A Feasibility Study of Peer-to-Peer File Sharing in Enterprise Internetworking Environments

BSc Networking & Computer Support Year Four

Submitted for the degree of BSc in Networking & Computer Support, 2005-2006

Author: Captain Haddock
Supervisor: Jim Paterson

Except where explicitly stated all work in this document is my own.

Signed: ____________________ Date:________________

Abstract

Peer-to-peer file sharing has become a proven and reliable method of distributing Internet content. Peer-to-Peer Programs such as BitTorrent are now widely used by Internet downloaders, due to the speed advantage compared to downloading from http or ftp dedicated servers.

File-sharing networks on the Internet create a distributed repository of information that look like a single source to the user, yet are often made up of millions of interconnected home desktop computer machines. It has been shown that distributed computing system can achieve higher aggregate download speeds than client/server methods and use resources efficiently, yet the nearly all businesses still adopt the traditional client/server model. I believe that the advent of this technology will in time remove the need for the hierarchal structure in networks, eventually rendering the client/server model obsolete.

In this paper, I will attempt to ascertain whether peer-to-peer technology is feasible for use within an enterprise Internetworking environment. I will focus of the advantages and disadvantages that such a system could bring compared to conventional client/server method. By constructing a test network and performing a series of controlled experiments, measuring transfer time of BitTorrent versus client/server, I was able to determine if there are advantages in performance. Another aspect of this project was to determine the performance impact such a system could have on participating peers. Performance has been measured by comparing logged resource usage and using a static benchmark suite while a file-sharing was active and while it was not. Next, subjects performed a series of tasks on PCs- firstly while BitTorrent was active and a second time while it was not. Subjects filled in performance logs while undertaking a series of day-to-day task on a computer, and were unaware if BitTorrent was on or off.

The results gathered by my experiments showed that the BitTorrent content distribution method was on average 2.6 times faster at transferring the test file compared to the client/server method. Also, the BitTorrent method was shown to be highly efficient, with each peer transferring very similar amounts of data, showing equal resource usage. The performance testing concluded that there was very little difference in the PCs performance and usability when the BitTorrent system was active compared to when it was not. In the majority of tasks the test subjects were unable to find any performance difference from when BitTorrent was active and when it was not.

The combined results indicate that peer-to-peer file sharing in enterprise internetworking environment have several advantages in both speed and efficiency. It was found that BitTorrent was able to function without drastically reducing the usability of the user’s workstation. There are however, concerns of security and manageability which need to be addressed before implementation of such a system.

Acknowledgements

Thank you to the following people who assisted with this project and the production of this document.

- Jim Paterson, supervisor of this project, who helped greatly with his advice and feedback.

- Bram Cohen, who was kind enough to personally answer several questions I had regarding BitTorrent.

- Victoria X, who helped by proof-reading this document many times.

Index

1.0 Introduction

1.1 What Is Peer-to-Peer?

Without a doubt, one of the most exciting areas of networking today is peer-to-peer (P2P) networking. With P2P, the potential application possibilities seem to be almost limitless. Tasks that a single computer is unable to cope with alone are no longer an issue. P2P is all about harnessing the power of cooperation and sharing resources, computers connect directly to form networks and work together, combining resources to create distributed systems, some of which harness more computational power and dedicated storage than some of the worlds largest super computers.

However, working together to harness the power of distributed computing is not a new idea; parallel computation has been an integral component of scientific computing since the earliest days of computers. It has always been common practice to have several processors linked in a symmetric multiprocessing environment to perform calculations quicker. (Verbeke et. al, 2002) The fundamental goal of networking has always been the cooperation of computers and sharing of data but until very recently the cooperation aspect was seldom utilised on home computers.

1999 was a landmark year for P2P- That year, the first attempt was made to harness the collective power of millions of PCs across the Internet. One of the projects' names was "SETI@home". SETI@home is a scientific experiment that uses Internet-connected computers in a distributed computing fashion to download and analyse radio telescopic data in the search for signals from extraterrestrial intelligence. The combined processing power of hundreds of thousands of desktop computers all over the world is combined creating a virtual supercomputer. (Korpela et. al., 2001)

SETI@home's idea, although far from being a new concept, made the scientific community realise for the first time that there were other ways to complete research. No longer would smaller, independently funded, research groups be limited by the computational power of their own machines. Until the SETI project created the SETI@home client, the project had very limited funding and could not afford the millions of dollars required to build or hire a supercomputer to analyse their data. P2P and the Internet provided an alternative solution and SETI@home's creators were visionary enough to recognise this. SETI@home's solution is today becoming increasingly popular. More and more research organisations have released P2P clients that enable ordinary people to participate in their work. Another example of this is the National Foundation for Cancer Research (NFCR) client, which sells off participants' unused computer cycles to other companies, who in turn pay NFCR for this facility, essentially allowing users a way to make a donation of CPU time.

The ability to participate in a scientific programme is not something that is greatly appealing to the masses. Fortunately the very same year that the SETI@home project began and the client was made available for download,, a distinctly different use for P2P computing was developed by a college student. The system known as “Napster” quickly took the world by storm and grew to become one of the biggest pass-times on the Internet: large scale Internet file sharing was born.

Napster provided a way for complete strangers to share music MP3 files across the Internet, creating a central repository of music which looked like a single source. This enabled users to pick and choose which music to download onto their computers. Although Napster was later shut down due to copyright infringement the system paved the way for other file sharing networks such as Gnutella and Fasttrack. Unlike Napster, these networks can be used to share all kinds of data. SETI@home and Napster were the initial catalyst which inspired the academic community and led to a sudden surge of research into the area of P2P (Saroiu et. al., 2001). This in turn led to the power of P2P computing being harnessed in new and exciting ways. Users began creating search engines, hosting bandwidth intensive websites across a number of machines, creating storage systems, file systems, connecting directly to play games, having voice and video conversations (Singh & Schulzrinne, 2004) and creating powerful virtual super-computers (Minar & Hedlund, 2001). P2P file distribution has since become a proven, reliable method by which to distribute Internet content, in many cases providing far better distribution capabilities than that of the traditional client/server architecture (Cohen, 2003). Cohen estimates that his BitTorrent protocol can support one thousand to ten thousand times as many concurrent downloads as a single http server can. Many software publishers have taken advantage of this and use BitTorrent-like protocols to publish software updates and release new versions.

Some academics argue that P2P development will lead us back to the origins of the original Arpanet, an Internet where no computer is any more important than any other: no client/servers, every computer will be both client and server. At the moment this seems impossible, but with ever increasingly powerful machines and high-bandwidth domestic broadband connections becoming more common, this may some day become a reality. Whether of not this is a good thing is another matter for debate.

1.2 Defining Peer-to-Peer

Due to the sheer diversity that exists in the world of P2P computing it became necessary to classify each type of system and come up with a definition to try and gain some perspective of what a P2P system actually is. Throughout the course of my literature review it quickly became apparent that there was no single definition. Some distinctly different types of system have been labelled as P2P, from file sharing systems to telephone systems or network topologies.

The definition below by Metacomputing provides a "universal" definition; it could potentially be used to describe any types of resources that are connected by a network, making this definition far too broad for study.

> *"Peer-to-Peer networks can be described as just as a collection of heterogeneous distributed resources which are connected by a network. "*
>
> (Metacomputing, 2001)

In this paper, I will focus on definitions relating to the content distribution aspect of P2P.

One particular journal by Saroiu concluded that although the exact definition of P2P was debatable, the systems typically lacked dedicated, centralised infrastructure and depended on voluntary participation of peers to contribute resources. Together these machines would then construct the logical topology of the P2P network. (Saroiu et. al., 2001)

The problem would seem to be that P2P is actually a much bigger subject area than was first realised. Today it seems there is a potential application for P2P in every facet of computing, yet all of these unique systems are still classified under one singular category of "P2P". It is arguable there is a need for further categorisation of these systems. During my literature review I found the different types were as follows:

- File sharing (Napster, Gnutella, BitTorrent etc.),
- Message sharing (ICQ, MSN etc.)
- Distributed processing (SETI@home)
- Storage archival (LOCCKSS, Lionshare)
- Multiplayer video games (Quake, Unreal etc.)
- Voice and video conferencing (Net Meeting, Skype)

It is true however, that some applications incorporate many of the above features. A good example of this is ICQ messenger, which includes P2P file sharing, P2P web hosting, voice and video conferencing as well as the standard instant messaging service.

As my particular area of interest in P2P is on the content distribution/file sharing aspect, a more suitable definition for my purposes would be:

> *"Peer-to-peer systems can be characterized as distributed systems in which all nodes have identical capabilities and responsibilities and all communication is symmetric."*
>
> (Druschel & Rowstron, 2001)

Although Druschel & Rowstron state that all nodes have identical responsibilities and capabilities we will later find out that this is not always true- it will depend greatly on the architecture of P2P that is used. On the Internet today it is impossible to make such a system, as not all computers have the same resources available e.g. CPU speed, memory and bandwidth. Furthermore, communication can never really be symmetric on the Internet as most home users are connected by asymmetric connections (ADSL, ISDN or Dial up modem). All of these connections have greater downstream bandwidth than upstream. In order for Druschel & Rowstron's definition to apply, every computer on the P2P network would need to be of identical specification.

Regardless of its given name, P2P file sharing is about decentralising content delivery. It allows PCs or local servers to swap and update content more efficiently by circumventing a centralised repository, which in traditional networks would be a server. Instead it creates a virtual space that looks like a single source of information, but is actually made up of multiple, sometimes hundreds or thousands, of peer machines (dependant on the size of the network on which the system is deployed).

1.3 The Future of Peer-to-Peer

Although P2P file sharing systems have been popularised by Internet systems such as Napster, Kazaa, Limewire, eMule & BitTorrent, similar systems could also potentially be used on large scale corporate networks. The benefits P2P can provide would seem to be obvious to see:

- Costs would be reduced as no expensive server farm would be required (or it would be dramatically reduced in size) as P2P file sharing systems can achieve higher aggregate download capacities without the need for any additional expenditure for bandwidth or server farms. (Chawathe et. al., 2003)

- Decentralisation would result in the document/file/data being located nearest to the person/department/company that is best able to keep its contents up to date.(Smith, 2003)

- Redundant copies of files could be maintained on multiple workstations across the network. This could be used as archival or backup storage as used in "PAST" (Rowstron & Druschel, 2001)

- Bandwidth could be saved because it redistributes the cost of upload to downloaders, theoretically making content distribution to a limitless amount of peers feasible. (Cohen, 2003)

- Resources would be used efficiently as equal load is divided between peers.

- The availability of files/data would grow along with the demand for it (as opposed to availability dropping when a file/data becomes popular as is the case with client/server architectures). (Cohen, 2003)

- Almost limitless scalability could be provide as each new peer joining the network brings with it new potential resources.

In a world where the client/server model is the favoured method of content distribution, the benefits listed above are hugely attractive to companies looking for efficient methods of distributing products without additional computation, central storage or bandwidth expenditure.

Although these benefits would seem to be very appealing to companies, P2P has not been widely embraced by the business world. This may be due to issues of control and security. In addition, it would also appear that network administrators may simply be resistant to change. - the typical stereotype of a network manager is one of a paranoid control freak, there is almost a “fear” of the changes that P2P could bring. Clearly P2P has a long way to go to convince the business world of its viability.

In spite of the above, some companies have recognised and embraced the benefits that P2P can bring. Some of the most influential companies in the IT sector such as Intel, IBM, Sun Microsystems and Microsoft have demonstrated a commitment to P2P which can be seem from their recent and future products and their well-funded research initiatives. (Edwards, 2001)

In business these systems are termed "content networks" or "grid distribution" even though the principles and some of the technologies are very similar to their Internet file-sharing cousins. (Smith, 2003)

1.4 Project Aims

The aims of this project is to find out if these grid distribution network systems would be suitable on a large scale private network. The principle objective will be to test whether the download method known as “swarming” - a method of distributing a file from multiple peer machines simultaneously- would result in a quicker and more efficient download rate for peers than would be possible if they were downloading from a single dedicated server. The research will challenge the existing status quo of the need for dedicated server farms, and may partly settle the argument over which distribution method is superior.

The questions this research will address are as follows:

- Is it faster, any more or less efficient to transfer data though P2P file sharing than from a single dedicated server?

- Is there any associated performance penalty to peers when distributing files across a P2P file sharing network; if so is the penalty sufficient to affect the usability of the workstation?

Using a series of controlled experiments I will endeavour to reach a definitive answer to both questions.

To answer the first question I will set up a small scale network to test the efficiency of BitTorrent versus the client/server method. I determined a network of five machines would be suitable for this purpose, I will first transfer a file of a fixed size from one machine to all others (one to many). This will simulate client/server architecture. The transfer time would be timed from start to finish. I will then enable the file sharing service on all machines and begin transferring the file again. This time the swarming method will be used; again this test will be timed. From the results, I hope to gather quantitative results that could be used to determine the speed and efficiency of the P2P swarming method versus the client/server round-robin method. By examining the results it should be clear which method of distribution is superior.

In order to answer the second question, I need to gauge the performance impact on peers of using a P2P file sharing system. I plan to monitor and log the resource usage of CPU, RAM, disk access and bandwidth to measure the amount of the systems resources the P2P application will consume while the file transfer is in operation. Using a resource metre program, this part of the testing will produce quantitative data that will show how the P2P system uses peer resources. This would however be nothing but meaningless data without a method of gauging the amount of these resources which can be consumed before the system performance degrades to unusable levels. To this end, another test is needed that will demonstrate the “real life” performance impact on peers. To do this I plan to have test subjects carry out a number of tasks on the computer while the P2P service is active. I will then have them fill in a performance questionnaire at each activity and allow them to state how well the system performed. In order to make a comparison I will ask them to perform the same tests again while the P2P service is not running. To achieve unbiased results, the subjects will not know if the P2P service is active. It is my hope that by analysing these quantative results that I will be able to determine whether the P2P file sharing system had a significant enough performance impact on the system to degrade performance so that users find the computer slow or unusable. This will generate quantitative data which may provide insight as to the viability of the P2P system.

2.0 Literature Review

As mentioned above, P2P computing is a wide area. There are many subtopics and issues which have not been investigated and problems which have not been solved. The following significant subtopics/issues appeared most in the literature review:

2.1 Architectures of Peer-to-Peer

There are two distinctively different architectures of P2P networks. The term "architectures" describes the logical manner by which data moves inside a P2P network as opposed to the physical layout of the network. It is arguable that the architecture upon which the P2P application is based has a substantial impact on the performance, efficiency, scalability and reliability of the network. The two subsets this paper will discuss are:

- Pure
- Hybrid

This paper will not consider "Mixed P2P architecture" as this term is only used to reflect slight variations of pure and hybrid architectures normally associated with closed source propriety network systems such as Kazaa/Grokster.

2.1.1 Pure Peer-to-Peer

Pure P2P clients will not generally use a central server, except to initially connect to the network. The entire network consists of interconnected peers. Upon joining the network, the peer will receive the IP addresses of some of the neighbouring peers on the network. When searching for files/data the client will send out a broadcast to these addresses and ask them if they have a match to the client's request. They will then also broadcast the received search query out to the peers they are aware of. The process will then be repeated until a match is found. If a peer has files/data that match the query request, that peer will then reply directly to the peer that initiated the query. Each search query packet has a predetermined time-to-live. Upon expiry of this time, the packet will be dropped. This unfortunately leads to a great deal of bandwidth being used on search queries alone. Very often, by the time the search results are returned, the peer has disconnected from the network. For this reason this type of solution may have difficulty scaling to large numbers of peers or complex queries. (Yang & Garcia-Molina, 2003) This has caused pure P2P networks to become quite unpopular because they generate a lot of overhead traffic in order to keep the network up and running. (Backx et. al., 2002)

(Clarke et. al., 2001) Clarke argues the case that pure P2P offers a great deal of anonymity which cannot be achieved on Hybrid networks. In terms of making P2P attractive to companies, anonymity would not be considered an advantage . In a business environment, companies would require individual employees to be held accountable for how they choose to use the network. In the event of irresponsible behaviour on the network, such as the sharing of illicit materials, the systems administrator should be able to trace the person responsible.

One of the defining characteristics of a Pure P2P network is that all peers should be considered and treated equally. No peer should be any more or less important than any other peer on the network. This characteristic has resulted in what were once considered Pure P2P networks being re-classified as Hybrid networks. An example of this is the Gnutella network onto which “ultra peers”, were introduced to help alleviate the searching problems on the network.

Although the idea of a Pure P2P network sounds appealing as it sounds similar to a “democracy of networks” , Gnutella proved that currently Pure P2P is simply not feasible on the Internet or in the workplace. There will never be such a thing as a large sized company with computer systems which are all of equal capability.

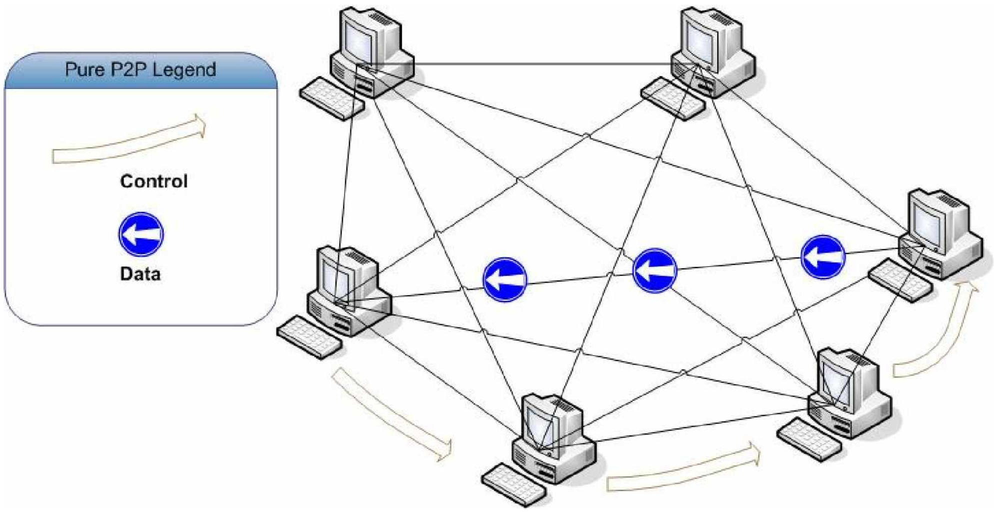


Figure 1: A pure P2P network query request and answer.

