

Analysis and Comparison of Peercasting Techniques and Implementations

Tampere University of Technology

Lassi Väättäimöinen

Marko Saukko

Modified: 06/03/2007

Version 15

Version History

| <i>Date</i> | <i>Version number</i> | <i>Change</i> | <i>Initials</i> |
|-------------|-----------------------|---|-----------------|
| 30.11.2006 | 1 | Initial version | MS |
| 10.12.2006 | 2 | Updated chapters 3 to 6 | MS |
| 17.12.2006 | 3 | " | MS |
| 31.12.2006 | 4 | Updated chapters 4 and 6 | MS |
| 2.1.2007 | 4_1 | Linguistic corrections to 1-4 | LV |
| 3.1.2007 | 4_2 | Added peercasting theory + tribler stuff | LV |
| 4.1.2007 | 5 | Added correction + RawFlow stuff | MS, LV |
| 6.1.2007 | 5_1 | Minor changes | MS |
| 7.1.2007 | 6 | Updated chapters 5 to 6 | MS |
| 9.1.2007 | 7 | Added some peercast theory | LV |
| 14.1.2007 | 8 | Modified references | MS |
| 15.1.2007 | 8_1 | Server notes added. Theory updated. Corrections | LV |
| 16.1.2007 | 9 | Corrections | MS, LV |
| 18.1.2007 | 10 | Additions to comparison chapter | MS |
| 19.1.2007 | 11 | Corrections | MS, LV |
| 21.1.2007 | 12 | Rawflow, Joost, Sop: Notes to text. | LV |
| 22.1.2007 | 13 | Opened notes | MS |
| 23.1.2007 | 14 | Corrections | MS, LV |
| 24.1.2007 | 15 | Finalizing the document | MS, LV |

Abstract

As amount of media delivered in the Internet seems to be ever-growing and speeds of end-users' access network connections are getting faster by the day, network capacity keeps getting growingly scarce resource, especially when streaming data is concerned.

Different kinds of solutions have been proposed and researched to overcome this problem. Recently, peer-to-peer approach has gained noticeable interest, both corporate and academic, in resolving the problems in massive distribution of streaming data. As we have seen during the project, no comprehensive research results on effectiveness of these techniques have been introduced, as the field seems to be heavily corporate-driven, so there are a lot of unresolved questions for future work.

The resulting documentary of this project serves as an limited overview of the current situation in peercasting: which products are ready for mass marketing, are these usable? Does anybody have the silver bullet? This document serves as an overview on the current status of peercasting technologies, software and service providers.

Table of contents

| | |
|--|-----------|
| 1 Introduction | 1 |
| 2 Background theory..... | 3 |
| 2.1 Unicast..... | 3 |
| 2.2 Multicast..... | 5 |
| 2.3 Peer-to-peer..... | 7 |
| 2.4 Peercasting..... | 9 |
| 2.4.1 Basic idea of peercasting..... | 9 |
| 2.4.2 Push and pull methods..... | 11 |
| 2.4.3 Multiple-stream approach..... | 12 |
| 3 Peercasting techniques, software and service providers..... | 14 |
| 3.1 SopCast..... | 14 |
| 3.2 Octoshape..... | 16 |
| 3.3 TVU networks..... | 17 |
| 3.4 RawFlow..... | 17 |
| 3.5 PeerCast..... | 18 |
| 3.6 Tribler Streaming..... | 19 |
| 3.7 JoostTM (The Venice ProjectTM)..... | 20 |
| 3.8 Software not included in comparison | 21 |
| 4 Reference techniques..... | 23 |
| 4.1 YouTube..... | 23 |
| 4.2 Bubblare..... | 24 |
| 4.3 Veoh..... | 24 |
| 4.4 Zudeo (Azureus 3.0)..... | 26 |
| 5 Protocol/Program comparison..... | 27 |
| 6 Testing the software..... | 36 |
| 6.1 Testing environments..... | 36 |
| 6.2 Clients..... | 36 |
| 6.2.1 SopCast..... | 36 |
| 6.2.2 PeerCast..... | 38 |
| 6.2.3 Octoshape..... | 39 |
| 6.2.4 TVU networks..... | 42 |
| 6.2.5 RawFlow..... | 43 |
| 6.2.6 Tribler streaming..... | 44 |
| 6.2.7 JoostTM (The Venice ProjectTM)..... | 45 |
| 6.3 Server (broadcasting) software..... | 47 |
| 6.3.1 SopServer..... | 47 |
| 6.3.2 RawFlow..... | 49 |
| 6.3.3 PeerCast..... | 50 |
| 6.3.4 TVUNetworks..... | 52 |
| 6.3.5 Octoshape..... | 53 |

