Unlocking the South West's Creative Archive

Deliverable B:

SWCA - Institution Based Research

FluffyLogic August 2006

Deliverable B:
SWCA - Institution Based Research

Introduction

The South West Creative Archive (working title: SWCA) is a proposed system for easy distribution of large files using peer-to-peer technology (p2p). A peer-to-peer system is a method of exchanging data that combines the networked, distributive power of all participating individual computers, enabling an interconnected mesh of network bandwidth.

This report builds on the conclusions set out in 'Deliverable A: SWCA - Research & Evaluation'.

Network Features

A number of network-related issues have been discussed in the recent round of meetings. It is recognised as vital that all institutions participating in the SWCA must be confident of their networks' security and integrity. Discussions around this issue have been fruitful, highlighting a number of areas for further examination and suggested methods for addressing these.

Network Security - Many p2p (Peer-to-Peer, also known as File Share or File Trade[1]) applications exist outside the mainstream and are used to trade copyrighted material. This has given such applications a negative connotation with regard to computer security issues. We note that such applications often require the opening of specific IP ports that leave a network less secure than before.[2] Also, there are a number of p2p applications that allow malicious software such as a 'trojan' to 'piggy-back' on the p2p application and so gain access to a computer system.[3]

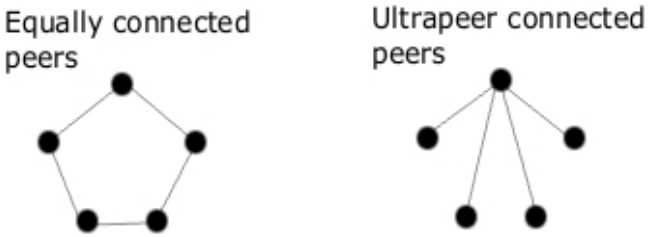
Network Load – All networks have capacity limitations. Concerns were raised regarding the possibility that, should a number of such applications be installed across an institution's network, their operation would consume a disproportionate amount of network bandwidth and so impact on other users. As an example how this can happen, recent research using the Voice over Internet Protocol (VoIP[4]) software Skype showed how this dynamic routing of traffic through nodes can escalate even if the installed client is not active.[5] This happens because the other users of the Skype network may be routed through this installed client to aid the traffic flow, thus resulting in the user becoming a 'super-peer' facilitating many other users.[6]

File Access - Files within many institutions (e.g. University of Bristol) are often stored on a shared network. This means that the files may only be accessed by the individual users whose files they are, or by a user with administration level access.

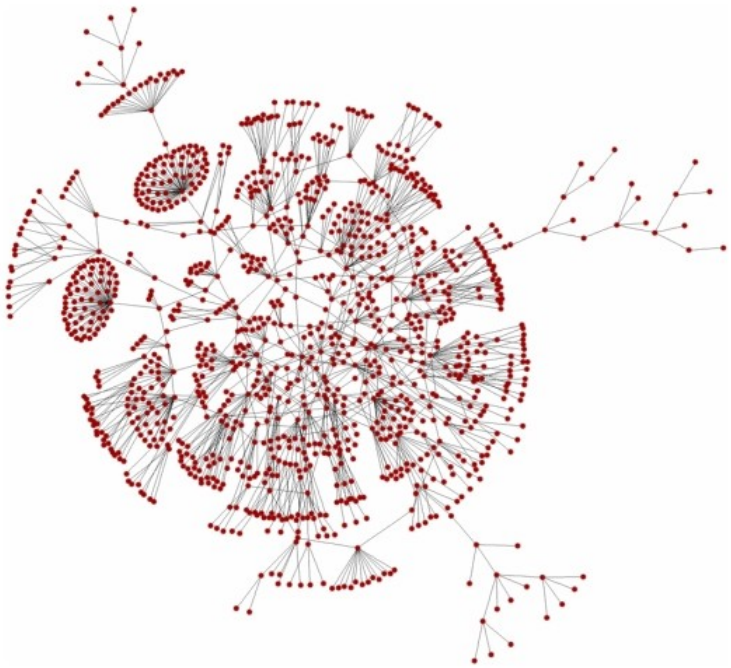
1 http://en.wikipedia.org/wiki/P2p
2 http://searchsecurity.techtarget.com/tip/1,289483,sid14_gci929175,00.html
3 http://en.wikipedia.org/wiki/Trojan_horse_%28computing%29
4 http://en.wikipedia.org/wiki/Voice_over_IP
5 http://www.ja.net/development/voip/skype&janet.pdf
6 Openp2p.net defines a superpeer as a participant (node) in the network that has the same attributes as other nodes but more of them that becomes a 'hotspot' of network activity. See http://www.openp2p.com/pub/gl/58