2.1.2 Hybrid Peer-to-Peer

A problem for P2P networks in general, and for file sharing in particular, is locating resources. This is commonly described as the “lookup problem”. (Balakrishnan et. al., 2003) Hybrid architecture serves as a means of solving this problem, Instead of having a network of peers where all peers communicate equally. there would be some peers with additional roles and responsibilities on the network. These responsibilities could involve the indexing of content available on the network, the logging of IP addresses that have joined the network and the communication of sending these addresses to the relevant peer upon finding a suitable match stored in its index.

(Yang & Garcia-Molina, 2003) Yang & Garcia-Molina in their paper “Comparing Hybrid Peer-to-Peer Systems” concluded that Hybrid file-sharing systems have better performance than Pure systems because tasks such as searching and indexing of content can be done much more efficiently in a centralised manner. This was further supported by Backx (Backx et. al., 2002) in the study “A comparison of peer-to-peer architectures”.

Typically the systems that would be assigned these additional duties on the network would be those most capable of handling the extra tasks. This would seem to make sense given that all computers and data connections are not of equal specification.

A good reason for supporting Hybrid architectures over Pure architectures is demonstrated by the events of the year 2000 when the Gnutella network experienced almost complete collapse. The influx of new peers migrating from the recently closed Napster network rendered the network unable to cope with the extra load due to its Pure P2P architecture. A study by a popular website (K. Truelove, 2000) later found that these problems were caused by peers connected via dialup modems. These modems were saturated by the increased load, dying and fragmenting the network due to their departure. The reason for this was because these low bandwidth analogue modems were given equal roles and responsibilities to peers with dedicated high-speed digital data connections. Yang & Garcia-Molina (Yang & Garcia-Molina, 2003) concluded that due to this problem, in order to make P2P work in an efficient manner, the system should take advantage of the diverse heterogeneity present on its network. This ultimately means assigning greater responsibility to those who are more capable of handling it.

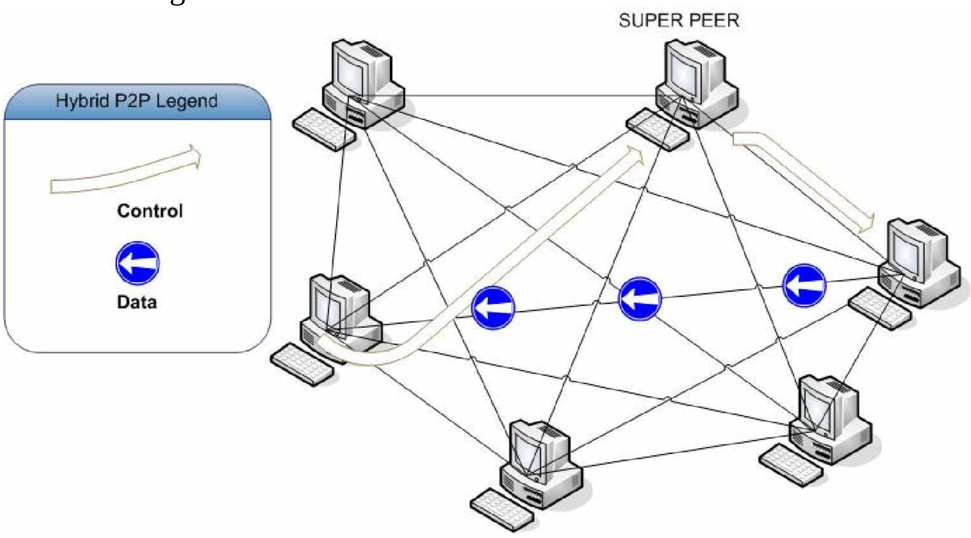


Figure 2: A Hybrid P2P network query request and answer.

2.1.3 Architecture Conclusion

In an ideal world, Pure P2P would be the fairest and better system, , however, the problems associated with the architecture in a “real world” environment make pure P2P unfeasible. Although all people are created equally, sadly all computers and data connections are not. For now it would seem that Hybrid architecture seems to have the edge in both performance and scalability. Perhaps this is in part due to the failing of the early Pure P2P networks. In the future, as more and more people gain access to high-speed data connections, perhaps some of these problems will be solved. For the moment the P2P development community has noted the problems associated with Pure architectures and as a result file sharing systems are evolving towards Hybrid systems.

2.2 Segmented Distribution with Distributed Hash Tables

In order for a P2P file sharing application to be able to function, peers need to have the ability to share data, bandwidth and computer resources such as memory and CPU time. However, during the course of the literature review I found that distinct approaches for sharing these resources are being adopted by different file sharing systems.

On the first generation of P2P file sharing applications such as Napster and Audiogalaxy, entire files needed to be transferred to another peer machine before they could be shared. If a host is sharing a file, and that file is highly sought after on the network, multiple peers may try to download the file at the same time. This could result in a very slow distribution of this file as each peer is required to download the entire file before another copy becomes available. If the file was of quite a large size this could take a very long time, resulting in the other peers not being able to access the file. Early P2P applications found this was acceptable, and worked around the problem by enabling each peer to set the maximum number of peers that could download from them at a single time. Peers with less bandwidth could set the amount lower, enabling only a few peers to download simultaneously. The rest of the peers would then queue for the file, resulting in quicker distribution to the initial downloaders. Once downloaded, another copy of the file would become available for download by other peers. Another method used by early P2P applications was "round-robin scheduling". This was the method whereby the amount of downloading peers was not limited, and each peer received a portion of the distributing peer's bandwidth. (Ma et. al., 2003) This method of distribution was however found to be inefficient.

Distributed hash tables (DHTs) provided a solution to this problem and are today considered an essential component in efficient P2P networks. A number of research DHTs have been developed by universities, picked up by the Open Source community and implemented. DHTs are a class of distributed systems that partition ownership of a set of keys amongst participating peers. This allows efficient routing of messages to the unique owner of any given key. DHTs are typically designed to scale to large numbers of peers and to handle continual new peer connections and departures/failures. (Wiley, 2003)

Some of the first DHT algorithms such as CAN (Ratnasamy et. al., 2001), Pastry (Rowstron & Druschel, 2001) and Tapestry (Zhao et. al., 2001) have been modified to create new DHT algorithms. This remains an area of continual research within the academic community. The aim of the above applications is similar: the creation of efficient decentralised lookup systems that connect to form an overlay network that is self organising and fault tolerant while providing load balancing in an application-independent manner. (Rowstron & Druschel, 2001)

DHTs have been implemented in the popular file sharing protocol BitTorrent as the basis of the "swarming protocol" where files are broken down into smaller pieces to be transferred in a swarm like fashion between peers. (Cohen, 2003) DHTs have also been incorporated into the Coral content distribution network, where WebPages can be accessed though a distributed P2P web proxy system. This enables websites with limited bandwidth to scale better.

From the file sharing perspective, the end result is that not only is the user able to download a selected file from another peer, but in the event that there is another copy of the same file available on another peer the user will begin to simultaneously download the file from both sources at once. This is possible because of the way data is segmented. This method also eliminates the problems faced by early networks such as data corruption caused by downloading different files on the network with the same name. This is because each piece of the file is assigned a digital fingerprint which is checked and verified on the receiving host. If any piece is corrupt or does not have the matching hash, it is simply re-downloaded and the sender of corrupt data is blocked from sending any more.

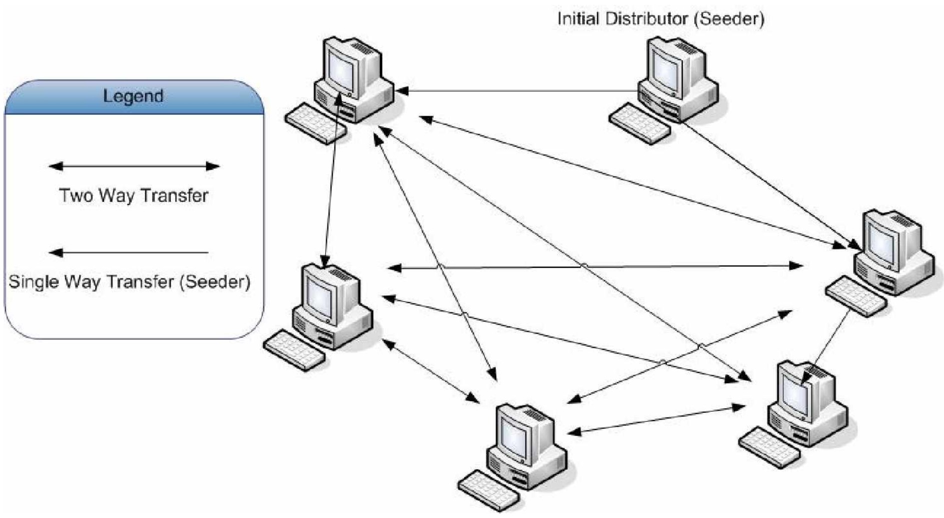


Figure 3: Swarming content delivery

2.3 Free-Riding

A common problem found on file sharing networks such as Gnutella, Freenet and Fastrack is that often peers share very little or no data. One study by Adar & Huberman (Adar & Huberman, 2000) found that in monitoring the Gnutella network for a 24 hour period, 70% of the peers shared no files and nearly 50% of all responses are returned by the top 1% of sharing hosts. Adar & Huberman argue that the practice of free-riding on P2P file sharing networks leads to degradation of performance of the network. However, upon reading in detail their experimental setup, I would argue that a 24 hour period may not have been enough time to monitor the network from which to draw these conclusions. Adar & Huberman make no mention of other factors such as the time of year, the day of the week or the time at which the experiment took place. Therefore, it is difficult to accept the validity of such results.

Due to the perceived severity of the problem of free-riding, there has been much research in to providing incentives to share data on P2P file sharing systems. Early P2P applications considered using simple “participation level” systems, whereby the user would be allocated an amount of credit on the network. Uploading data to other peers would then give the user more credit, and downloading from other peers would cause it to decrease.(Buragohain et. al.,2003) This method was, however, unsuccessful for a number of reasons:

- Many other “hacked” applications that connected to the same networks existed. These would typically set a user’s participation level to the maximum available, meaning he never had to upload any data.

- The client would give the user a set amount of “participation credit” the first time he ran the program. This would allow him to download some files straight away when he first joined the network. However, when this credit ran out, many users simply uninstalled the client and re-installed it again or installed a different P2P application, giving them more free participation credit without the cost of contributing back into the system.

- Some users did share a great deal of files, however the files were not highly desirable. This brings in a whole new quantity versus quality argument. Should someone who shares garbage-data be awarded credit?

The only file sharing system to have eliminated the problem of free-riding is BitTorrent. In BitTorrent, files are split up into pieces. As a result, the downloaders of the file barter amongst themselves for pieces they do not have in a “tit-for-tat” like manner. This method has been found to be hugely successful at preventing parasitic-like behaviour. Each peer is responsible for maximising its download rate by contacting suitable peers with which to barter. Essentially a peer’s download rate is proportional to its upload rate. When a peer has finished downloading a file, it may become a “seed” by staying online for a while and sharing the file for free, i.e., without bartering. (Pouwelse, et. al., 2004)

After investigating BitTorrent’s user FAQ, I reached the conclusion that even BitTorrent’s system may not be perfect. The documentation makes mention of difficulties peers may have in connecting to clients if they are behind a NAT/PAT router or firewall. Administrators need to first configure port forwarding on these devices in order to allow peers to connect and download pieces. The failure to configure port forwarding results in slow transfer rates, however the security implications of opening these ports have not been discussed. There may also be another way to look at this “problem” - the need to configure NAT/PAT in order for BitTorrent to function correctly outside a private network could be considered a security feature.

The impact free-riding could have on a file sharing network in an enterprise internetworking environment could potentially be the same as that on the Gnutella network. This gives merit to the work of Cohen (Cohen, 2003), supporting the argument that future P2P systems on private networks should adopt this or a similar incentives method.

2.4 Security

Security concerns have unfortunately inhibited widespread use of P2P networks in a business environment. The major concern is that some of the peers in networks are malicious and may wish to corrupt the network's behaviour. Many network administrators also feel that perhaps the greatest vulnerability associated with P2P applications is that most of them can be used to turn any computer into a network file server. This increases the chances of both intentional and unintentional sharing of intellectual property. There are also vulnerabilities such as exposed log files and even client and network denial-of-service conditions caused by poorly written programs, heavy traffic, or even worse, huge files filling up hard drives.

Malice can take various forms, from Virus/Trojan planting to denial-of-service attacks. Without adequate security mechanisms and settings, users are especially vulnerable. Although numerous researchers in P2P and ad-hoc networking communities have worked on addressing some of these flaws, existing solutions are still considered to be limited.

Shieh & Wallach (Shieh & Wallach, 2005) determined that solving security problems without being able to trust nodes to operate correctly is quite difficult. Furthermore, without any kind of central, trusted administration these security problems are likely to remain open research problems.

Security threats on P2P file sharing systems can fall into the following categories:

- Viruses/ worms/ trojan horses
- Distributed Denial of Service attacks (DDoS attacks)
- Spyware/ ad-ware

2.4.1 Viruses, Worms and Trojan Horses

Viruses and worms have always been a problem on the Internet and in computing in general. Since P2P computing became more wide-scale, however, there have been increasing amounts of viruses/ worms aimed at targeting users of P2P networks. An article by a popular website "The Register" by Thornton (Thornton, 2001) found that 6% of all music downloaded on Kazaa is in fact not music at all. Rather, users are unknowingly downloading viruses and trojans. Clearly this is a problem on the Internet, but within a private network this is equally threatening. Perhaps the solution to this kind of problem on a private network could possibly be the screening of content before it is added, or a filtering system whereby potentially dangerous files are filtered from searches. Neither solution would be ideal. In the white paper published by Net-Markets Europe in 2001, the proposal is to adopt a Public Key Infrastructure (PKI) similar to that which is currently used on the Internet for secure transactions. (Bond, 2001) While a system such as this would no doubt increase security, it would however require a centralised certification authority to issue these certificates, which would further complicate the system.

Cohen (Cohen, 2003) argues that his system, BitTorrent, is a much safer service than other P2P networks due to the way in which it functions. While other P2P services allow a certain degree of access to a shared folder or to someone's hard drive, BitTorrent users cannot share anything outside of the desired file type that is in an open BitTorrent window. Due to the fact that you are only downloading pieces of the file as opposed to the full file, it also makes it incredibly difficult (if not impossible) to transmit viruses through the BitTorrent system.

2.4.2 DDoS Attacks

DDoS attacks are nothing new on the Internet either. The basic premise is that of an attacker exploiting a large number of hosts to concurrently send seemingly legitimate packets to an intended victim host machine. The goal of the attack is to consume all resources of the target, effectively stopping the target from either sending or receiving data. In the paper "Exploiting P2P Systems for DDoS Attacks" by Naoumov & Ross (Naoumov & Ross, 2005) the possibility of exploiting P2P systems to create DDoS attacks was examined and the conclusion reached was that some P2P systems can potentially be exploited in this way. Although the study only examined one particular P2P network, there could also be such vulnerabilities in others.

2.4.3 Spyware and Ad-Ware

Another major threat currently on the Internet is the unwanted software known as spyware or ad-ware. According to Hackworth (Hackworth, 2005), spyware has been a persistent problem present on computers since the 1980s, when key-stroke loggers were discovered on computers at university campuses. Today, many programs which can be downloaded from the Internet contain either spyware or ad-ware. Usually the revenue generated by the information retrieved is how developers are able to give software away for free. There are many different kinds of spyware. The term is generic and is similar to using the term "virus" to define all malicious programs. Like viruses, users often do not realise they have installed spyware or ad-ware. The problem is that when many people install software, they often fail to read the documentation or user licence agreement that accompanies it.

The basic premise of spyware or ad-ware is usually to collect data that can be used for some purpose. Hackworth (Hackworth, 2005) mentions in his study that the groups most likely to use spyware/ ad-ware to gather data are:

- Online attackers and organised crime
- Marketing organisations
- Trusted insiders

The most commonly targeted data includes:

- Internet activity
- Email and contact information
- Windows PStore data (user accounts, email passwords etc)
- Clipboard contents

- Keystrokes
- Screenshots
- Network traffic

Hackworth (Hackworth, 2005) suggested that undoubtedly the far most dangerous group of those listed above for any P2P file sharing system in an enterprise internetworking environment is the “Trusted Insider”, who could be an employee who has a grudge against the company.

Clearly spyware/ ad-ware poses a significant threat to intellectual property, especially in P2P networking environments, due to lack of centralised management or control.

Curtin (Curtin, 2004) discusses many ways by which to nullify the spyware threat using internal network systems rather than individual host prevention measures. Curtin recommends the configuring of intrusion detection systems, firewalls, and other policy enforcement mechanisms to prevent spyware packages from working. Curtin (Curtin, 2004) also recommends that the first step in being able to do this is to first identify potentially unsafe content (e.g., ActiveX controls). Another means would be to block attempts of spyware to “phone home," effectively preventing them from being able to report their activity, but not preventing the spyware from being installed in the user's system in the first place. A third mechanism would be to enforce a policy that refuses connectivity from trusted systems to unknown sites or to allow downloads of unidentifiable content.

2.5 Legal Issues

One of the major issues that many people have with P2P is the abundance of illegal downloads that are available on these public file sharing networks. The major concern that companies have is the possibility that their network could potentially be misused in this way. Not only would this incriminate the company if the network was misused, it could potentially leave them liable to legal action.

This seems to be a subject where there are relatively few ideas as to how to control content. Known methods are:

- An acceptable usage agreement could be setup on the network that would state that users are responsible for the way they use the network themselves. Education of network users is also vital. This would however still leave the company liable to legal action if it failed to take action to remove this type of content from the network. In the past, Napster pleaded ignorance as to what its users were sharing and was closed down as a result.

- Lack of anonymity would be considered a security feature on a system deployed on a private network. On Internet systems, users often want a greater level of anonymity, affording them the protection of being able to share the data they wish to, without having to worry about legal consequences. This has made developers of P2P applications build-in anonymity features to better hide the identity of their users. However, systems such as Lionshare

(Lionshare, 2004) enable the sharing of academic documents over the Internet and hope to discourage users from sharing copyrighted material by including no support for such anonymity. Users are required to register for the service and “log in”. It is hoped that greater risk of legal action will discourage this kind of misuse of the network.