| | |
|--------------------------------------|------------------|
| <u>6.3.6 Tribler Streaming.....</u> | <u>53</u> |
| <u>6.4 Reference techniques.....</u> | <u>53</u> |
| <u>6.4.1 YouTube.....</u> | <u>54</u> |
| <u>6.4.2 Bubblare.....</u> | <u>54</u> |
| <u>6.4.3 Veoh.....</u> | <u>55</u> |
| <u>6.4.4 Zudeo.....</u> | <u>55</u> |
| <u>7 Conclusions.....</u> | <u>56</u> |

List of Acronyms

| | |
|------|--|
| ASF | Advanced System Format, video format developed by Microsoft for streaming video over Internet |
| EULA | End User License Agreement |
| FAQ | Frequently Asked Questions |
| GPL | GNU General Public License |
| GUI | Graphical User Interface |
| ICD | Intelligent Content Distribution. Delivery mechanism developed by RawFlow. |
| IE | Internet Explorer is an Internet browser that is installed by default in Windows operating systems |
| IP | Internet Protocol |
| IRC | Internet Relay Chat |
| LGPL | GNU Lesser General Public License |
| MP3 | MPEG-1 Audio Layer 3, most well known audio format |
| NAT | Network Address Translation |
| NSV | Nullsoft video, an encoding format |
| OGG | Often used to refer to audio format Ogg Vorbis |
| OGM | Ogg media |
| RM | Multimedia container format developed by RealNetworks |
| RMVB | RM with Variable bit rate |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| UI | User Interface |
| VLC | VideoLAN – VLC media player is free cross-platform media player that is under GPL license |
| WMA | Digital audio format developed by Microsoft |
| WMP | Windows Media Player is video player that is shipped with Windows operating system |
| WMV | Digital video format developed by Microsoft |
| VoIP | Voice over IP is technique used to transfer phone calls over IP network instead of normal phone line |

1 Introduction

As amount of media delivered in the Internet seems to be ever-growing and speeds of end-users' access network connections are getting faster by the day, network capacity keeps getting growingly scarce resource. As opposed to traditional client-server architecture, some solutions to relieve network congestion have been proposed, different peer-to-peer techniques being the most interesting and popular ones. In a peer-to-peer network, a single host acts as a server and a client simultaneously. Although peer-to-peer techniques don't seem to directly reduce network load compared to client-server approach, it has been observed that network load is distributed more evenly to the whole network when using peer-to-peer techniques. This leads to single links within the network being less congested.

Peercasting is a method for multi- / broadcasting streaming media, for example audio or video, over the internet using a peer-to-peer network. It can be seen as a combination of traditional television or radio broadcast type of media delivery over a new kind of delivery medium, the Internet. The aim for these techniques is to allow bandwidth-consuming streaming media to be delivered to vast number of consumers without unnecessary network congestion.

Some of the techniques examined in this document promise that they are the solution for media mass delivery over the Internet. Is it so? Is everybody willing to install additional software to their computer compared for example of using YouTube and more importantly are people willing to share their connection to others?

Peercasting techniques are some kind of solution to mass delivery over Internet, but they still need some work to be available for everyone. When users can just turn software on and start watching their favorite series or sports event by choosing the right channel, can these techniques be considered of replacing normal point to point delivery.

This document is structured as follows. After the introduction, the second chapter describes the basic traffic schemes in Internet. Chapter three introduces seven peercasting solutions publicly available. In chapter four we introduce four reference techniques based on traditional point-to-point delivery to compare with the peercasting techniques. After this in chapter five both peercasting and point-to-point techniques are compared with each other. Chapter six is an testing report of peercasting solutions with both client- and server-end applications as well as tests done with reference techniques.