The Co-peer Application

All of the above issues need to be addressed. We propose addressing the above through the use of an application that employs a methodology combining some of the attributes of an ultrapeer along with new security-conscious controls, and some of the attributes of a typical network.[7] Before we explain the proposed application, it is worth considering the context to this approach: p2p networks operate by connecting users (peers) together. An 'ultrapeer' is the name given when one of the peers in a network takes on more connections than others. They then become the centre point of a number of peers, hence the term 'ultrapeer'. In the image below there are two groups of five peers, in the left image they are connected equally with two connections per peer. In the right image they are connected using the ultrapeer method, with one peer connecting to all via four connections.



Ultrapeer networks then cross-link the ultrapeers to produce a larger network. This can be described as a hybrid decentralised network.[8] This method of connection brings the advantages of faster interconnections between the peers, plus optimal scalability to allow the entire network to grow/shrink easily. This method of peer-connection is used by a number of p2p applications such as Gnutella and EMule.[9] This is how such a network might look (*image from: iaSee graph of a Gnucleus ultrapeer network*[10]):



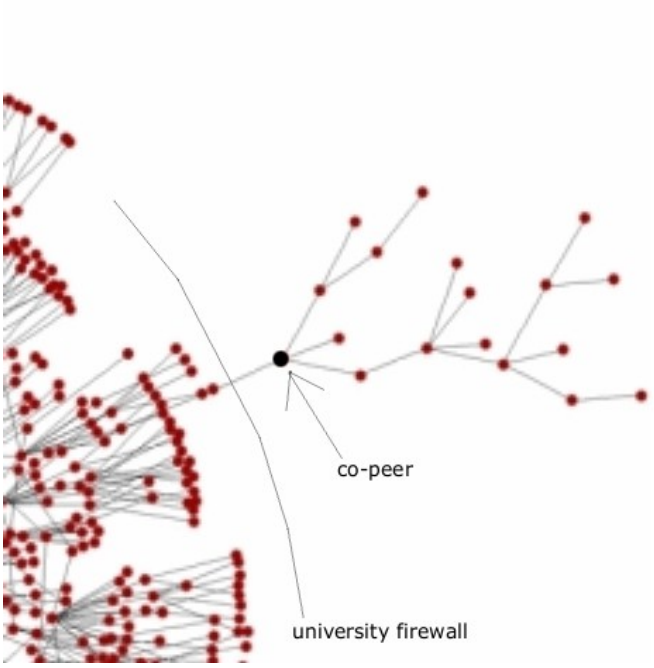
7 It appears that the term 'ultrapeer' is often interchangeable with 'supernode' or 'superpeer', for example see http://www.p2pnet.net/story/5677

8 http://www.p2pnet.net/story/5677

9 http://www.sysdesign.ca/archive/berkes_gnutella_freenet.pdf

10 http://www.aisee.com/graph_of_the_month/gnumap.htm

We propose to use a variation of this ultrapeer method – of allocating a single peer to become the connection point for a number of other peers. This means that users from within an institution connecting to the p2p network do so via a new ultrapeer, or co-peer (as in 'co-ordinating peer'). This would result in a network that looked something like the image below (*in reference the iaSee graph of a Gnucleus ultrapeer network used above*):



The co-peer method, unlike the existing ultrapeer method, will have a number of attributes that other peers don't have:

- ***Gateway*** – It will act as the 'gateway' between the institution's systems and the wider p2p network. This means if a network administrator, for whatever reason, needs to limit or control the p2p traffic it can be done via the co-peer.

- ***Administrator Access*** – The co-peer would have a degree of administrator access to the system to allow him/ her to deal with files that users have published, that are on the university network.