In the paper “Content and control” (Slater, et. al., 2005) the conclusion reached is that it was not possible to fully control content shared on P2P networks; for the time being we need to rely on user education and policies set by government and other relevant authorities.

3.0 Hypotheses

- It will be faster and more efficient to download from multiple peers than a single server. There should be a notable decrease in file transfer time for the computers connected to the branch on the test network due to the swarming/bartering for pieces between machines on the branch network. I do not believe the workstation in the headquarters will receive the file any faster.

- There will be a slight performance penalty to peers sharing files across the network. A typical sharing peer will notice an increased amount of CPU cycles being used, more RAM will be used, more hard disk space will be occupied, the hard disk will be accessed more often and less bandwidth will be used between the workstations and server due to the bartering for pieces between workstations.

- I do not believe that the increased resource usage will be sufficient to be noticeable or cause a problem to the average user performing day-to-day tasks on their machine.

4.0 Methods

The following section will describe in detail the exact methods that will be used to test my hypotheses:

4.1 Experiment One – Speed and Efficiency Test

The speed and efficiency test experiment which I wish to carry out is to test how a P2P application which employs a swarming protocol performs in an enterprise internetworking environment compared to traditional client/server architectures. I wish to know if there is any benefit in performance and efficiency of downloading data in a swarming fashion compared to downloading from a traditional client/server architecture, which would spread load evenly in a round-robin scheduling-like manner.

4.1.1 Design

The testing of my hypotheses requires the construction of a small test network. I have detailed the layout and addressing scheme in Appendix one. I have run a simulation of the proposed network design in Cisco's packet tracer to make sure there were no unforeseen problems with the design. Although I specify here that all computers should be of equal specification, this will not always be true in a "real life" network environment. I made an argument above for use of Hybrid P2P systems. Although this is the way the system would most likely be used, for the purpose of testing it is better if the machines are all of equal specification. This provides a more reliable set of results that can be analysed. If machines of various specifications were used it would be more difficult to determine if slower transfer rates or problems faced by the network were due to PC/component bottlenecks rather than problems with the swarming method itself. Each computer will have an identical operating system installed (Microsoft Windows XP professional edition). The version used will have service pack one installed and the Windows firewall will be disabled. All machines will have the following hardware specifications:

- Intel Pentium 4 1500Ghz (Willamette 180nm core)
- Microstar MS-6534 Motherboard
- 256MB RAM
- Maxtor 6Y08L0 80GB (Partitioned to 19GB and using NTFS)
- 3com Etherlink XL10/100 Ethernet (100Mbps Mode)

Software installed and used in this test will be as follows:

- BitTornado client (http://www.bittornado.com/)
- BnBT Easy Tracker (http://bnbt.depthstrike.com/) (seeding machine only)
- MakeTorrent 2 (http://krypt.dyndns.org:81/torrent/maketorrent/) (seeding machine only)
- Multi-track Stopwatch (http://www2.neweb.ne.jp/wd/morimoto/en/mwatch/index.html)
- IIS Web server installed on WS1
- Test file "hnd.rar" 21.6MB

There are two stages to my testing. The first is to ascertain how long it will take to simultaneously distribute the test file to all machines on the network by the traditional client/server round-robin method. The transfer of the test file will be started on all machines at once and will use a batch file to ensure transfer begins simultaneously. This will be a one-to-many test. The test will be timed, and the time of each computer receiving the full file will be recorded using Multi-track stopwatch (see Appendix two).

The second stage is to test whether the transfer would be quicker using the BitTorrent swarming content delivery method. The file will be seeded on a single machine in the headquarters network and all machines in the network will then begin to download the file using the BitTorrent client (many-to-many). Again, the download completion time of each machine will be recorded.

In the of the BitTorrent method, details regarding how much data was transferred from each individual peer will be recorded. This will allow me to determine how efficient the swarming method is in locating resources and load balancing between all peers on the network.

4.1.2 Apparatus

Equipment required will consist of the following:

- 2X Cisco Catalyst 2950 switches
- 2X Cisco 2800 routers
- 5X Identical computers (Specification listed previously)
- Network Cabling (Category 5e twisted pair)

The network will be configured in accordance with the addressing scheme as detailed in Appendix one. The addressing scheme is simple and although it does not accurately reflect the kind of addressing scheme used in large scale company networks this will have no bearing on the test results. The serial link between the two departments is the most critical part of the design, as this will show how the computers connected through the slow serial link will benefit from the swarming distribution technique, with each computer bartering with the others for pieces of the file instead of trying to download the file as a whole from the main branch server over the slow serial link.

The choice of routing protocol, RIP, is also not a constraint that will affect the performance of the network. Larger scale routing protocols such as OSPF could have been used, but this would not have any impact on the results. The serial link speed between the two branches is set to 128k and is present to simulate a frame relay connection between the branch and headquarters.

4.1.3 Procedure

- Set up network (IP addresses/router configurations);
- Install and configure software;
- Place the test file on seeding machine;
- Set up IIS web server on branch “seeding” machine;
- Create a simple index page with the link to the file to be downloaded;
- Simultaneously from all four computers begin to download the file, start the counter on Multi-track stopwatch program;
- Fill in log of when each machine finishes the transfer;
- Delete the file from each machine;
- Start up the BNBT tracker program on the seeding machine;
- Create the torrent file with MakeTorrent, default data hashing settings used and upload to tracker;
- Have the seed begin seeding the test file;
- Have each machine download the torrent file;

- Start downloading from all machines via BitTorrent client. Starting Multi-track stopwatch to time the download;
- Fill in log of when each machine finishes the transfer; and
- Analyse results.

4.2 Experiment Two – Resource Usage Test

This experiment is to gauge the performance impact on peers of using a P2P file sharing system. I plan to monitor and log the systems resource usage of CPU, RAM, disk access and bandwidth to measure how much of the system's resources the P2P application would consume while the file transfer was in operation using a resource metre program.

Software installed and used in this test is as follows:

- Bandwidth Monitor Pro (http://www.bandwidthmonitorpro.com/)
- Free Meter (http://www.freemeter.us)
- Sissoft Sandra lite 2005 (http://www.sissoftware.co.uk)

4.2.1 Design

This test will be performed on one computer attached to the branch network. Firstly I will run the test on an idle machine with no non-essential programs or background processes running (using ctrl-alt-del to verify this in the task manager). I will then record CPU, RAM, disk access and bandwidth usage when the system is in an idle state for sixty seconds. The reason for this is that the operating system and system device drivers may already be using vital system resources. If this check was not performed first, the test may return inaccurate results. The CPU, RAM, disk access and bandwidth utilisation will be monitored for a period of sixty seconds. During this time I will log the usage of the aforementioned resources at five second intervals. (See Appendix three for example log).

The next stage of testing will then begin, which will involve the monitoring of the same resources for the same time period. The difference is that this time the computer will have the BitTorrent program running and will be actively engaged in swarming activity. Once again the resource usage will be logged at five second intervals for sixty seconds.

The final stage of this test will require the use of a popular benchmark suite, SisSoft Sandra lite edition 2005. This is a static benchmark suite that performs a series of tests which indicate how the system is performing. In SisSoft Sandra I will perform the "CPU arithmetic benchmark", the "CPU multi-media benchmark", the "file system benchmark", and the "network/LAN bandwidth benchmark". These tests will be performed twice, once while no P2P service is running and once when BitTorrent is engaged in swarming activity. Each test will produce a score and by monitoring and comparing scores under both conditions it will be possible to gauge how much system/network performance is lost when the P2P service is running.

4.2.2 Apparatus

Equipment will consist of the same equipment used previously for experiment one. The same test file will be transferred to each PC.

4.2.3 Procedure

- Set up network (IP addresses/router configurations);
- Install & configure software;
- Place the test file on seeding machine;
- Test network conditions using ping;
- Clear all system running processes and programs;
- Begin monitoring system resource usage through resource metre & free metre logging usage every five seconds;
- Start up the BNBT tracker program on the seeding machine;
- Start downloading from all machines via BitTorrent client;
- Begin monitoring system resource usage through resource metre - logging usage every five seconds;
- Run Sisoft Sandra benchmark suite while BitTorrent is not running;
- Run Sisoft Sandra benchmark suite while BitTorrent is running;
- Log results; and
- Analyse results.

4.3 Experiment Three – Performance Impact Test

The previous experiment should show the quantity of system resources which are consumed by the P2P file sharing application, in the forms of numbers and graphs. Whilst these may or may not show how the computer is either degraded in performance or not different, they cannot possibly show how the effect of the consumption of these resources affects the usability of the workstation for the user. Due to this “human” factor, I would have to determine an acceptable usage level of resources that the P2P program could use without affecting the usability of the workstation. To do this I plan to have test subjects carry out a number of tasks on the computer while the P2P service is running. The subjects will be required to fill in a performance questionnaire at each activity on which they must state how well the system performed (see Appendix four). In order to make a comparison I will repeat the same tests while the service is not running. To achieve unbiased results, the subjects will not know if the P2P service is active. It is my hope that by analysing these quantatitive results I will be able to determine whether the P2P file sharing system has had a significant enough performance impact on the system to degrade performance so that users find the computer slow or unusable or just the same as if these resources were not consumed.

4.3.1 Design

This test requires ten test subjects in total. These subjects will be from various backgrounds, reflecting the fact that some users in a company environment will be more computer competent than others. The diverse mix is required as people with more knowledge of computers may be able to determine whether or not the P2P client is active, due to his experience of how well a particular system/processor configuration should perform. I will recruit my test subjects from various departments of the university to achieve the competency mix.

The subject will be given a performance log for completion (see Appendix four). The instructions will be clear and testing should not take any longer than ten minutes in total. I will prepare the system as necessary, either turning on the BitTorrent client or turning it off without the subject being aware. There will also be a small piece of card stuck over the right hand corner of the monitor to prevent them from seeing the BitTorrent icon in the system tray.

Software installed and/or used in this test is as follows:

- Microsoft Wordpad
- Microsoft Internet Explorer 6
- Test game – “Unreal Tournament” (http://www.unreal.com)

Testing will be performed on the PC “WS1” on the HQ side of the network. This will eliminate the problem of the slow serial link being blamed for some of the speed issues.

4.3.2 Subjects

The experiment requires a minimum group size of ten users, with varied levels of competency. To achieve a suitable mix of subjects, I recruited them by sending out a generic e-mail asking for volunteers in advance.of the experiment. I sent the e-mail to a variety of university departments, to ensure as far as possible that there would be a mix in the subjects’ levels of computer competency.

No importance has been placed on the test subjects’ age, sex or occupation, , The test will be based on user opinion, as in a real life environment there would be a diverse mix of people using the system, not all of which will know anything about computers or P2P networks.

4.3.3 Apparatus

The network built previously for the other experiments will be used. No additional equipment is required.

4.3.4 Procedure

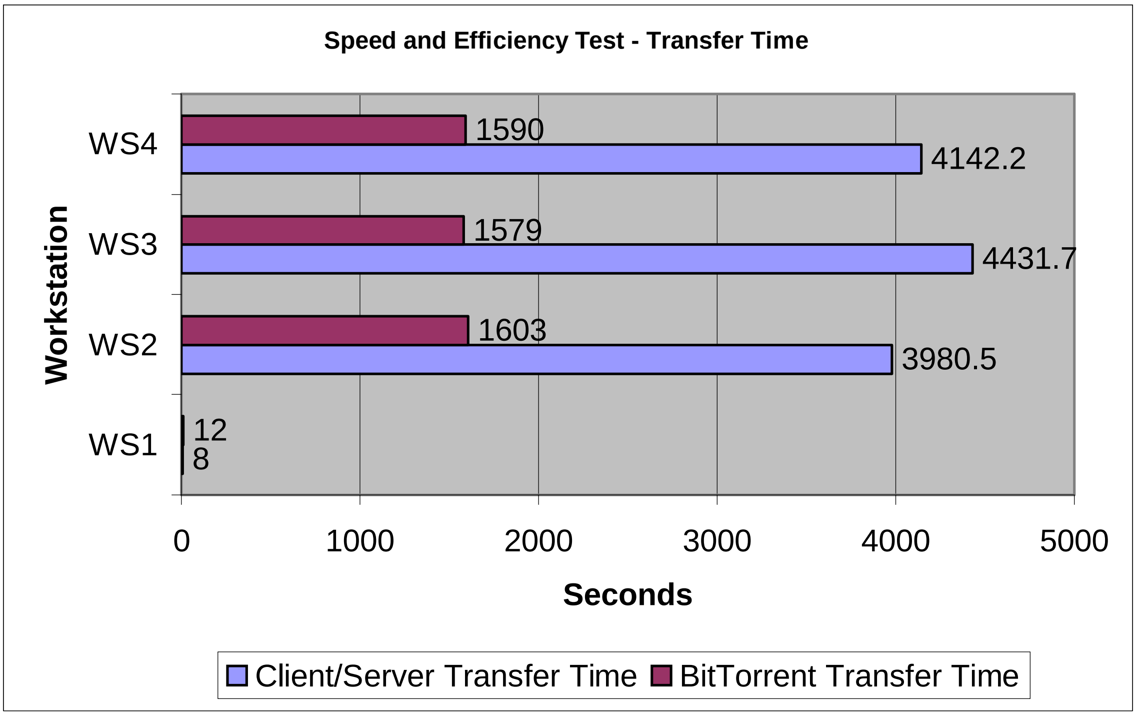
- The same configuration will be used as for previous tests;
- When the subject arrives I will explain what I wish them to do and explain what the tests hope to achieve;
- I will ask the subject to look away while I configure the PC. I will then disable the P2P file sharing system;
- I will ask the subject to begin testing and completing the performance log;
- Once the subject has completed his tasks, I will ask him to look away while I enable the file sharing system;
- I will give the user a new performance log and ask him to repeat the same tests he performed previously and to complete the performance log again;
- I will repeat tests with other subjects; and
- I will analyse the data.

5.0 Results

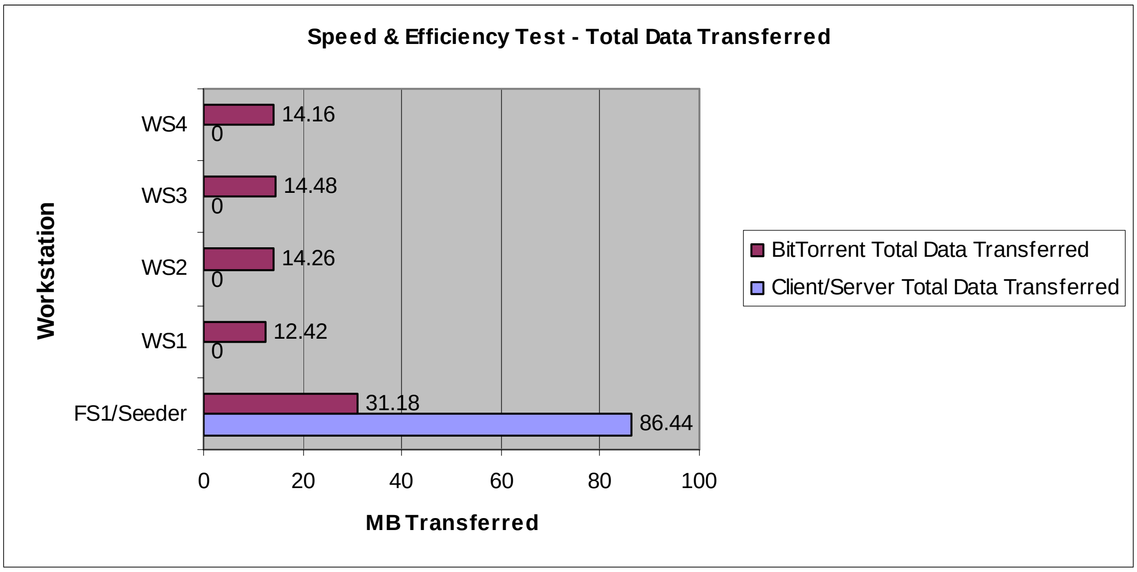
5.1 Experiment One - Speed and Efficiency Test Results

This test was designed to determine the advantages (if any) that can be gained from using a swarming method to transfer files as opposed to using traditional client/ server architecture. The chart below summarises the time taken to transfer the test file by swarming method (shown in burgundy) and client/server method (shown in blue).

As can be seen from the chart below, the time for transferring data by swarming method would seem to indicate that BitTorrent is able to transfer the test file to computers connected on the branch side of the network much faster than traditional client/server architectures. It is also worth noting that in transferring data from the seeder to WS1 (both machines on same network segment), BitTorrent is somewhat slower than the client/server method.