2 Background theory

In this chapter we present three different traffic types in the Internet and their suitability for providing streaming media. After each traffic type, pros and cons related to the traffic type are presented. Finally, peercasting and some related basic principles are introduced.

2.1 Unicast

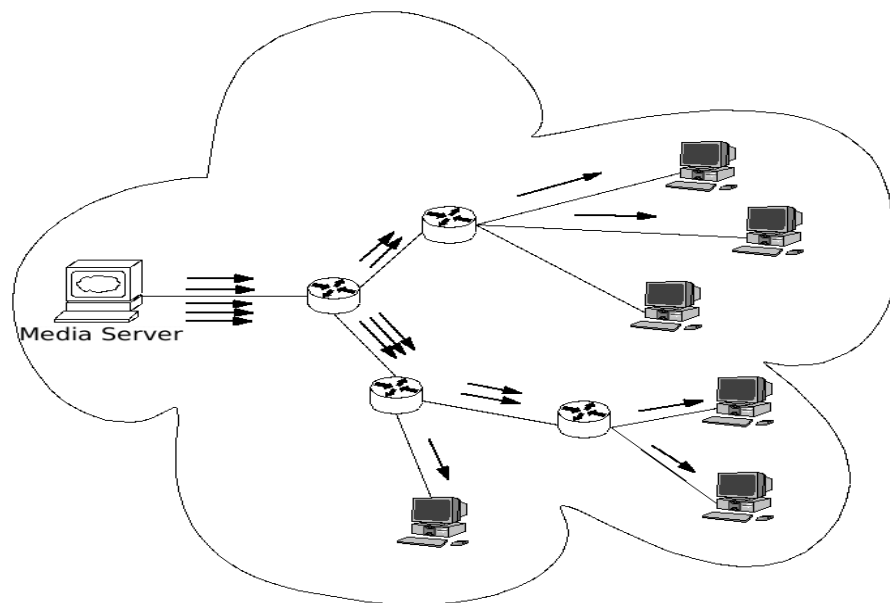


Figure 1. Unicast delivery

Simplest form of traffic in the Internet is unicast. Traffic is based on the traditional client-server -architecture, where individual user-end hosts act as the client and media provider / operator acts as the server end, so traffic is point-to-point in nature. Each client wishing to join the service requests the service from the server and server responds to each client separately. If several clients wish to use the same service having the exact same content (the actual data), same data must be individually delivered to each client. This situation is depicted in figure 1.

The arrows in figure 1 represent actual IP datagrams originating from the server streaming down to the clients. All of the clients have requested the same content from the server and we can assume that this content is delivered to each client at the same time instant. Because of the nature of the unicast traffic, each IP datagram must be replicated

and delivered to each user-end host separately. This is the case, whether the hosts locate in same IP subnet or not. As it can be seen from the picture, the links in the upstream near the media server are far more congested than the ones further downstream. This means that for each user requesting service from given server, the server load and amount of delivered data grows proportionally.

Growing server load may lead to server failure, causing the server system to crash, leading to situation where service provider is not able to provide service content to any of clients requesting it. Also, if enough clients from the same geographical location, that is, the user hosts remain under certain network or behind certain network node (router), and request the same content, the network congestion may cause a router node failure. This kind of node failure could lead to denial of service for the users in the requesting network(s), if there is no alternate route between server and requesting client. When considering streaming media, the amount of delivered payload data can easily grow to be quite massive, so none of the schemes described above seem impossible.

Despite the centralized nature of unicast approach, server load can be relieved by distributing the server system. This means that more than one server is used for serving requesting clients, making single server load more acceptable than in single server case. An example for distributed client-server system is the Akamai, where service content is distributed from a single origin server to edge servers nearer the clients, edge servers then serving the end-user clients [1]. When using this kind of server distribution, single point of failure risk is reduced: client requests may be redirected to other servers in case of server failure.

However, it must be observed that asynchronous delivery is easily obtained using traditional unicast method. "Asynchronous" in the sense that each client wishing to receive data requests the data from the server in its own turn, independent of the service provider. That is, there is no scheduling related to data transfer.