- ***Throttle*** – It is normal for p2p users to be able to control the amount of network bandwidth they allow the application to use. This individual control will remain, but each peer will only be able to access resources from a maximum pool set by the co-peer. In effect the co-peer will be allocated bandwidth which can then be divided amongst him/ her peers. This will ensure that the p2p's bandwidth within the institution does not exceed acceptable limits.

- **Port Control** – The co-peer controls the port used by this application, allowing the institution to select their preferred port for traffic. This would then follow for other peers (note: this dissemination of port number may alter depending on how the final application deals with peers, see point below).

- **Update Control** – The co-peer can be used as a central point to deal with updating other peers with the latest software patches (note: this dissemination of port number may alter depending on how the final application deals with peers, see point below).

- **Thin Client Method**[11] – This refers to systems where there is a central point that all users connect to, and where users are akin to subsidiaries of the central system. It may be that the best method to deal with peers is not to install any (or any full) applications to each user, but the users (using a web-interface such as Azureus HTML WebUI[12]) connect to the co-peer which does all the work. This would mean the application is effectively a pseudo-p2p system within an institution's network, and a full node of a p2p to outside users. The degree of balance between peer and co-peer is something that is still under consideration.

Attributes of the Co-peer

We are looking at the following technologies and issues in the development of this co-peer system:

- **Azureus**[13] – This application will form the base of the system. This was discussed in the first research report in full detail. The University of Bristol uses Java 1.4/1.5 on all its machines. The current version of Azureus (2.4) uses Java version 1.5
- **LAN Peer** – We would look to have the individual peers use the Azureus LAN Peer plugin to optimize file searching and downloading. This application searches the Local Area Network (LAN) for a peer with the data you need and if such peers are found, it automatically downloads from this peer via LAN, usually resulting in a much higher download rate.[14]
- **MSI** – The University IT people requested that the application be wrapped up as an MSI (Microsoft Installer[15]).
- Any authentication to be encrypted.

Trackers

The issues of operating trackers is another important point as to how the SWCA will operate. It is important here to clarify what a tracker is and how bittorrent works.[16] Bittorrent essentially creates a mini 'network' when it connects to other torrent applications, then breaks files down into smaller parts for ease of passage. When you request a file, it begins by downloading and running the torrent file (a small text file ending .torrent) which contains the outline information about the file you want (i.e what it is; film, music, etc and how big it is). This torrent file also points you towards another file running on a server called a 'tracker'.[17]

The trackers keep an account of communication between those attempting to download the files referred to in the torrent. Azureus (the client software we have chosen to base the project on) has two main methods of tracking: one is its own 'Azureus Embedded Tracker' system - this is a form of 'torrent-less' tracker,[18] and means that each Azureus client running can act as a tracker.[19] This makes it easier to create torrents (as you are less dependent on a central server and/or there is no need to transfer files to a central server), however it can then only reliably be picked up by other Azureus clients, and when the computer is online. The other method is to

11 http://en.wikipedia.org/wiki/Thin_client
12 http://azureus.sourceforge.net/plugin_details.php?plugin=azhtmlwebui
13 http://www.azureuswiki.com/index.php/Main_Page and http://azureus.sourceforge.net/
14 http://www.azureuswiki.com/index.php/UG_Plugins#LAN_Peer_Finder
15 http://en.wikipedia.org/wiki/Microsoft_Installer
16 http://en.wikipedia.org/wiki/Bittorent#How_BitTorrent_works
17 http://en.wikipedia.org/wiki/BitTorrent_tracker
18 http://azureus.aelitis.com/wiki/index.php/PublishingFiles
19 http://azureus.sourceforge.net/faq.php#17

use a separate tracker. For this to work, the contributing user would need a static IP address if the torrent is to be of any use.[20]

If the tracker server and/ or peer client systems can copy and seed the torrent, then this has the advantage that the host can turn off their computer and the torrent is still available. This leads to the important are around how the SWCA system could look to copy the functionality of a system, by creating an apparently static IP address to aid publishing or to go with the embedded tracker.[21] It is possible that we can work towards a system that has the advantages of both, possibly using the Multi Tracker system, which is an unofficial feature of bit torrent, and is available as a plugin for Azureus. This is also dependent on the method we use for the overall system (see 'Thin Client Method', above).[22]

Managing SWCA Content

The issue of the content of the project has to some extent been kept separate from the technical discussions thus far. However, there is one area where the two issues meet, and that requires consideration here: if SWCA users publish files in a format that other users cannot access, then a large portion of the power this project may unlock is lost. This means that files published by SWCA will need to observe the following:

- Technical Standards for Media Files
- License Information
- Content Information

Technical Standards for Media Files

Our experiments in downloading files from existing torrent networks have pointed to a lack of clear information about the technical needs and content of a file. It is vital that users know both what is in a file and how they can use it. To this end, we suggest that the project engages with current discussions around file format for most types of file. While it is not envisioned that the standards set out will be binding, they will help users, and help to ensure the success of the project.