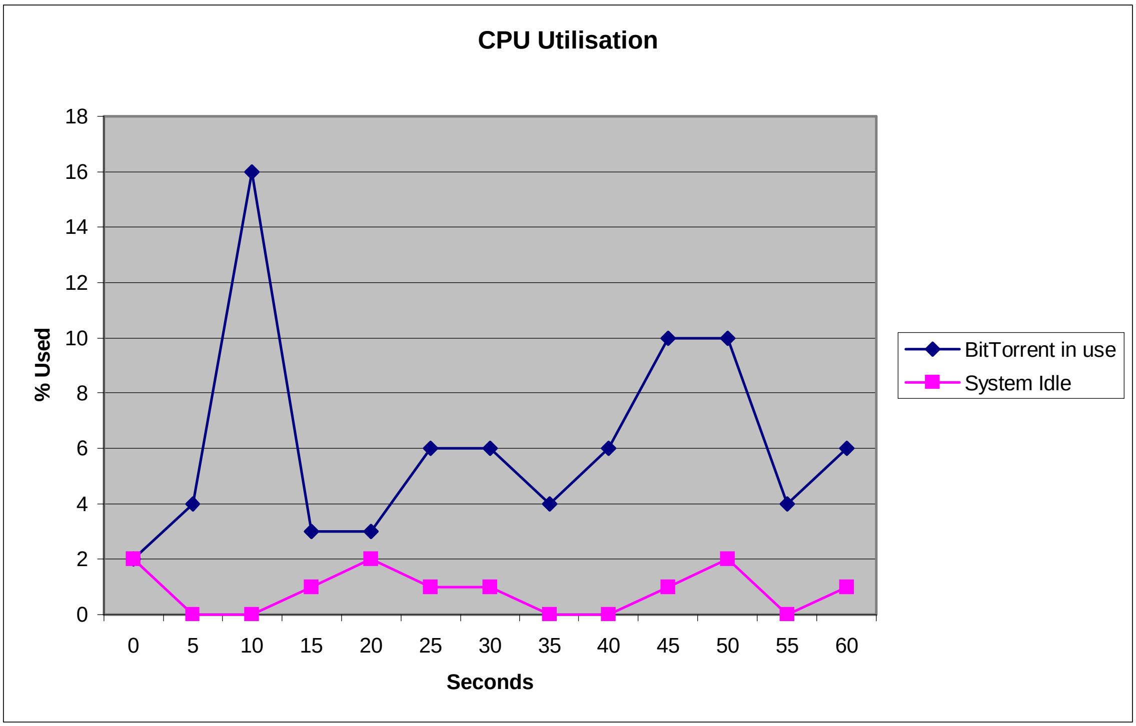
The bar chart below shows the total amount of data transferred by each host on the network by both BitTorrent swarming method and by client/server method. Obviously this shows that under normal client/server operation, the peers do not contribute to the transfer of the file. The chart also shows that by BitTorrent's method, the peers on the branch side of the network transfer very similar amounts of data due to the bartering for pieces in a "tit-for-tat" manner. The chart also shows the total data transferred from the main file server/seeder machine (FS1) for BitTorrent. This shows that despite the bartering for pieces between peers, the seeding machine still had to transfer more than twice as much data as other peer machines. The raw data collected in this test is viewable in Appendix five.



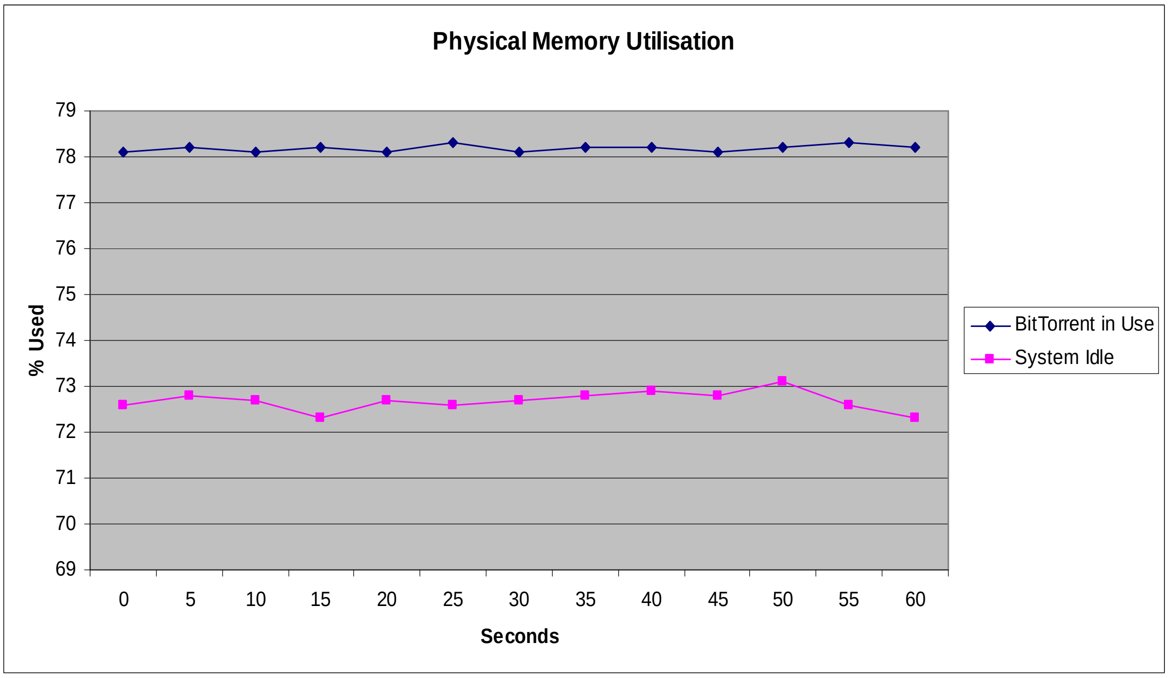
5.2 Experiment Two – Resource Usage Test Results

The purpose of this test was to determine the additional resources the P2P file sharing system would consume, compared to when the system was in an idle state. The graph below shows CPU utilisation for a peer when the BitTorrent client is activated and it is actively engaged in the swarming process compared to when the system is in an idle state. The blue line represents BitTorrent client CPU utilisation, and the magenta line represents normal CPU utilisation while the system is in an idle state.

As can be seen from the graph below, BitTorrent CPU usage peaks at sixteen percent CPU utilisation, and constantly fluctuates at levels far below this.

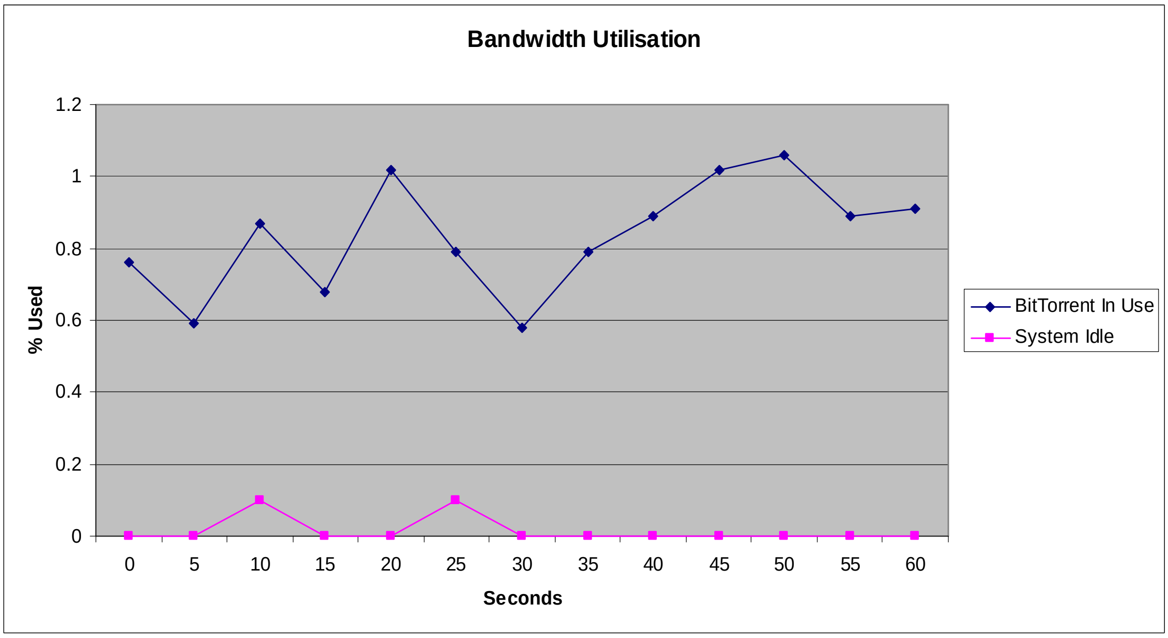


The next graph below shows system physical memory utilisation (RAM) while the BitTorrent system is running and actively engaged in the swarming process compared to when the system is in an idle state. The blue line represents BitTorrent memory utilisation, and the magenta line represents the system in an idle state. As can be derived from the results, BitTorrent at all times uses more than an additional 5% of physical memory.



The next graph below shows the percentage of system bandwidth utilisation while the BitTorrent system is running and actively engaged in the swarming process compared to when the system is in an idle state.

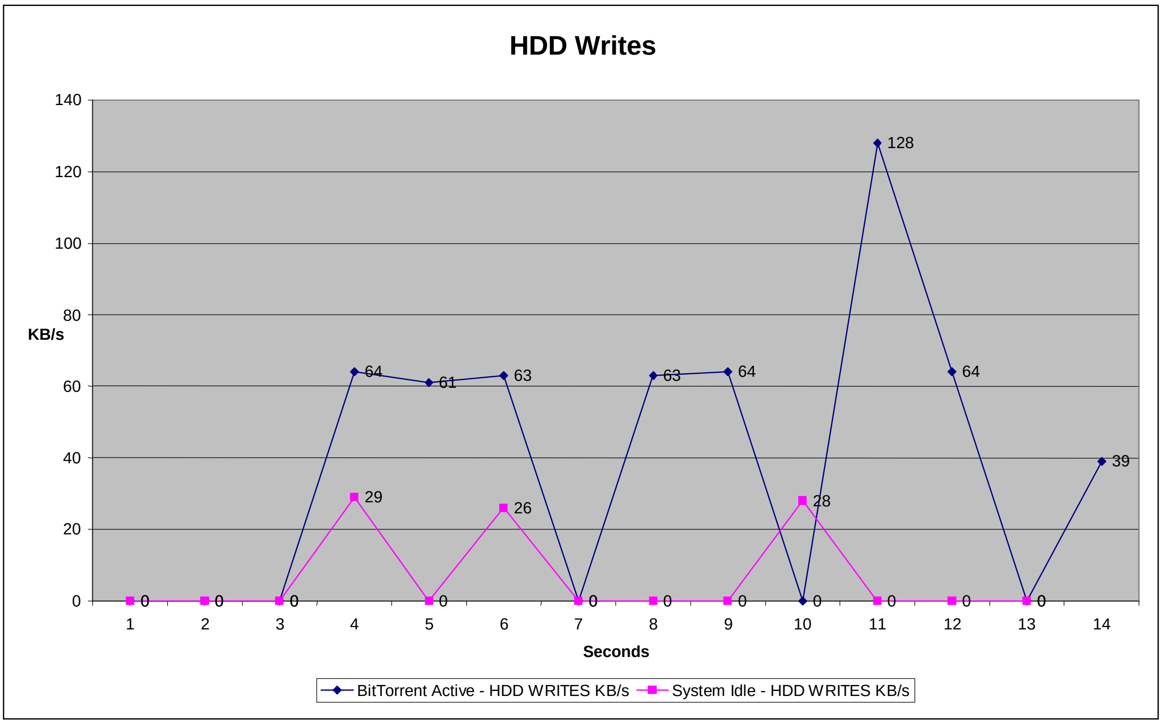
The system tested was on the branch network connected via a 100Mb fast Ethernet network interface card. The graph shows that BitTorrent is using little over one percent of the systems bandwidth in some cases and far less than this the majority of the time. By comparison, when the system is in idle state, no bandwidth is being used.



The following graph shows the amount of data being written to the hard disk drive (HDD) over the course of the sixty seconds the system was monitored. The blue line represents BitTorrent activity, and the magenta line represents the normal activity when the system is in an idle state.

As can be seen, BitTorrent HDD writes peak at 128KB/s and for the majority of the experiment are greatly below this.

System idle HDD writes peak at 29KB/s and is possibly due to windows writing data to the page file/virtual memory.



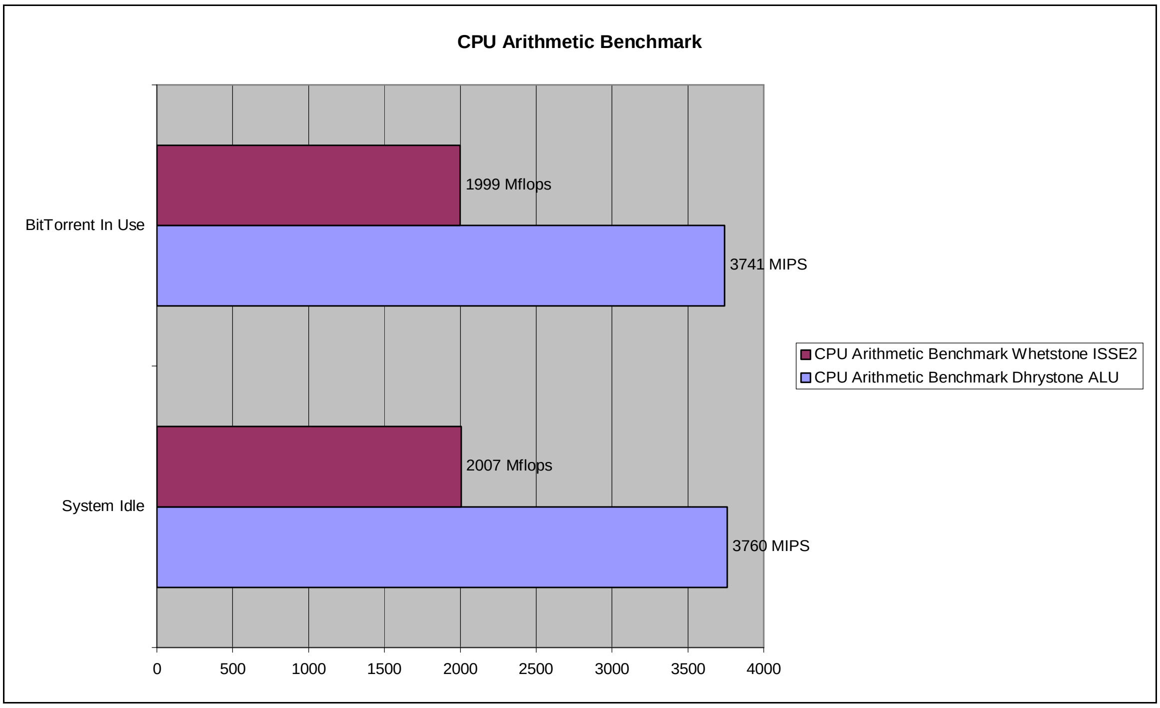
The following bar charts are results from the static benchmark program SisSoft Sandra 2005 lite edition. The aim of this part of the test was to see how the use of BitTorrent would affect the performance of the system in a benchmark environment. Should we see considerable differences in the benchmark results from when BitTorrent is active compared to when the system is in idle state we should be able to understand better how the use of the BitTorrent program affects the performance of the workstation.

The first benchmark ran in SisSoft Sandra was the CPU Arithmetic Benchmark - a higher result for this test is better. There are two components of this test, first is the “Whetstone ISSE2” (represented by the burgundy bar) and second is the “Dhrystone ALU” (represented by the blue bar).

The Whetstone ISSE2 test is measured in Mega-floating point operations per second (MFLOPS) and is used as a measure of how powerful a CPU is, MFLOPS gauge the capability of the system to deal with floating-point math.

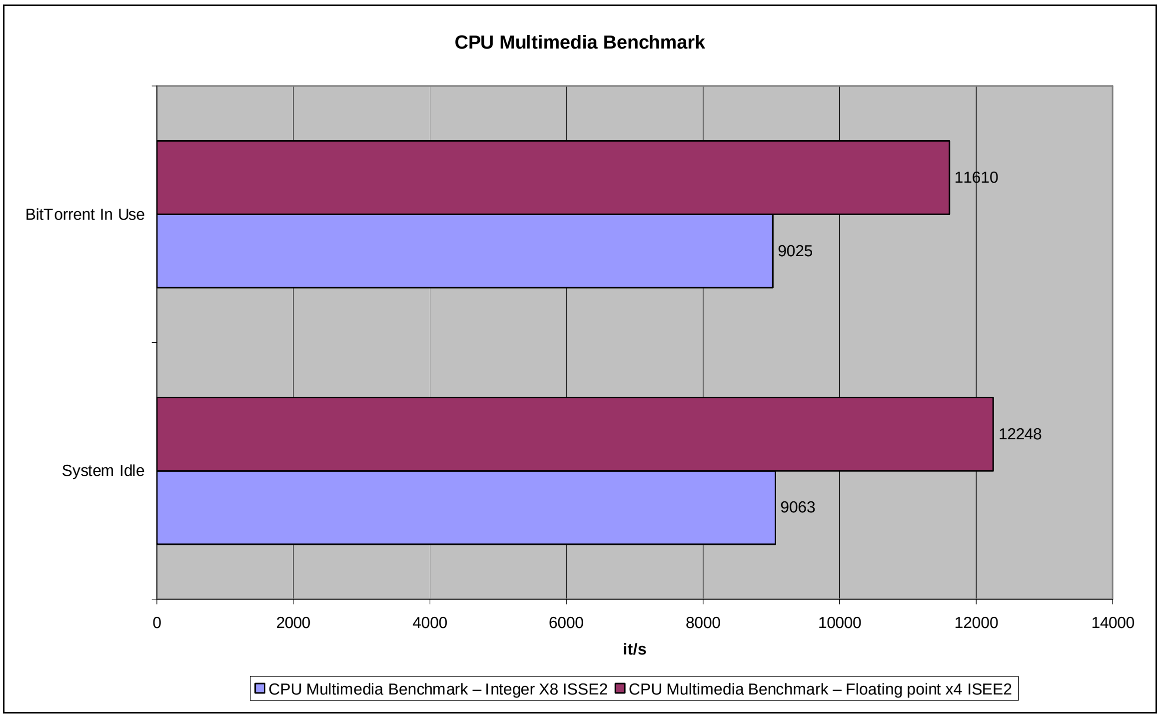
The Dhrystone ALU test is measured in million instructions per second (MIPS) and is used to gauge the capability of the CPU to handle instructions.

As can be seen from the chart below, there is very little difference in the benchmark results from when BitTorrent is in use and when the system is idle. This could indicate that using BitTorrent only very slightly reduces the CPU's ability to handle instructions and carry out floating-point math.