Pros and cons related to unicast:

- + Supported "everywhere" where IP applicable, "guaranteed" transfer (TCP)
- + Service content control easy at the server end, because of centralized architecture (copyrights etc.)
- + Unsynchronized data service easy to implement.
- Server load / network traffic increases linearly to amount of users requesting service

Even though unicast is simple to use and can be used almost anywhere, there are some disadvantages when delivering large amounts of data, as is the case when streaming data is considered.

2.2 Multicast

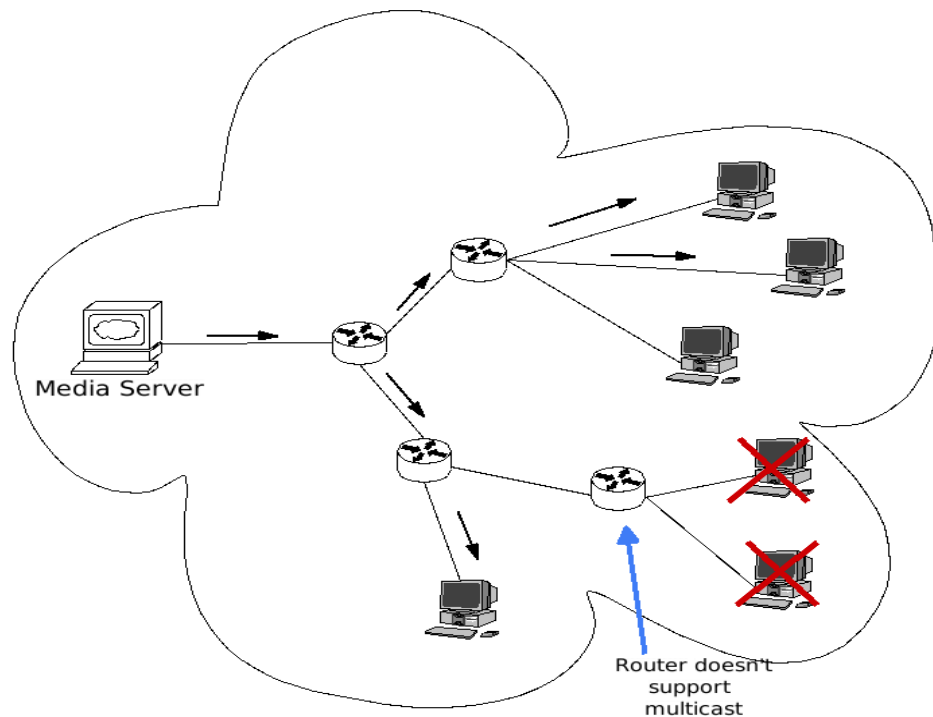


Figure 2. Multicast delivery

Multicast is an elaboration from the simple unicast delivery scheme. While still utilizing the nature of client-server model, the amount of data delivered within the network can be remarkably reduced compared to unicast. This is achieved by using point-to-multipoint delivery. Multicast delivery scheme is presented in figure 2. Again, black arrows represent IP datagrams delivered to clients from the media server. As we can see, the amount of IP datagrams is reduced compared to the unicast delivery (Figure 1).

When using multicast, an IP datagram does not need to be replicated at the server end, for the same datagram is delivered for each receiver. The receivers must register to "listen" the multicast traffic, though, in order to receive the data delivered by the server. Also, the network infrastructure (routers) must support multicast traffic to make it possible to forward the data stream to end-users. As presented in figure 2., one of the routers doesn't support forwarding multicast traffic, so the users located in the network behind the router are not capable of receiving the data, even if desired. So the network infrastructure sets a restriction to service availability for the users wishing to receive the service.

- + Content control easy, as in unicast
- + Scalability to large amount of clients
- + More economical network usage

- + Reduced server load
- Needs network infrastructure support
- More complicated usage than unicast (asynchronous delivery harder to implement).

As you can see multicast reduces needed bandwidth remarkably and single packet needs to be sent only once. Disadvantage in multicast is that it needs support from network infrastructure, leading to more complicated usage.