This discussion will focus on film content as it is the largest file type we expect to encounter at this stage, and the tests to date are based in the Drama: Theatre, Film, Television department of the University of Bristol.

It is important to clarify the terminology at the outset. A 'player' is the software used to view/hear a media file. The file format (such as .avi or .mov) is the 'wrapping' a media files comes in. A codec (short for COmpressor-DECompressor) is the compression technology applied to the media file. All three are important in this discussion.[23]

Our research shows that the University of Bristol technical support for players is at its optimum with Windows Media Player, though there is some support for QuickTime and Real. In the Drama: Theatre, Film, Television department (where the system is to be tested) users tend to make film files as QuickTime or MPEG, as files often come from Macs/Avid. More discussions are needed to agree upon the recommended media player. It is also worth considering VLC Media Player – a free player that does both

20 http://azureus.aelitis.com/wiki/index.php/HostAndShareYourTorrents#Setting_Up_Your_Tracker
http://azureus.aelitis.com/wiki/index.php/HostAndShareYourTorrents#Setting_Up_Your_Tracker

21 Such as using http://www.dyndns.com/

22 http://en.wikipedia.org/wiki/BitTorrent#Multitracker

23 http://www.plugincinema.com/plugin/about_us/plugin_book.htm see glossary.

.avi and .mp4 formats.[24]
Windows Media Player can play files encoded as MPEG, but not if they are created as MPEG4 files (.mp4 – though there is a plugin to allow this).[25] QuickTime can play .mp4 files. As with Windows Media Player, Real requires a plugin.[26] Our experience with existing torrent networks shows that the most common form of file is as a .avi format. More discussions are needed to agree upon recommended file format/s.

In terms of compression, our research shows that existing torrent networks use the DivX or Xvid codecs.[27] Xvid is an open-source codec that many consider to be the best balance of compression and quality.[28] However currently the Xvid codec does not enjoy mainstream support, so some editing software, such as Premiere, is not compatable with files encoded with Xvid. More discussions are needed to agree upon recommended codec/s.

License Information
All media files should carry licence information. This tells the user what they can and cannot do with the file. It is envisioned that services such as CreativeCommons.org can be used to create the document that would accompany each file, but this is something for future consideration.[29] Technically, this information would be either displayed on the content catalogue website and/ or embedded in the file in question.

Content Information
It has been suggested that the content catalogue use the RDF syntax for the descriptions on files.[30] This would allow the system to better integrate with other academic projects around data usage. This issue needs more feedback from University of Bristol.

Handling Less Popular Files

As has been discussed before, torrent systems are adept at dealing with large files that have a large demand – turning the demand into 'seeds' from which the files are transferred. However, if a file is not popular, torrent systems may be less efficient at passing it to other users. The idea of 'super-seeding' has been raised to deal with this issue.[31] The BMEX system would host a server that took copies of all new SWCA material published, resulting in a form of 'super-seeding' that would aid the progress of less popular files.[32]

24 http://www.videolan.org/vlc/
25 http://support.microsoft.com/kb/316992/
26 http://jogin.com/weblog/archives/2004/02/29/real_obnoxious
http://www.apple.com/quicktime/technologies/mpeg4/
27 http://www.divx.com and http://www.xvid.org/
28 http://www.doom9.org/index.html?/codecs-final-105-4.htm
29 http://creativecommons.org/
30 http://www.w3.org/RDF/ and http://www.xml.com/pub/a/98/06/rdf.html
31 http://en.wikipedia.org/wiki/Bittorent also more on http://en.wikipedia.org/wiki/Super-seeding
32 http://www.joltid.com/index.php/peercache/