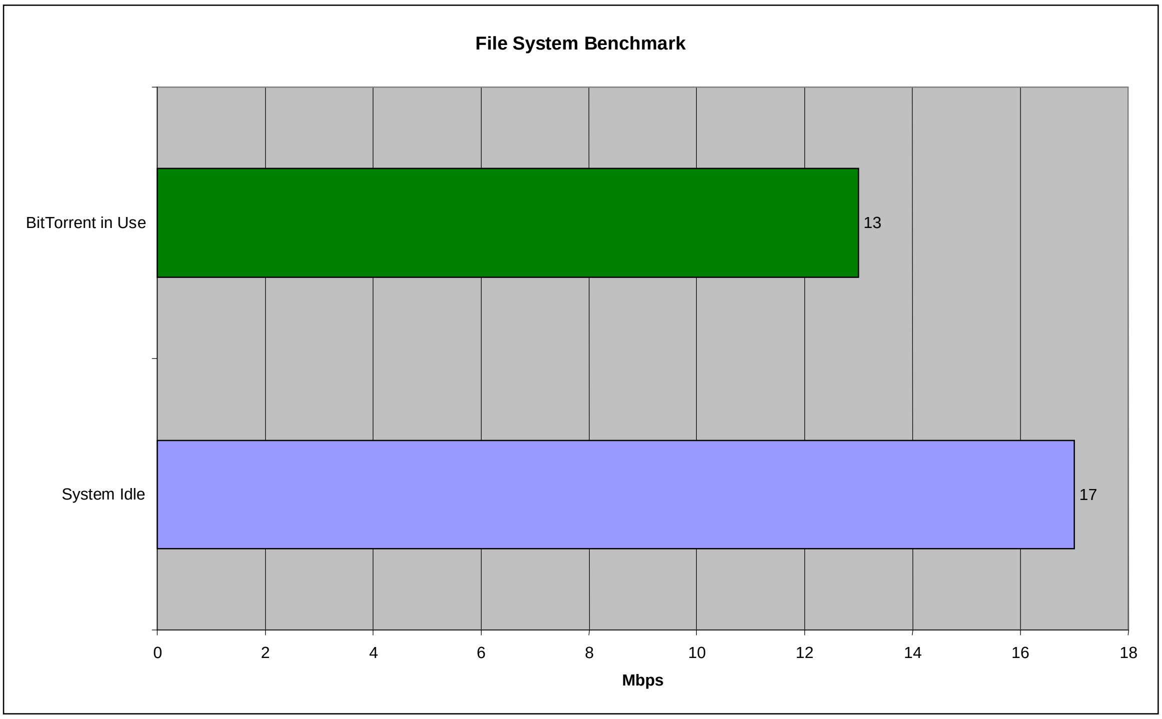


The next bar chart shows how using BitTorrent will affect the CPUs ability to carry out multimedia processing. A higher result in this test indicates increased performance. There are two components to this test also, the first is "Integer X8 ISSE2" (represented by the blue bar) and the second is "floating-point X4 ISSE2" (represented by the burgundy bar). The test is measured in integers per second.

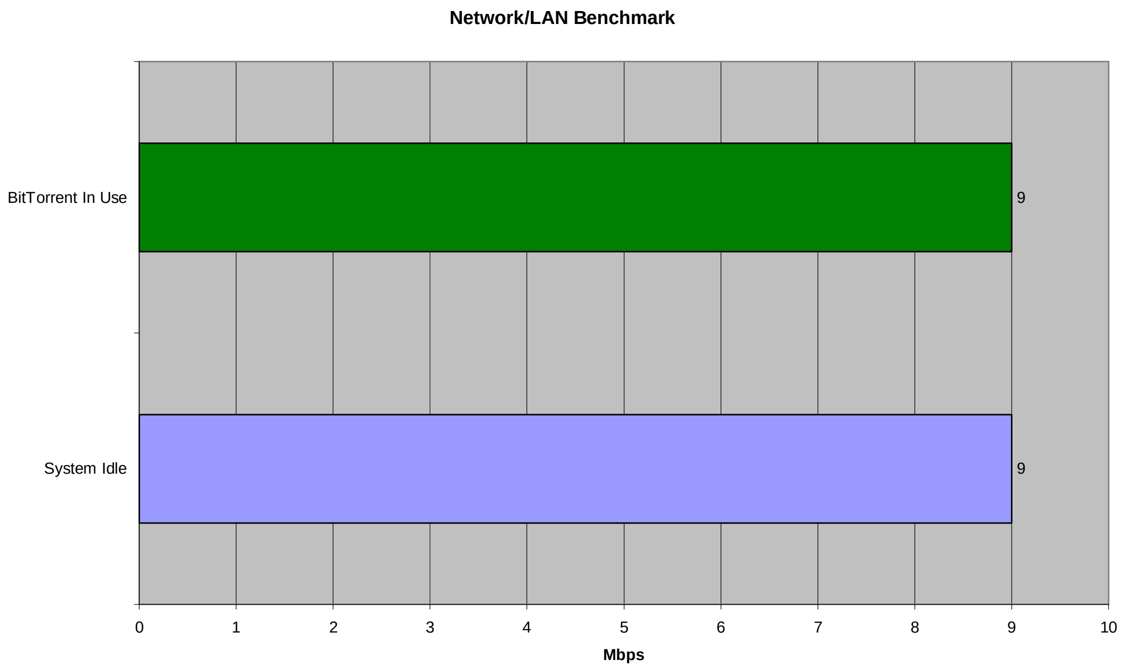
As can be seen from the chart, it would appear that the use of BitTorrent has little effect on the benchmark results, suggesting that using BitTorrent does not greatly reduce the CPU ability to perform multimedia calculations.



The next bar chart shows how the PC file system performs while the BitTorrent client is operating and actively engaged in the swarming process compared to when the system is in an idle state. The blue bar represents the file system performance when the system is in an idle state and the green bar represents the file system performance when BitTorrent is in use. As can be seen from the chart, BitTorrent seems to decrease the file system performance.



The network/LAN benchmark in SisSoft Sandra is designed to benchmark the available bandwidth of a particular network segment. In this case the bandwidth measurement was taken between the workstation on the branch network and the branch router. The results of this test would seem to indicate that BitTorrent is not using enough bandwidth to show a measurable difference in this particular benchmark.



5.3 Experiment Three – Performance Impact Test Results

The purpose of this test is to determine whether BitTorrent being in use is sufficient to degrade the performance of the PC so that users would find the PC slow or unusable. Each of the ten subjects performed the same tests twice, once while the BitTorrent service was active, and once when it was not. The subjects were not told if the BitTorrent service was running, and asked to fill in a performance log while they undertook each test. After performing the test for the first time, the subjects' completed performance logs were taken away from them before they were given another one.

There were a total of ten activities/ questions on the performance log. Each question had the option of a "positive", "neutral" or "negative" answer. A positive answer indicates that the user is completely satisfied with the performance of the PC in that particular test; a neutral answer indicates that the user is neither satisfied nor dissatisfied in that particular test and a negative indicates the user is completely dissatisfied with the PC performance in that particular test. By examining the differences in subject's answers while the BitTorrent system is active and while it is not we can gain insight as to how the use of BitTorrent may affect the performance of peer machines in a "real life" environment.

Below summary tables of the test results:

System Idle

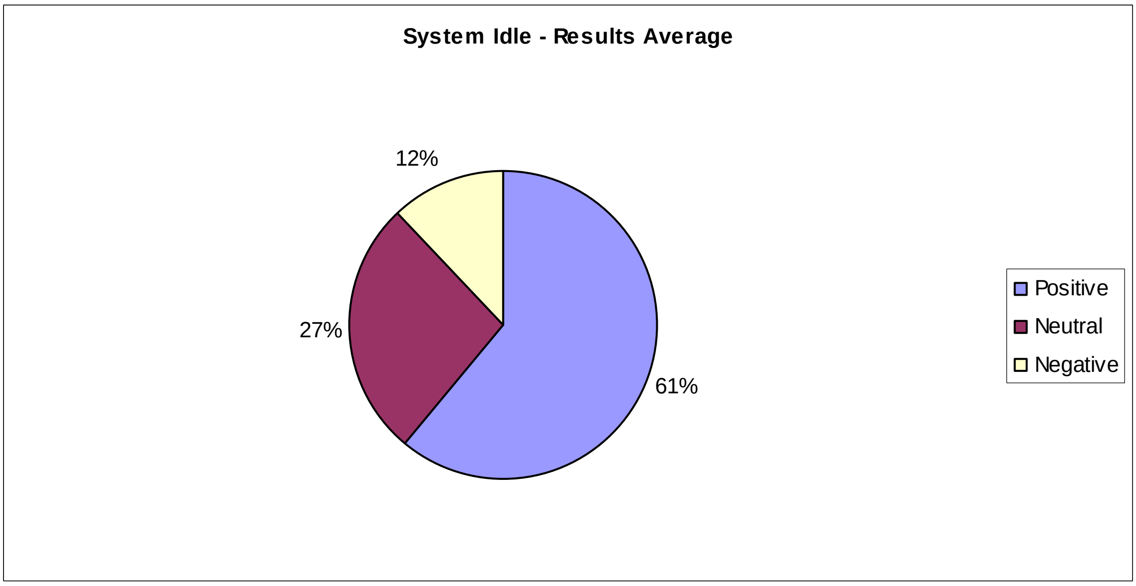
Question Number/ Answer	Positive	Neutral	Negative
1) MS Wordpad Load Speed	9	1	0
2) MS Wordpad Type Lag	9	1	0
3) MS Wordpad Save Speed	8	2	0
4) IE Load Speed	6	3	1
5) IE Page Retrieval Speed	5	4	1
6) IE Page Load Speed	4	4	2
7) Network Browse Speed	4	3	3
8) Network File Copy Speed	3	5	2
9) Game Load Speed	7	2	1
10) Game Performance	6	2	2

BitTorrent Active

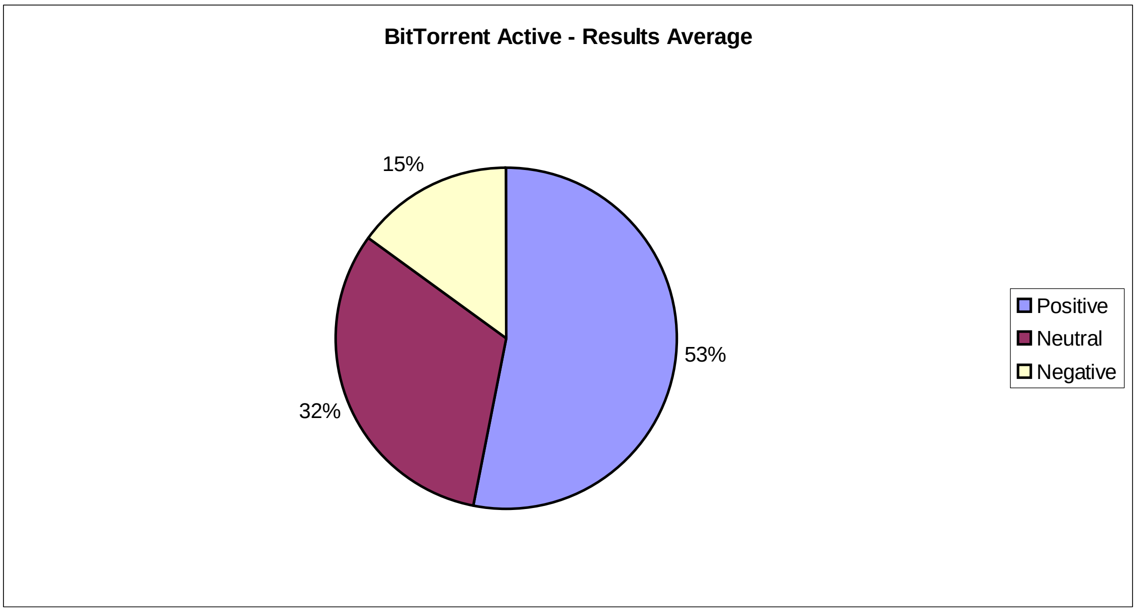
Question Number/ Answer	Positive	Neutral	Negative
1) MS Wordpad Load Speed	10	0	0
2) MS Wordpad Type Lag	9	1	0
3) MS Wordpad Save Speed	7	3	0
4) IE Load Speed	6	3	1
5) IE Page Retrieval Speed	4	3	3
6) IE Page Load Speed	2	6	2
7) Network Browse Speed	4	4	2
8) Network File Copy Speed	1	5	4
9) Game Load Speed	5	4	1
10) Game Performance	5	3	2

As is shown, despite the occasional exception, the number of positive answers decreased, and the number of neutral and negative answers increased when BitTorrent was used.

The next pie chart shows the results average of positive, neutral and negative answers as a percentage for the questionnaire when the system is an idle state:



The next pie chart show the results average of positive, neutral and negative answers as a percentage for the questionnaire when the BitTorrent system in active:



6.0 Discussion of Results

This section of the report will consider how to interpret the test results, and try to understand why certain results were obtained, it should also show how the project links with other peoples work and will discuss the successes and failures of the project, it will also discuss ways in which to improve and extend upon it.

6.1 Discussion

The purpose of this study was to determine the feasibility of using P2P technology in an enterprise internetworking environment. The tests were designed to compare the existing client/server technology with P2P swarming technology. The question here was not only whether P2P could do the same job, but also to ascertain if there would be any advantages or disadvantages of using such a system. Another aim of the experiments was to find out how the use of such a system might affect the performance of the peers attached to the network.

Experiment One – Speed and Efficiency Test

The speed and efficiency test was used to determine how the speed of the swarming content distribution method compared to traditional client/server architectures.

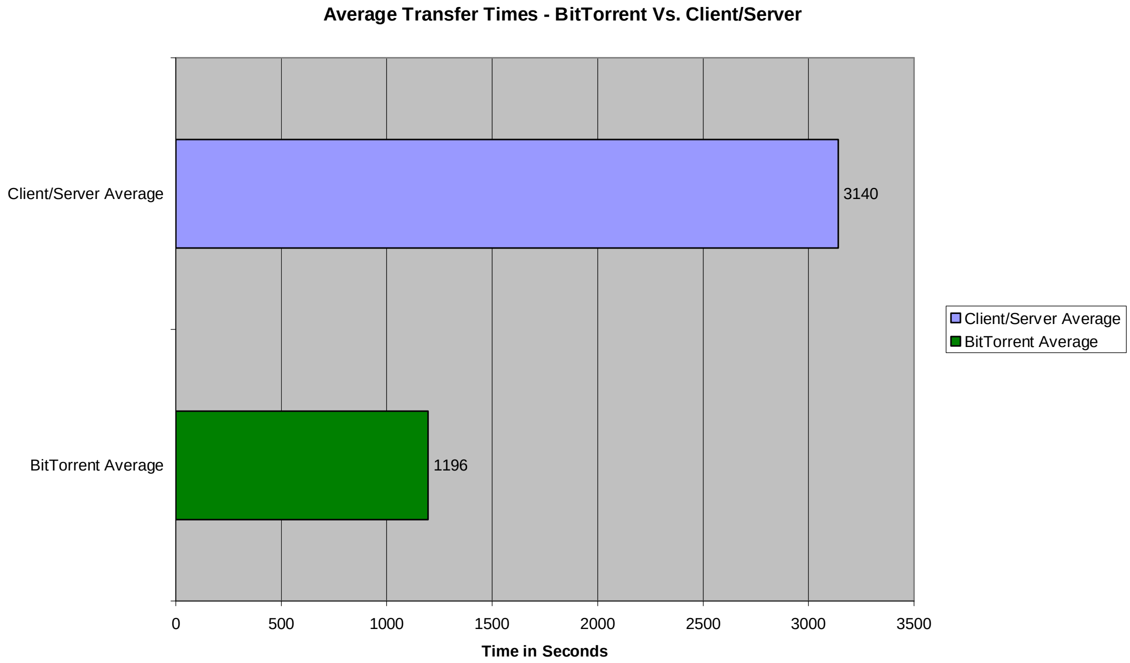
My hypothesis was:

> *"It will be faster and more efficient to download from multiple peers than a single server. There should be a notable decrease in file transfer time for the computers connected to the branch on the test network due to the swarming/bartering for pieces between machines on the branch network. I do not believe the workstation in the headquarters will receive the file any faster."*

The results obtained were on the whole very positive, and arguably support my hypothesis that it is both faster and more efficient to transfer data through a P2P file sharing system than it is to download from a single dedicated server.

The results in the "speed and efficiency test – transfer time" diagram show that the workstations connected to the branch server received the file at more than double the speed as a result of using the BitTorrent swarming method compared to the traditional client/server method.

If we average the results of all transfer times via BitTorrent vs. client/ server method we can see that on average, the BitTorrent method is 2.6 times quicker.



This would appear to be due to the bartering between peers in a "tit-for-tat" manner for pieces of the file, thus creating the effect of having multiple proxy servers located on the network and each supplying unique pieces of the file. Peers were able to retrieve pieces of the file from their neighbouring workstations instead of connecting directly to a remote file server. The interesting thing about how these peers received the file was that each workstation also had a connection to the seeding machine and each workstation chose to download different pieces of the file from the seeder. This meant that each peer contained new and different pieces of the file from their local peers. This is supported by examining the "speed and efficiency – total data transferred" graph, in which it shows how each workstation on the branch network transferred very similar amounts of data. This would indicate how well BitTorrent is able to organise the swarm, so that the load is shared quite equally across these peers.

Another interesting result obtained from this is the speed by which the local "WS1" computer received the file. It would seem by examining the "speed and efficiency test – transfer time" diagram that the BitTorrent method was actually slower at transferring the file than the client/server method. The reason this happened was quite interesting. When WS1 started to receive the file by BitTorrent, it first had to contact the seeding machine to request the file, however, at this point, the seeding machine had not transferred a significant enough amount of data to other peers to allow WS1 to download anything from them. WS1 effectively received the entire test file from the seeding machine, but before BitTorrent began sending the file, it checked the network first to see if any other peers existed that could participate. This, along with the somewhat lengthy handshaking procedure each new client connecting to a BitTorrent swarm must complete accounted for some of the extra time required to download the file. When WS1 had received all the pieces of the file, it then had to assemble them. By comparison the client/server method transferred the full file in its entirety, with no need for the receiving workstation re-assemble pieces of the file.

This also accounts for the extra data that the BitTorrent seeder had to transfer. On the "speed and efficiency – total data transferred" bar chart this extra amount is shown. At first glance it seems very inefficient that a seeder had to upload twice as much data as WS1, even though both workstations are located on the same network. If we consider, however, that the connection speed between WS1 and the seeder is 100Mbit this makes a great deal of sense, as there was nowhere else from which WS1 could download the data other than from the seeder. It would not be efficient for WS1 to have to wait to download pieces of the file from the distant branch network when it had a dedicated 100Mbit connection direct to the file. This seems to suggest that BitTorrent offers the best of both worlds, minus a slight penalty in download time. However, if there were more than one peer on the HQ network receiving the file, the results would show more even levels of bandwidth usage due to bartering between peer machines.