2.3 Peer-to-peer

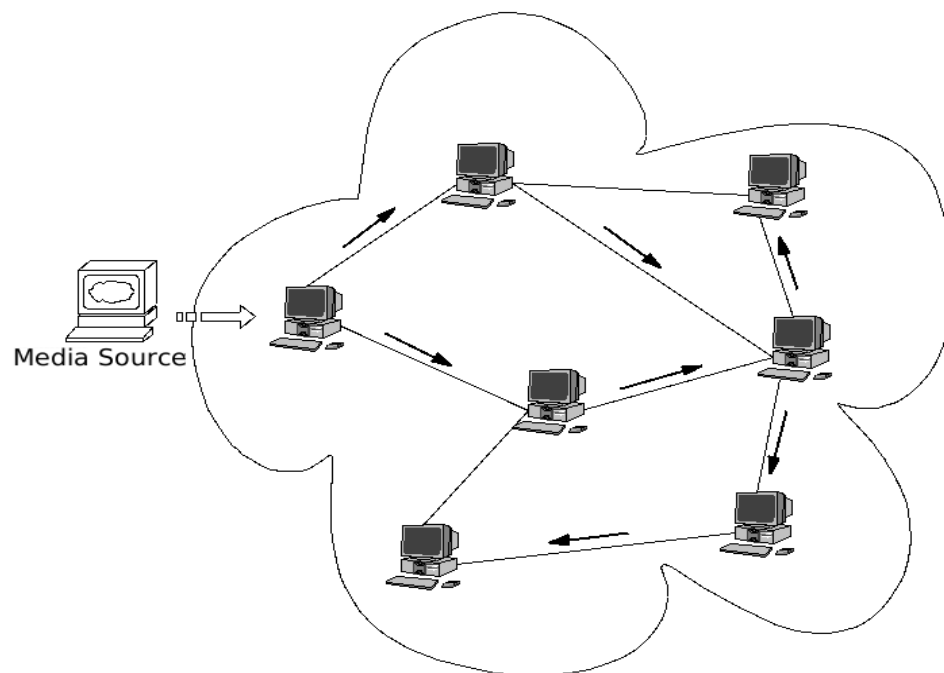


Figure 3. Peer-to-peer overlay network

A peer-to-peer network utilizes client-server -architecture between number of hosts in a network. That is, each host acts as a server and client simultaneously. The network is depicted in figure 3., and as it is obvious, you don't see any routers there. This is because a peer-to-peer network is actually an overlay network. This means that the logical connections between hosts, peers, are formed on higher level than the network (IP) level. Typically the peer connections are formed using TCP.

As each host acts as server and client, the actual traffic is unicast (point-to-point) in nature. But because the hosts are distributed all over the network, the actual traffic load is distributed more evenly over the whole network. In contrast to an ordinary server-client architecture, where there is one centralized server (or more, if server system is distributed) serving all client hosts, peer-to-peer network utilizes multiple “servers” distributed over the whole network.

Because a peer-to-peer network is an overlay network working on top of the network level, direct comparison to the traffic amount of network-level (IP) multicast isn't possible. The reason for this is that peer-to-peer network communication already contains more overhead (IP overhead + TCP overhead + additional inter-peer traffic) than the multicast traffic, whose overhead typically consists only of IP and UDP overhead. The TCP overhead results from the connections between peers, because two adjacent peers can reside two or more hops (routers) apart from each other. The inter-peer traffic is the protocol-dependent data communicated between peers in the network.

Before being able to transfer data in a peer-to-peer network, a host must first somehow join the overlay network and the means vary between different peer-to-peer protocols. There is one major advance compared to multicast, though: Because peer-to-peer traffic often relies on the existing network infrastructure capable of unicast traffic, there are no or a small number of users left outside of service from the traffic reachability point of view.

Because the lack of actual media server in peer-to-peer network, there must be some way to inject the content for delivery to the network. This is usually done by making the data available on one or more hosts within the network (media source), allowing the content to be delivered to the users. This makes content controlling difficult in the network, when compared to centralized content servers used in multicast and unicast networks.

Pros and cons:

- + Available everywhere
- + Even distribution of network traffic over the whole network
- Network content hard to control
- Limited upstream bandwidth when “modern” asymmetric access networks are used

Compared to unicast and multicast, peer-to-peer is the most rapid way to transfer data. One individual with limited bandwidth can share large files over huge target group. Although peer-to-peer utilizes more bandwidth than multicast, it is used more widely because it doesn't need support from network infrastructure, for example routers.

