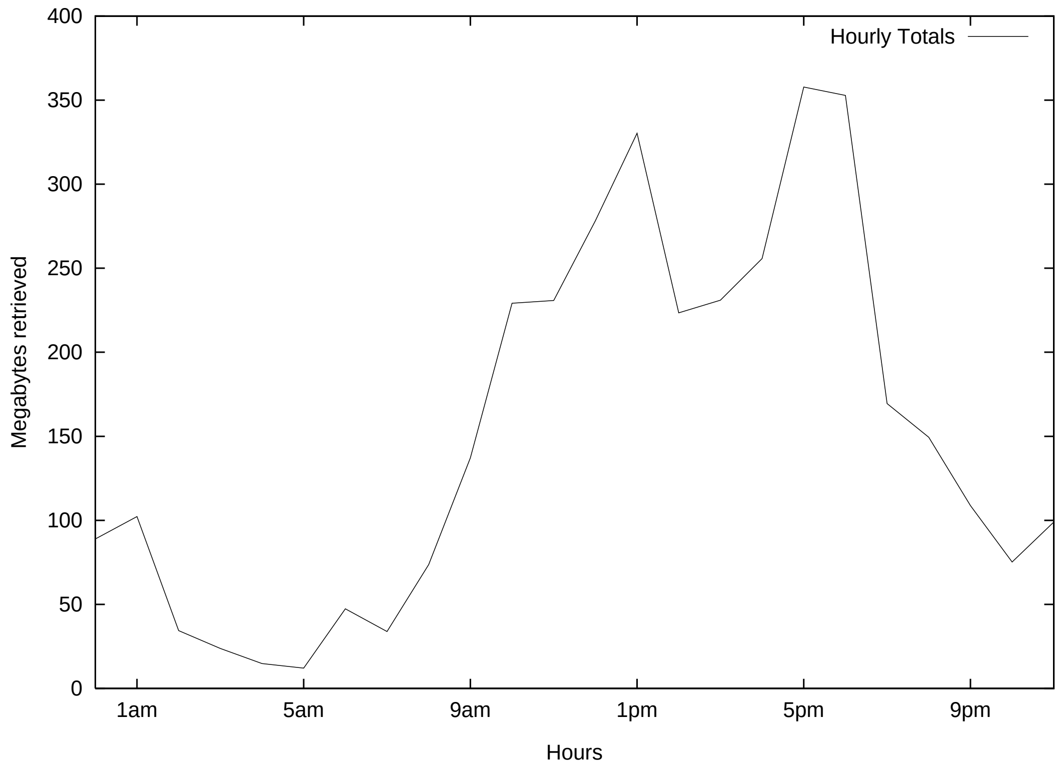

Figure 6: Pattern of data retrieval by the UK National Web Cache from remote servers through the day

Other Future Directions

The widespread use of Web caches and effective FTP caches would surely have a dramatic effect on the crisis that we are currently undergoing. Of course, the deployment of faster national and international network connections will also help to some extent, but with demands on resources continuing to grow exponentially we need to put into place smarter bandwdith saving schemes.

Figure 5 shows how the HENSA Unix FTP archive and the UK National Web Cache respond to the demands placed upon them over a 24 hour period. The volumes of data that the two services ship differ by at least an order of magnitude and so the plot for FTP traffic has been scaled to lie on top of the HTTP plot during the early hours of the morning. At this point it is hoped that the Web proxy will be unconstrained by the available bandwdith. Obviously the performance of the FTP archive is never constrained by bandwdith.

If we assume that the plot of demand on the FTP service is representative of general access patterns throughout the day then we can see that the Web cache is performing extremely poorly in comparison. Looking at figure 6 we can see the reasons for this.

The cache services about 25% of its requests without ever contacting a remote site. The other 75% of requests result in a connection to the remote site, even if it is only to determine whether a document is up-to-date or not. While performing these checks and retrieving the necessary files the cache is severely constrained by the bandwidth available. The capping effect of the limited bandwidth is clearly visible in figure 6, and at peak times, during the afternoon, the throughput actually drops away from optimal as congestion on the international network links wastes more and more bandwdith.

The heuristic rules in place on the cache mean that once checked most documents do not have to be checked again during that day, unless a user explicitly requests it. This suggests that if we could anticipate what was going to be requested during any particular day we could perform the fetches when there is free bandwidth, between 2am and 5am for example, and push the graph of Web cache responsiveness up closer to that of FTP responsiveness.

The challenge is to determine what should be prefetched. The scheme currently under investigation at HENSA Unix is the simplest possible. Prefetches are performed based the previous days demand. This means that any document fetched more than once on the previous day is scheduled to be automatically fetched early the following morning. It is currently too early to announce any results from this investigation.

An alternative strategy would be to have the cache build some knowledge of the most popular pages and search for new links appearing on those pages. It is likely that any pages linked to, from a popular page, will be requested a significant number of times.

Having some method whereby information publishers can automatically notify proxies of changes to files would take a lot of the guesswork out of prefetching. However, such a scheme would require a lot more support from the information publishers and probably extra support in the HTTP protocol allowing a publisher's server to contact a proxy, or network of proxies, and "suggest" that it retrieve a URL.

Conclusion

The World-Wide Web may be the fashionable protocol to use, and for interactive applications it certainly deserves its place. However there is evidence to suggest that FTP is still heavily used in a number of specific cases. If software distribution across international networks via FTP is to remain a viable proposition in a world where virtually all the bandwidth is used up through Web surfing new techniques will have to be used to ensure that the software gets to the places where it is wanted and needed.

References

[1] Neil G. Smith. *What can archives offer the World-Wide Web?* First International World-Wide Web Conference, CERN, Geneva, 1994 - Advance proceedings.

Biography

Neil G. Smith

Neil Smith graduated from the University of Kent in 1992. The following two years were spent trying to balance the demands of system administration for the University and the ballooning success of HENSA Unix. Since May 1995 he has worked exclusively for HENSA and has been developing and promoting the UK's National Web Caching Service.

Contact information can be obtained from the page at *http://www.hensa.ac.uk/team/N.G.Smith.html*