In hind-sight it may have been better to have at least two peer machines on the HQ side of the network so that a more equal transfer ratio would have been shown in the results.

6.1.1 Experiment Two – Resource Usage Test

The purpose of this test was to determine how much of the systems resources a workstation would use when the P2P system was in use compared to when the system was idle. The results were displayed in the previous section on many different graphs and charts.

My hypothesis for this test was as follows:

> *"There will be a slight performance penalty to peers sharing files across the network. A typical sharing peer will notice an increased amount of CPU cycles being used, more RAM will be used, more hard disk space will be occupied, the hard disk will be accessed more often and less bandwidth will be used between the workstations and server due to the bartering for pieces between workstations.."*

By examining the CPU utilisation graph it is arguable that my hypothesis is supported. The workstations did show increased CPU utilisation when the BitTorrent service was active. Interestingly the CPU utilisation never exceeded 16% and for the majority of the test it was well below this, the average over the course of a sixty second test was just 6.15%.

The physical memory graph supports my hypothesis that more physical memory will be used. In the case of my results, more than an extra 5% was utilised, since the test machine had 256MB total physical memory. This 5% accounts for 1.9 % of the machine's total physical memory which in this case was an extra 5MB. Considering these workstations are now quite old and are considered to be very low end, with most new machines having at least 1GB of RAM, this 1.9% of memory utilisation would be considered acceptable. There may however be more testing required in this area with differing sizes of files and torrent files which are split into different size pieces. In this test I used the default piece sizes – these may not always be optimal depending on the size of the file being shared.

The results shown in the bandwidth utilisation graph showed that of the test machines 100Mbps connection that a maximum of 1% was utilised, showing that through use of the swarming technique a speed of 1Mbps was attainable in some cases even though the connection between the branch and HQ only had a 128kB/s connection.

The HDD writes graph shows that over the time monitored the biggest write was 128KB and most of the results gathered were measured at 64KB/s. This could indicate possible block sizes of data which BitTorrent is writing to disk after being received. By comparison, we can see that the system idle process writes tiny sized blocks of data at somewhat regular intervals, possibly due to the operating system writing new data to virtual memory/page file.

A potential way to improve the accuracy of this test would be to log the resource usage at smaller intervals over a larger time-frame. Whilst the results gathered by our test were are useful, they represent only a fraction of the activity the computer components were performing, for example CPU usage, could be logged in milliseconds or even better in yoctoseconds because these CPUs are able to perform calculations far more quickly than humans. , Furthermore, reducing the logging intervals would improve accuracy of results. A problem here is also that, by monitoring the resources we are actually using more than we normally would. Much like any other program, the system resource meter uses resources.

The results obtained by using SisSoft Sandra 2005 lite offer conclusive evidence that the performance of the system is decreased as a result of using the BitTorrent client compared to when the system is in an idle state. There is a clear difference in the results obtained.

- The CPU Arithmetic benchmark results, while not substantial, show a difference in the CPU performance under both conditions tested. This provides evidence that the CPU's ability to perform floating-point operations and carry out instructions is decreased as a result of using the BitTorrent client.

- The CPU Multimedia benchmark results show the biggest change of the two CPU tests performed. Performance is lower when BitTorrent is in use. This could indicate decreased multimedia performance while BitTorrent is in operation, suggesting that video performance could drop a few frames or a game may not run quite as smoothly. This was later supported by the results obtained in the Performance impact test.

- The file system benchmark results showed that the use of BitTorrent causes the file system to decrease in performance. As a result it may take longer for other data to be written to the hard disk, programs may take longer to load and files take longer to be saved.

- The Network/LAN benchmark results on the other hand were inconclusive. No difference could be seen in performance either when BitTorrent was or was not active. This could be due to the amount of bandwidth used being so small that this benchmark was unable to measure the difference.

6.1.2 Experiment Three – Performance Impact Test

The purpose of this test was to determine the “real life” effect that BitTorrent would have on the performance of a peer. The previous experiment did show that there was a performance penalty to peers while using BitTorrent compared to when the system was in an idle state, however, without performing this test there was no way to determine how the consumption of those extra resources would effect the usability of the peers.

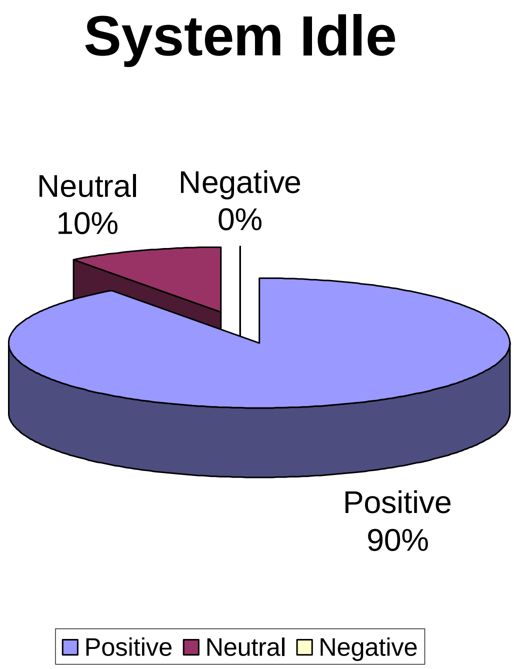
My hypothesis regarding this test was:

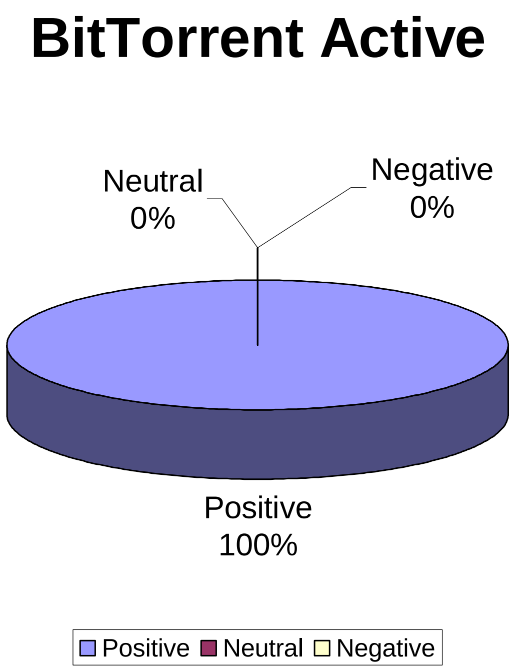
> *“I do not believe that the increased resource usage will be sufficient to be noticeable or cause a problem to the average user performing day-to-day tasks on their machine.”*

The tests performed were a variety of day-to-day tasks on the workstation, and were designed to test all the same performance aspects as in the previous test, but this time in a “real life” environment. Users were asked to fill in a separate performance log when the BitTorrent system was active, and once when it was not. The results (viewable in the previous section) showed that in certain tasks the number of negative answers increased.

The following pages are an individual break-down of the answers received in the experiment. It is necessary to analyse the results step-by-step:

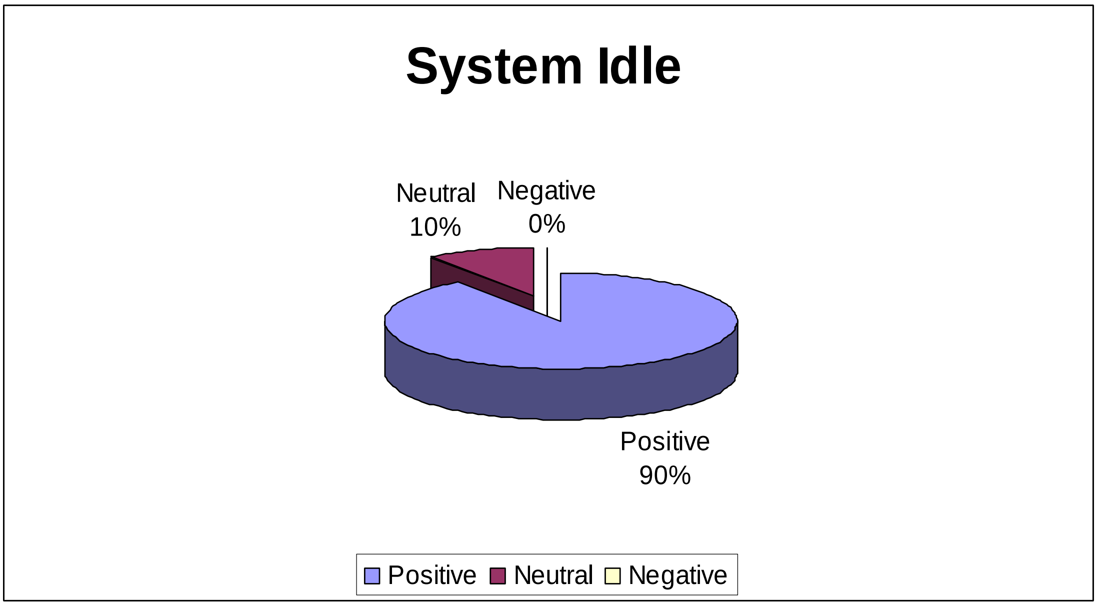
Q1) How would you rate the speed that MS WordPad opened up ready for use?

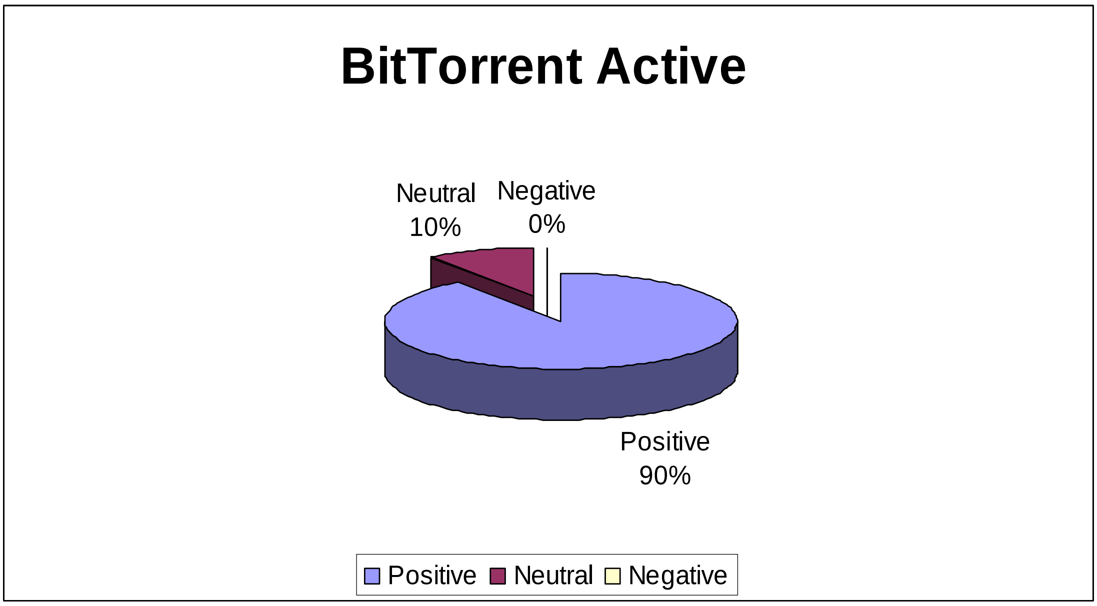




This would seem to indicate that users were unable to notice any difference in the performance between the two conditions. In one case one subject actually thought BitTorrent increased the system performance.

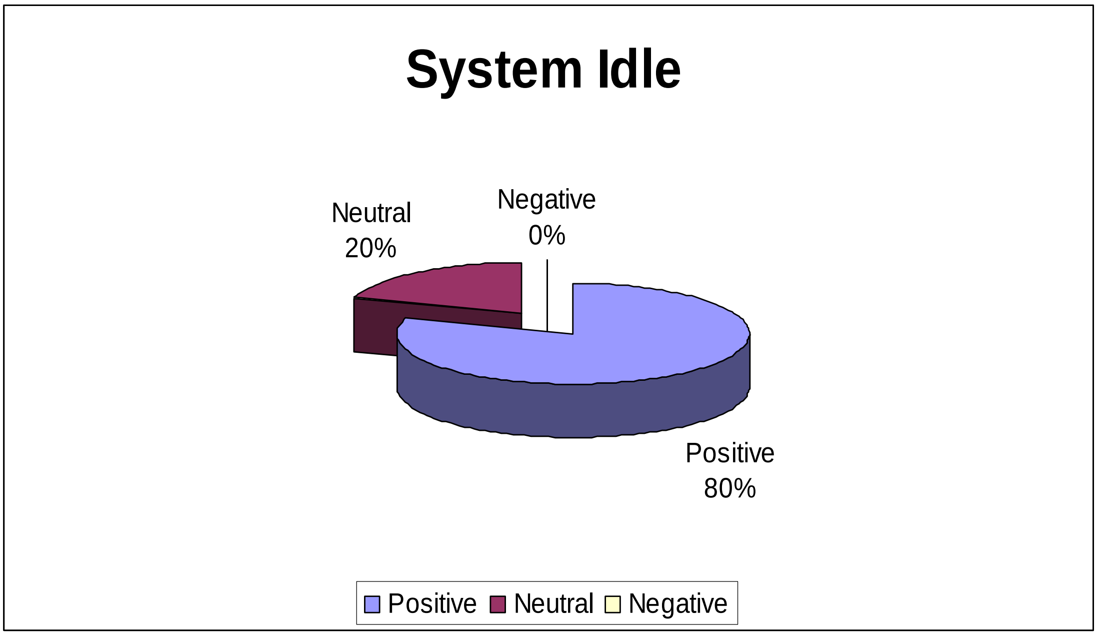
Q2) When typing the text did there appear to be any visible lag?

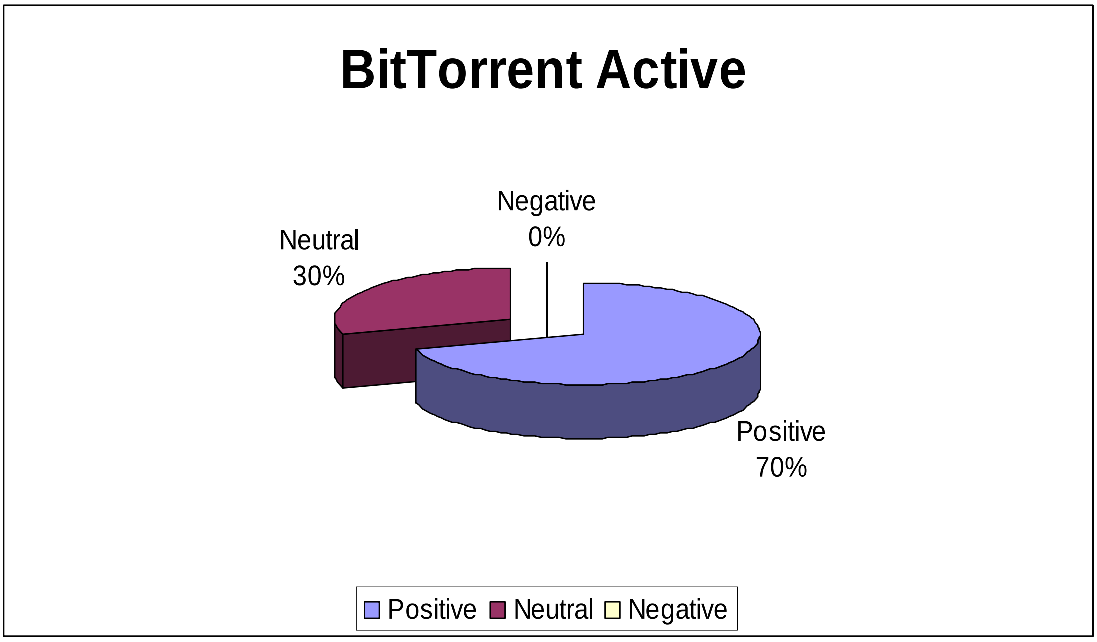




Again, with identical results, we have evidence that users are unable to notice any difference in the machines performance.

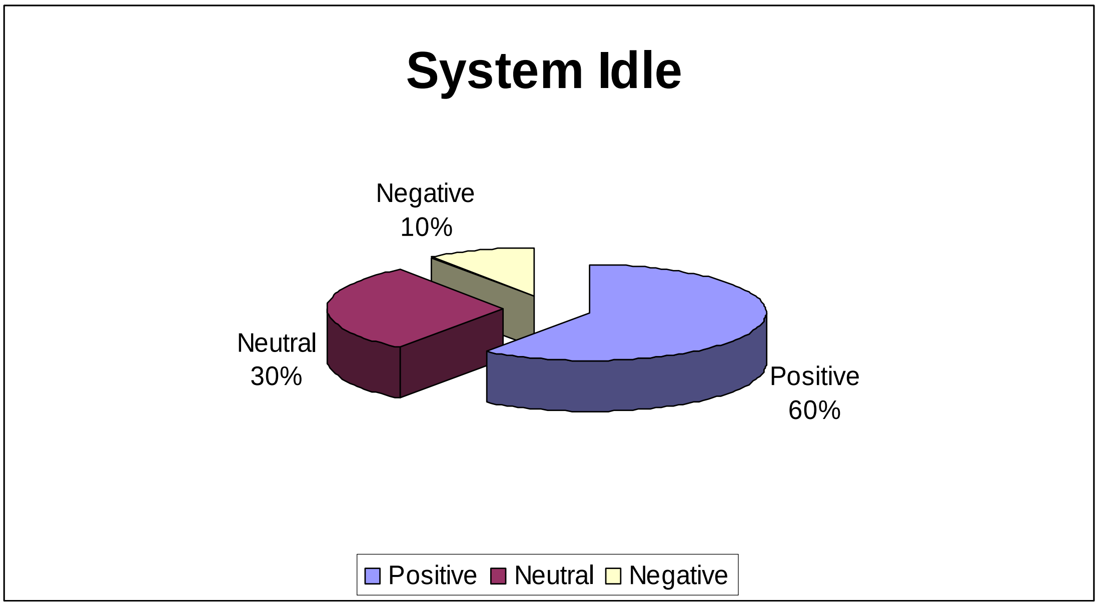
Q3) How would you rate the speed that MS WordPad saved the file?