2.4 Peercasting

In this chapter we describe basic principle of peercasting. After that some basic concepts and algorithms related to peercasting are introduced.

2.4.1 Basic idea of peercasting

Peercasting, often referred to as peer-based multicasting or peer-to-peer streaming, is a concept for distributing streaming content over a peer-to-peer network. Also the term “application level multicasting” is sometimes used because of application layer (layer 4)

connections are used to form the peer connections. The streaming media needs to be injected to the network for delivery, and it is further being delivered through the whole network to the clients wishing to receive the data. Sometimes the users may not want to receive the data (for example a TV channel), but will act as a relay node, also referred to as "reflector", to other clients within the network. Reflectors are hosts that pass the streaming data to other hosts without consuming it themselves.

As these networks are peer-to-peer networks, the user wishing to receive (or act as a reflector) needs to join the network before actual data traffic can occur. After the user is joined to the network, there is a varying warm-up time before any data can be consumed. This is because there needs to be initial buffering of data before stream can be presented to ensure seamless viewing or listening of streaming media. The length of warm-up period depends on the amount of users attending in the network, as well as users' network capacity and overall network latency. These parameters may have different effect depending on the protocol used in the network. For example, after exceeding certain amount of users the warm-up period may get longer and stream quality may suffer, rather than be improved as we might expect.

Also the software used for receiving and playing the stream and stream encoding format affects to the duration of warm-up. When larger reception buffer is used in the software, the longer the warm-up. Higher quality (less compressed, higher bit rate) stream usually takes longer warm-up than poorer quality.

The network layout varies depending on the technique in use. Typical layouts are tree, diamond and mesh. In a tree-layout network, the stream is divided to several hosts in each node, so that after each node the amount of receivers within the network is multiplied. So, each host in the network acts as a point-to-multipoint server, so that one host receives one stream and delivers it down to several hosts. In a diamond or mesh [2] network, each node is connected to several other hosts, and each can receive and send out multiple streams (Figure 4).

On a tree layout network there exists a single point of failure type of problem. When considering strictly a tree layout network, every node of the tree (excluding leaf nodes) is a root node having one or more child nodes. Stream is always passed through the root node to its children. If any of the root nodes happen to fail, the whole network originating from this failed root nodes fails to receive the stream. Worst case scenario is that the primary root for the whole tree fails, leading to denial of service for the whole network.

Such single point of failure issues can be avoided, or at least reduced in a combined tree-mesh or a mesh layout network. In these kind of networks all or at least some of the peers (nodes) have more than one connection to other nodes, so that instead of just passing the stream from root to children, also children to root -direction is used. This allows other nodes to receiving the stream when a single (root) node fails. Thus compared to tree layout, mesh-layout requires more complicated routing algorithms or request mechanisms between peers due to increased number of peer connections.

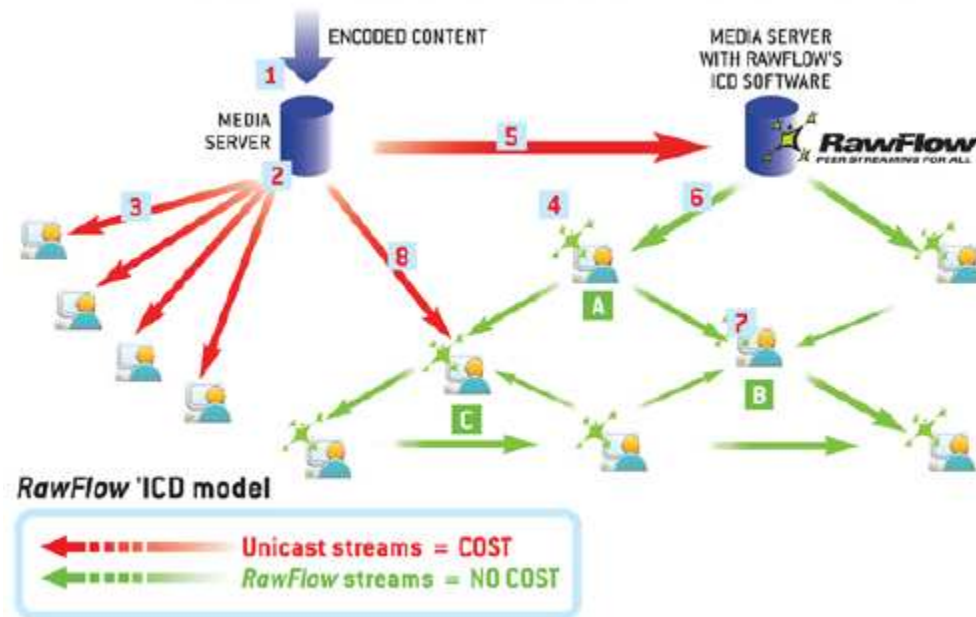


Figure 4. RawFlow peer streaming [3].

Pros and cons:

- + Scalability
- On a tree layout, "single point of failure"-type of problems.
(can be avoided using mesh-layout)
- Limited upload bandwidth

2.4.2 Push and pull methods

One common feature shared by some peercasting systems is being push-based systems. This means that after a peer has received data (the stream) it sends it on to other peers in the network, without explicit requests on the data from other peers. Forwarding decision is based on some predetermined routing algorithm, and the same algorithm is globally used over the whole network. This also leads to the network layout to be somewhat rigid, at least to some extent, because it is determined by the routing algorithm.

The problem with these push-based systems is that they are poor in recovering from transmission losses, caused by the lack of requests for data. For example, if peer connection is broken between two peers, a sending peer fails to forward the data to the receiving peer across this broken connection. This leads to the receiving peer never receiving the data, because of the broken connection, thus experiencing corrupted stream. Other problem in a push-based network is the amount of duplicate data. Because of the routing algorithm used for "blindly" forwarding (pushing) the data, it may well be that one or more peers may be sending same packets to a common destination host. [4] This can be avoided using requests to get desired packets from sending peers, leading to pull-based system.

In a pull-based system, peers wishing to receive the stream request missing packets from other peers. That is, a peer wishing to receive a packet from other peers must request it prior to receiving. After receiving a packet, peer must notify other peers about the packet it received in order to pass the stream along in the network, thus enabling other peers to request the data.

However, if for some reason a packet is not received by a peer, it may request it from one or more peers announcing to have that packet. This results in better resilience against packet loss in reception, because in the case of a failure the receiving peer can redirect request packets to another peer having the wanted data. Also, when using a request based method, there is no need for using predetermined routing algorithms as in push-based approach. [4] The amount of duplicate data sent within the network is reduced, because requests from the peer wishing to receive a packet may only be addressed to one sender who provides the packet to receiver.

A great weakness with pull-based method is dealing with free-riders. A free-riding peer is only requesting and receiving packets from other peers, without uploading anything to others. Because of free-riders, other peers may not receive full streams, because free-riders do not send requested packets.

In a pull-based network the layout may well be more "ad-hoc" in nature, because there might not be any predetermined way to form peer-relations, but the requests and announcements sent within the network define peer-relations.

2.4.3 Multiple-stream approach

To deal with for example the free-riding problem apparent in single stream- utilizing peer-to-peer streaming networks, there is an alternative approach to deliver streaming data: using multiple streams. In this approach a single stream (full stream) is divided in to several substreams, also referred to as "descriptions" [5]. Each substream is then forwarded separately in the network. A peer receiving all of the substreams can reconstruct the full-quality original stream from the sub-streams received, and therefore enjoys best quality.

Using multiple streams in a P2P-streaming network is a potential way for implementing an incentive mechanism [6] in such a network. The incentive mechanism can be considered similar to "scoring" in traditional P2P file sharing networks, where uploading peers are scored by the amount of data they upload. The more scores peer has, the more privileged are peers download opportunities. In a streaming network, peer uploading more streams has better "scores", thus being entitled to download more substreams and getting higher quality overall stream himself. For example for every uploaded stream a peer is entitled to download one stream, thus enabling fairness in the network among peers.