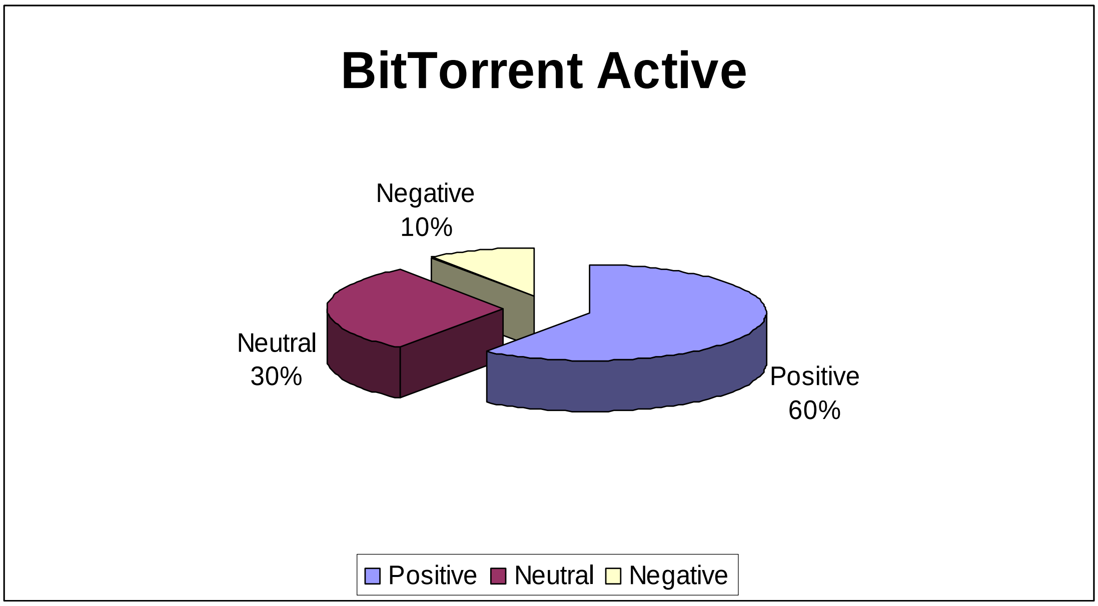




Here we have a very slight difference. 10% of subjects felt the save time was slightly too long. This could be due to BitTorrent reducing the file system performance, but with only 10% change the difference in the result is still inconclusive. User error could have accounted for this. A far more significant difference in results would be required to determine if there was a performance decrease or not.

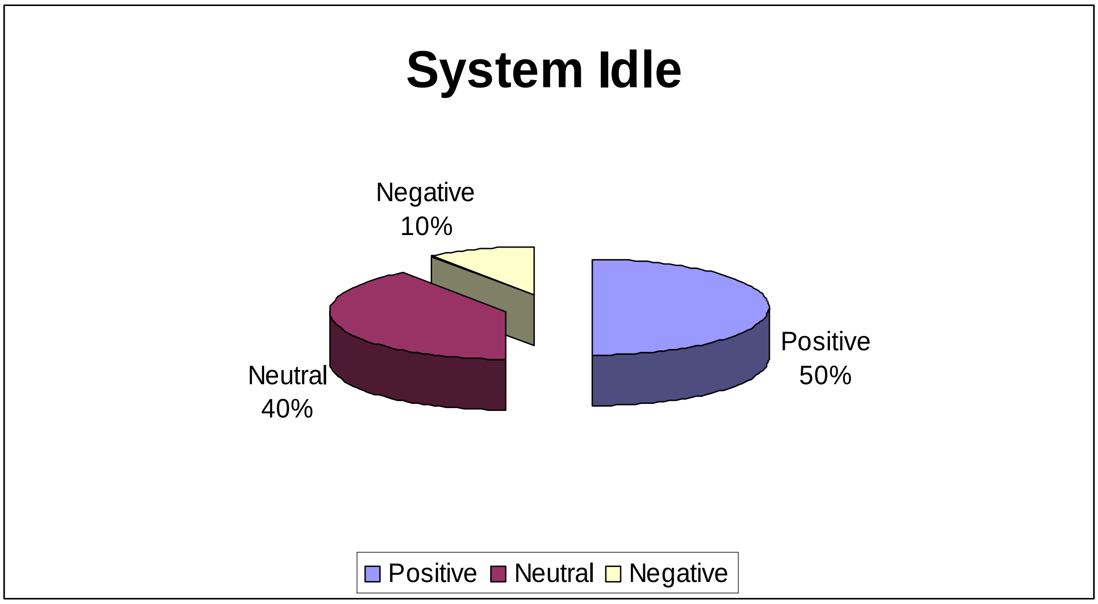
Q4) How would you rate the speed that Internet Explorer opened up ready for use?

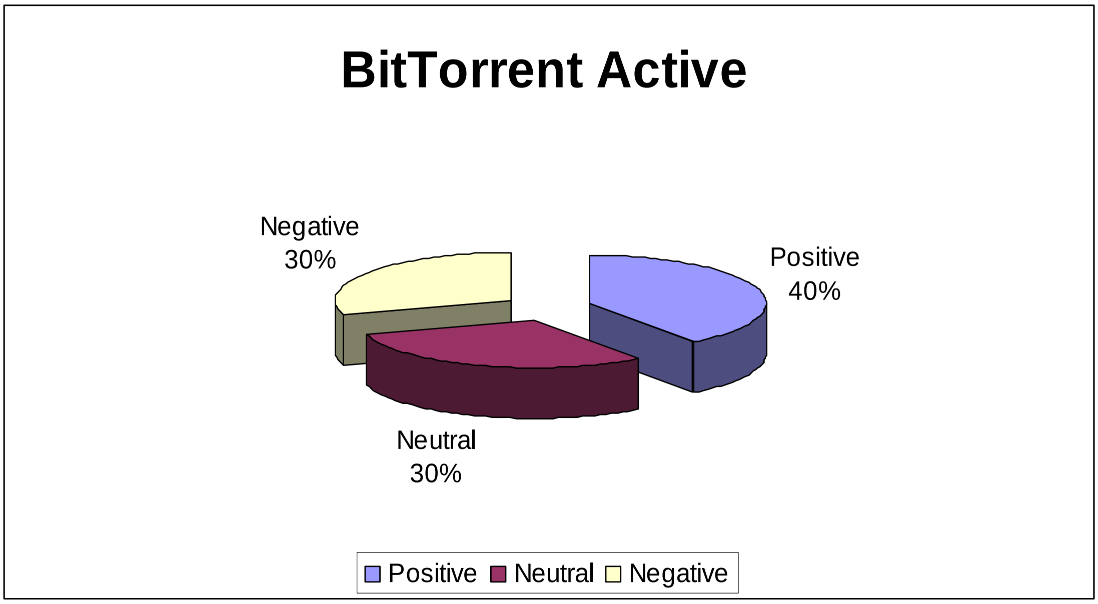




Identical results were received for this activity under both test conditions, showing that once again, users were unable to differentiate between the system being on or off.

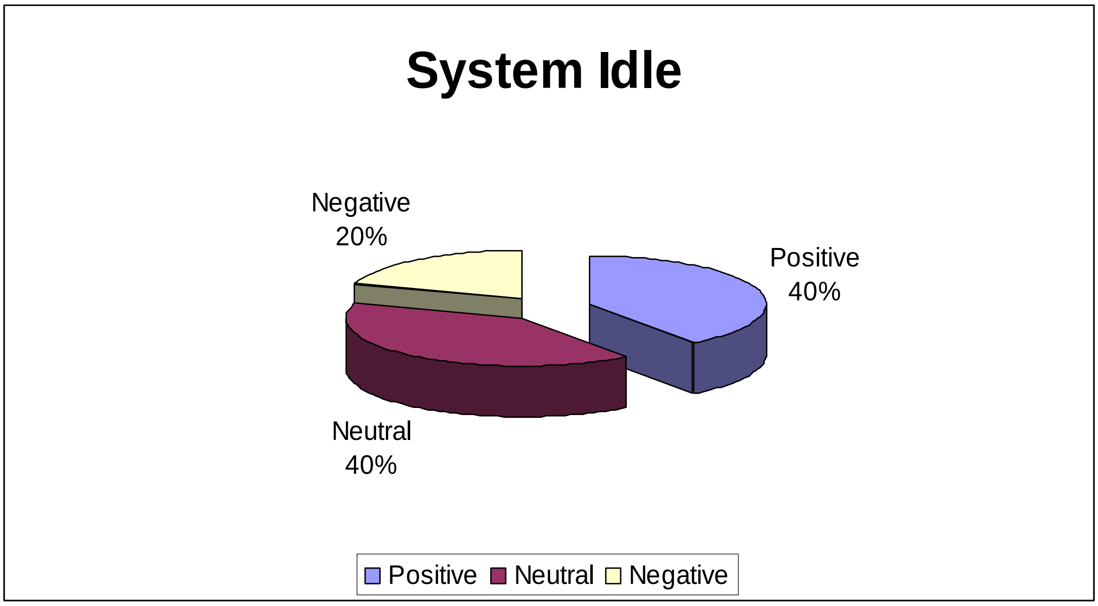
Q5) After typing in the address and pressing return, from the choices below please indicate what best describes how you felt about the time taken for the page to load.

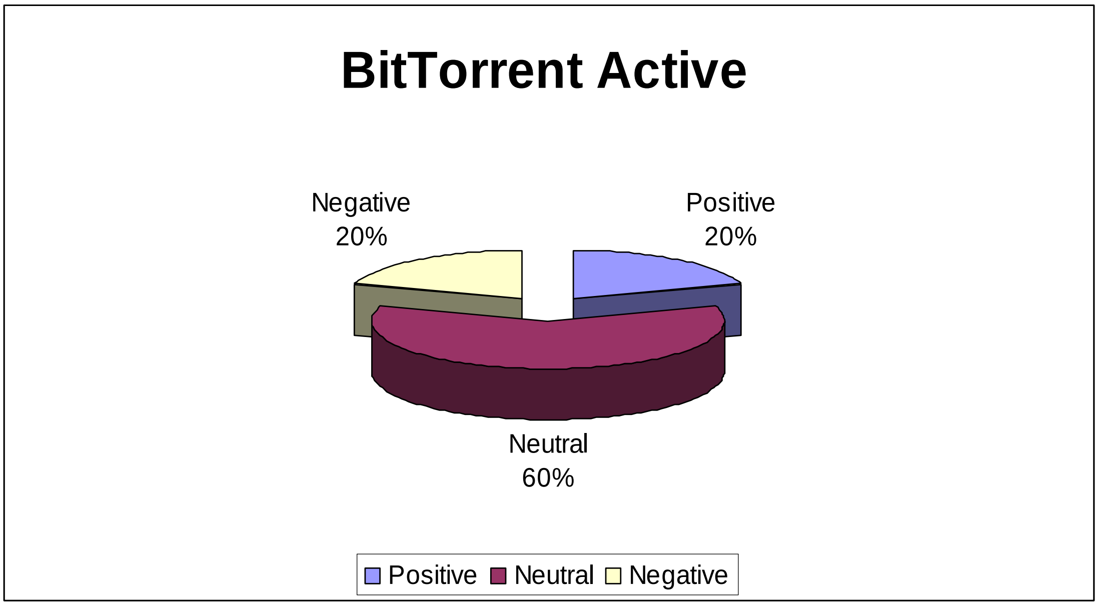




Here we see a significant difference in performance- 30% of subjects now claim that it takes a long time for the test website to be loaded in Internet Explorer and 10% more subjects provide a neutral answer. This problem could be caused by BitTorrent using up the systems available bandwidth on the local machine, or it could be caused by the Seed/File server (on which the website is hosted) having its bandwidth exhausted due to uploading data to all peers. Either way it would seem to indicate a performance difference while BitTorrent is in use. However, still less than 50% of subjects provided a negative answer.

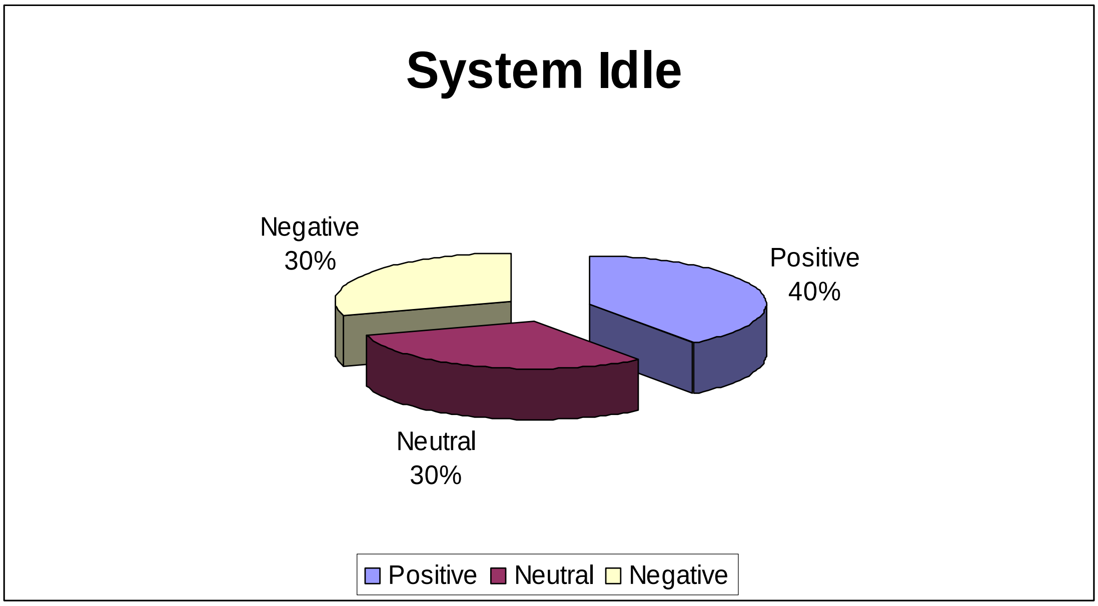
Q6) Browsing the webpage, clicking on each link on the left hand pane and reading the message on each page, how would you rate the speed of each page loading?

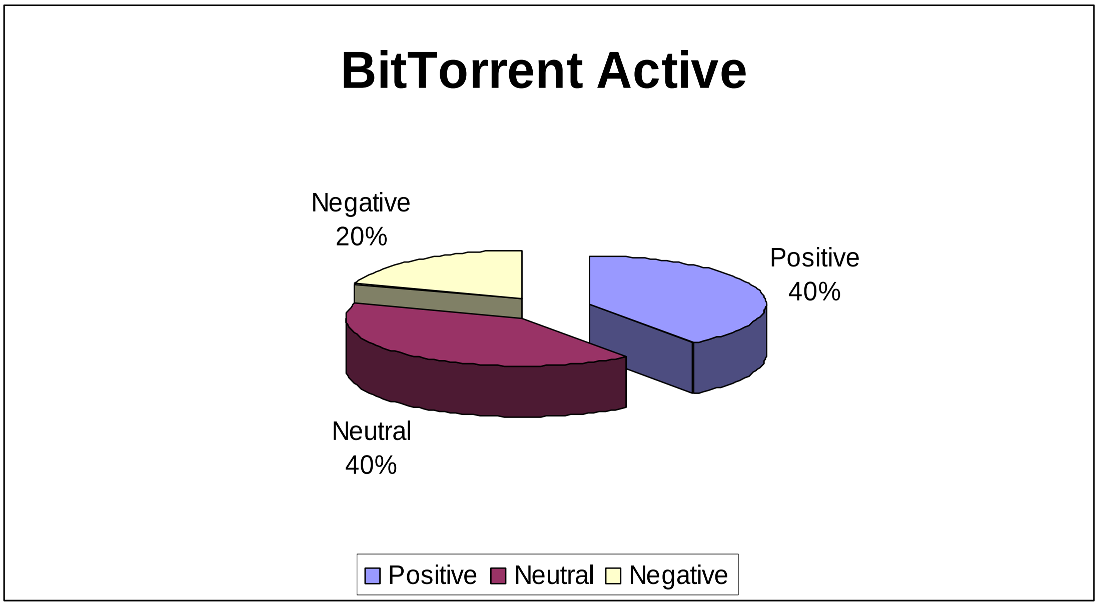




Here we see a significant difference in the results. 20% more subjects choose the neutral option of "acceptable". Although this indicates a decrease in performance it is not substantial enough for the subjects to choose the negative option.

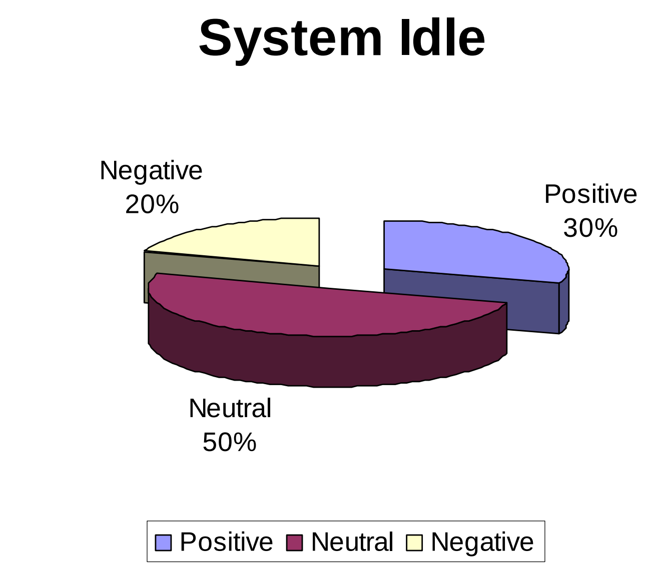
Q7) Browsing the "my network places" how would you rate the time taken from clicking "My Network Places" to the shared folders becoming visible?

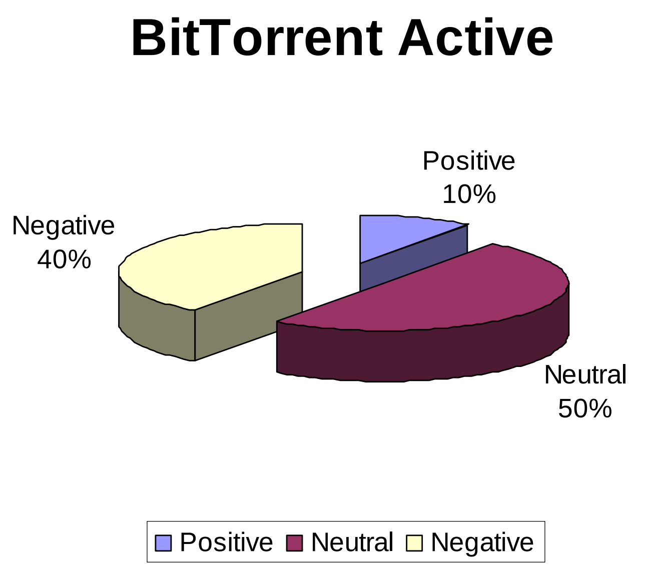




Again, in this activity we see very similar results, leaving us unable to provide a definitive argument as to whether there is a performance difference or not.

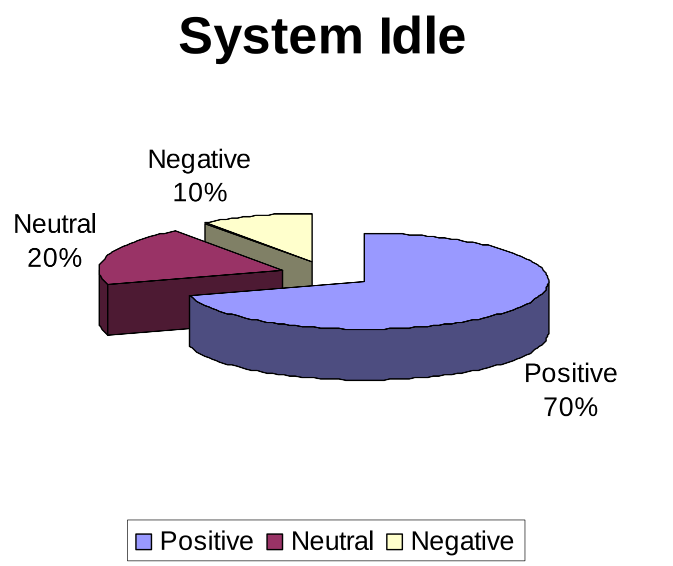
Q8) After you right clicked and pasted the file, how would you describe the time taken for the file to be transferred to the shared folder?

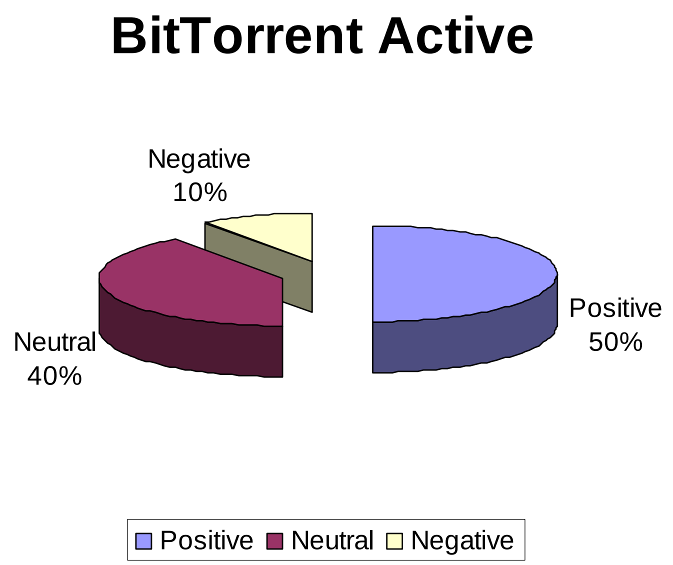




Here we see a large difference in results. Now 40% of subjects find that it is unacceptably slow to transfer files across the network, compared to 20% when the system is in an idle state. This shows a 30% increase. Interestingly, the amount of subjects who provided a neutral answer of “acceptable” remained the same.

Q9) How would you describe the time taken from the game loading until the game was ready for play?

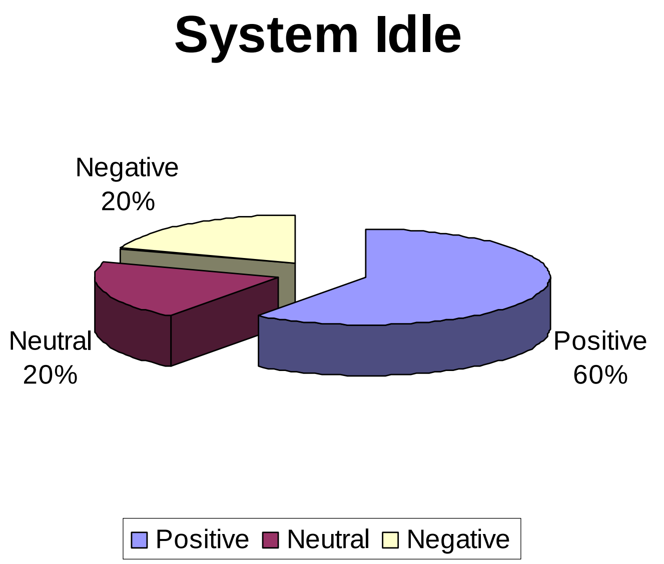


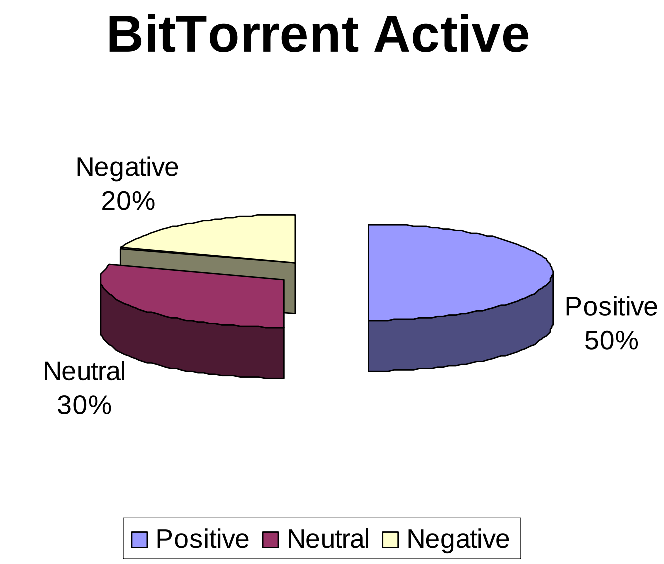


Here we can see that 70% of subjects were happy with the load speed of the game compared to only 50% when BitTorrent was active, however, the 20% changed their answer to the neutral answer of "acceptable" indicating they still did not feel the performance of the system was sufficiently diminished to choose the negative answer.

Q10) When playing the game, how would you describe the game frame-rate? (Was it

smooth and consistent?)





Again, we see a difference in results, but it is not significant enough to say there is a definite noticeable performance decrease. The majority of subjects fail to notice the difference under both test conditions.

After reviewing the results I believe that on the whole they support my hypothesis that the performance penalty is not sufficient to cause the system to appear slow or unusable, although there are more significant results in certain tests than in others (the game load time being the best example) there is still not sufficient grounds to say that users felt the performance was diminished on account of using the BitTorrent program. What matters overall is the average performance. This is where the “System Idle – results average” and “BitTorrent Active – results average” diagram shown in the results section are significant. We can tell by examining these two pie charts side by side, that overall results were largely similar, indicating that BitTorrent does not have a sufficient enough impact on system performance to cause it to degrade performance to noticeable levels. Overall, the result would seem to support the findings which Donald reached in his study “System Performance impact of Peer-to-Peer file sharing“(Donald, 2004) that the peer resource overhead of P2P network system is relatively efficient.

7.0 Project Critique

In terms of meeting the original project aims, I believe the study was a success as a definitive answer was reached for each question. We now know that it is both faster and, under certain circumstances, more efficient to transfer data from multiple sources in a swarming manner than by traditional client/server means.

We also know that although peers on a P2P network receive a small penalty to performance that it is not sufficient to affect the usability of the workstations in the majority of circumstances.

There were however, a few problems which emerged during the course of my research. These were as follows:

- The “Speed and efficiency test – transfer time” diagram makes it appear that BitTorrent is a slower method of distributing the file to peers on the same LAN segment when this is not really the case. In hind-sight there should have been more than one peer on the HQ side of the network, this would have shown more equal data transfer totals in the “Speed and efficiency test – total data transferred” diagram.

- SisSoft Sandra “Network/LAN benchmark” was unable to detect any difference in performance between when BitTorrent was activated and when it was not. This was possibly because the amount of bandwidth usage was so small. Another benchmark could have been run that would have been able to produce more detailed results.

- Although I had in advance asked for the help from many more subjects than were needed for the “performance impact test”, less than 50% were able to make the test on the day. This meant that the testing network had to be set up again for testing another day. I will keep this in mind when organising tests, in the future and organise more subjects, or guarantee of attendance before testing, perhaps by providing incentives.

- Access to equipment was initially a problem. I was unable to obtain access to the equipment I required for a long enough period of time due to the research depending on timing transfer times for files. If I were to do this research again, I would try to book the lab time in advance and secure a whole day. This comes down to organisational skills, and this is an area I need to improve on in future projects.

- Setting deadlines for myself is also important. One particular section of the project was undertaken in haste as I was somewhat pushed for time. In the future I will endeavour to set and adhere to my own deadlines to ensure that tasks are performed.

All of these issues aside, the project has been an enjoyable and rewarding experience that I hope will benefit both myself and others in the future.

8.0 Further Work

The next challenge in developing such a system is solving the two biggest problems which inhibit further development. These issues are namely security and control. Security is critical on any network system, but P2P computing requires drastically different security strategies which are yet to be conceived. Control is also a vital component that inhibits further development, covering issues such as content control, the protection of intellectual property and management of the network.

9.0 Conclusion

In conclusion I would say that this study has been successful. The research set out to determine whether P2P file sharing in enterprise internetworking environments would be feasible. A further objective was to ascertain whether there would be any advantages or disadvantages of such a system. The results of this research provide evidence that there are real advantages from a performance point of view compared to the traditional client/server architectures.

Although the findings of this report are very much in favour of such systems, at the moment there are still some issues of security and control which will require further research. The next step in developing these systems would be to solve these two main problems.

My hope is that this research could be useful to companies with large network systems looking for ways to reduce both bandwidth expenditure and server computational power. Based on my research, I am confident that P2P will play an important role in improving efficiency in large scale company networks.

10.0 References

Journal Papers

- Eythan Adar & Bernardo Huberman, (2000) Freeriding on Gnutella. *First Monday*; Vol. 5, No. 10, October 2000

- Hari Balakrishnan, M. Frans Kaashoek, David Karger, Robert Morris, Ion Stoica (2003) Looking up data in P2P Systems. *Communications of the ACM*, Vol. 46, No. 2, February 2003

- Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong, (2001) Freenet: A Distributed Anonymous Information Storage and Retrieval System. *Lecture Notes in Computer Science,* Vol. 2009, No. 46, 2001

- E. Korpela, D. Werthimer, D. Anderson, J. Cobb, M. Leboisky. (2001) SETI@home-massively distributed computing for SETI. *Computing in Science and Engineering*, Vol. 3, No. 1, January- February 2001 (P 77-83)

- J.A. Pouwelse, P. Garbacki, D.H.J. Epema, H.J. Sips, "A Measurement Study of the BitTorrent Peer-to-Peer File-Sharing System", *technical report PDS*-2004-003, April 2004

- Antony Rowstron & Peter Druschel. (2001) Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. *Lecture Notes In Computer Science,* Vol. 2218, 2001 (P 329– 350)

- Slater, Derek, Smith, Meg, Bambauer, Derek, Gasser, Urs and Palfrey, John G. (2005), Content and Control: Assessing the Impact of Policy Choices on Potential Online Business Models in the Music and Film Industries. *Berkman Publication Series Paper* No. 2005-01.

- Jerome Verbeke, Neelakanth Nadgir, Greg Ruetsch, Ilya Sharapov. Sun Microsystems Inc. (2002), Framework for Peer-to-Peer Distributed Computing in a Heterogeneous, Decentralized Environment. *Lecture Notes In Computer Science,* Vol. 2536, 2002 (P1-12)

- Zhao, B. Y., Kubiatowicz, J. D., and Joseph, A. D. 2001 Tapestry: an Infrastructure for Fault-Tolerant Wide-Area Location and. *Technical Report. UMI*, University of California at Berkeley.

Conference Papers

- Peter Backx, Tim Wauters, Bart Dhoedt, Piet Demeester, (2002) A comparison of peer-to-peer architectures. *Eurescom 2002 Powerful Networks for Profitable Services*; October 21-24, 2002, pp. 215-222.

- Chiranjeeb Buragohain, Divyakant Agrawal, Subhash Suri, (2003) A Game Theoretic Framework for Incentives in P2P Systems. *Third International Conference on Peer-to-Peer Computing* (P2P'03), September 1st, 2003.

- Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau , Nick Lanham, Scott Shenker, (2003) Making Gnutella-like P2P Systems Scalable. *ACM Sigcomm 2003*; August 29th 2003.

- Peter Druschel, Antony Rowstron, "PAST: A Large-Scale, Persistent Peer-to-Peer Storage Utility," *hotos*, p. 0075, *Eighth Workshop on Hot Topics in Operating Systems*, Oberbayern, Germany 2001.

- Richard T. B. Ma Sam C. M. Lee John C. S. Lui David K. Y. Yau (2003) Incentive Resource Distribution in P2P Networks. *IEEE International Conference on Distributed Computing Systems (ICDCS)*, Tokyo, Japan, March 2004.

- Gurmeet Singh Manku (2003) Routing Networks for Distributed Hash Tables. *Annual ACM Symposium on Principles of Distributed Computing Proceedings of the twenty-second annual symposium on Principles of distributed computing*. Boston, Massachusetts 2003 (P 133 – 142)

- Naoum Naoumov and Keith Ross (2005) Exploiting P2P Systems for DDoS Attacks. *P2PIM 2006*; January, 2006.

- J.A. Pouwelse, P. Garbacki, D.H.J. Epema, H.J. Sips, (2004) The Bittorrent P2P File-sharing System: Measurements and Analysis. *Proceedings of the 4th International Workshop on Peer-to-Peer Systems (IPTPS'05)*, Ithaca, New York, Feb 2005.

- Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Shenker (2001) A Scalable Content Addressable Network. *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications,* San Diego, California, United States 2001.

- Stefan Saroiu, P. Krishna Gummadi, Steven D. Gribble, (2001) A Measurement Study of Peer-to-Peer File Sharing Systems. *Proceedings of the SPIE/ACM Conference on Multimedia Computing and Networking (MMCN) 2002,* San Jose, CA, January 2002.

- Kundan Singh & Henning Schulzrinne (2004) Peer-to-Peer Internet Telephony using SIP. *International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV).* Stevenson, Washington, USA 2005:

- Beverly Yang & Hector Garcia-Molina, (2003) Comparing Hybrid Peer-to-Peer Systems *Proceedings of the 27th International Conference on Very Large Databases (VLDB)*, Roma, Italy, September 2001.

- Beverly Yang & Hector Garcia-Molina, (2003) Designing a Super-Peer Network *Proceedings of the 19th International Conference on Data Engineering (ICDE)*, Bangalore, India, March 2003.

Book References

- Nelson Minar and Marc Hedlund (2001) *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, 2001) (Andy Oram) O'Reilly Media, Inc., UK

- Clay Shirky, Kelly Truelove, Rael Dornfest & Lucas Gonze (2001) *2001 P2P Networking Overview - The Emergent P2P Platform of Presence, Identity, and Edge Resources*, 2001) O'Reilly Media, Inc., UK

Web References

- Julian Bond, (2001) *Business uses of Peer to Peer (P2P) technologies* (White Paper) http://www.voidstar.com/downloads/PB2B.pdf Last viewed 11 December 2005

- Bram Cohen (2003) *Incentives build robustness in BitTorrent.* http://www.bittorrent.org/bittorrentecon.pdf Last viewed 04 May 2006

- C Matthew Curtin, (2004) *Spying on Spyware. Interhack Publications* (White Paper) http://web.interhack.com/publications/spyware.pdf Last viewed 23 April 2006

- James Donald (2004) *System Performance impact of Peer-to-Peer file sharing*. http://www.princeton.edu/~jdonald/taiwan/p2p_performance.pdf Last viewed 14th December 2005

- Lionshare White Paper (2004) *Connecting and Extending Peer-to-Peer Networks* http://lionshare.its.psu.edu/main/info/docspresntation/LionshareWP.pdf Last viewed 13th December 2005

- MetaComputing *Peer-to-Peer Architecture Proposal Legion*. (2001) An Integrated Architecture for Secure Resource Sharing. (White paper)

- Steve Smith (July 2003) *P2P in B2B: Getting past the "N" Word.* http://www.streamingmedia.com/r/printerfriendly.asp?id=8473 Last viewed 16th November 2005

- John Thornton, (2001) *Kazaa the virus desktop* http://www.theregister.co.uk/2001/10/09/kazaa_the_virus_desktop/ Last viewed 11 December 2005

- K. Truelove (2000) *To the bandwidth barrier and beyond.* http://web.archive.org/web/20010202143400/http://dss.clip2.com/gnutella.html Last viewed 8th December 2005

- Brandon Wiley (2003) *Distributed Hash Tables, Part I.* http://linuxjournal.com/article/6797 Last viewed 11 December 2005

Video/News/Magazine Articles

- Morris Edwards (2001) *Startups vie for P2P application niches.* CNN Money. http://onionnetworks.com/press/CNNMoney_112101.pdf Last viewed 16th November 2005

- Shiuhpyng W. Shieh & Dan S.Wallach (November 2005) *Ad Hoc and P2P Security*. IEEE Internet computing magazine. http://csdl2.computer.org/comp/mags/ic/2005/06/w6014.pdf Last viewed 12th December 2005

11.0 Appendices

1. Test Network Diagram
2. Speed and efficiency test log (unfilled)
3. Resource usage test log (unfilled)
4. Workstation usability test log (unfilled)
5. Speed and efficiency test log results (raw data)
6. Resource usage test log (raw data)