In addition to free-riding, multiple stream approach helps to relieve churning phenomenon also. If a peer sending out one substream leaves the network, the overall

stream quality doesn't collapse remarkably, because satisfactory quality of the overall stream can be achieved from the substreams that still exist. In churning multiple (probably hundreds) peers rapidly attach and detach to the network within a short period of time, making substreams rapidly available and unavailable. Compared to the single stream situation: a peer sending out full stream leaves the network, and possibly being only provider for that stream cuts off reception for all the peers listening that stream sent by the detached peer. Large amount of peers joining and parting the network in single stream system may cause full loss of stream at times.

3 Peercasting techniques, software and service providers

In this chapter we introduce peercasting software and techniques that can be tested and evaluated to some extent. This chapter also serves as an overview of the software and the image of the product from the end users' point of view.

3.1 SopCast

SopCast is a free peercasting program. It is closed source and based on their own Sop technology and protocol which is named Sop://. The SopCast Team started to work with the project in December 2004 and have gained considerable popularity especially in China. Because the software is closed and the developer team has not published any technical papers of their protocols, the SopCast software is very hard to compare in theoretical sense to other techniques.

SopCast has its own client and server software built to Windows operation systems. The software is divided into three different components: SopPlayer, SopServer and WebPlayer. Although they have built their own UI in which they have embedded everything needed to view streams, the software needs either Windows Media Player (WMP) or RealPlayer 10. Separate media player applications are required for showing the content to user. Notable is that using incorrect version of RealPlayer will cause crash of their SopCast player (version 1.0.1) [7], which is also told in SopCast's web pages. This can be interpreted so that software isn't complete yet.

In addition to Windows there is software built for Linux platforms by third-party provider. The Linux application is also available from SopCast's website. Next are listed SopCast's features as mentioned on their website [7]:

1. A very simple exclusively your media platform, real-time audio and video publishing your own programs
2. Setup your own channel group and limit the viewers
3. Build a group based on interest and easy to share
4. Subscribe your favorite broadcasting group
5. Get stream data from many nodes on the SOP network at the same time, make the channel more available and stable.
6. Support many stream types, like asf, wmv, rm, rmvb. etc.
7. Support multiple files play loopy.

8. Support broadcast source quality and channel quality monitor to help to select a perfect channel
9. Total memory cache, no harm to the harddisk.
10. Support authentication on both the source and the clients depend on the uses choices.
11. Support logging and analyst all the channels visit.

What is notable from the feature list above, is that they mention only stream types that are associated to either WMP or RealPlayer. After some research, we found out that other supported types are are *.mp3 and *.spl (Sop play list). In addition to peercasting and internet TV, SopCast is integrating VoIP (Voice Over Internet Protocol) to their client in near future. However, it is not mentioned when exactly is this integration taking place.

To use SopCast, users are not forced to register and software can easily be downloaded from the website. However, registration is required, if user wishes to broadcast, i.e. start his own channel(s). A channels may either be public and shown to all SopCast users or private, so that your stream can be reached through a link, for example to advertise the stream on your own website.

SopCast's homepage itself is quite clean and easy to use. It does not have the most polished professional look among the software providers' website mentioned in this chapter. Mentioned web pages contain some misspelling and deprecated information. We had also from time to time problems accessing these pages and also to download the SopCast's client.

3.2 Octoshape

Octoshape is a Danish company founded in 2003. They have also built their own protocol used for live-streaming radio and TV broadcasting over the internet. There is a technical overview on Octoshape available, written by Alstrup S. and Rauhe T. [8]. The overview is about the general principles how Octoshape works. This overview may seem rather questionable and marketing oriented, rather than being focused on the technical aspects, even though one of the two writers is CTO (Chief Technology Officer) of Octoshape, the other being CEO (Chief Executive Officer). But like SopCast, Octoshape is closed source and therefore there is no detailed information available about the protocol(s) they use.

Currently the content delivered by Octoshape is video and audio. The software is available on many different platforms, Windows, Mac(beta) and Linux(beta). The Octoshape client is plug-in based, built with Java, so users need to have Java Virtual Machine in order to use Octoshape in the first place [9]. The plug-in downloads the media stream to the users computer and streams it to local UDP port, from which any stream-capable software can play it. So, in order to view or listen Octoshape streams, a third party software capable of playing HTTP stream, for example WMP 6.4 (or later), VLC or

WinAmp, is required [9]. When viewing actual streams there are at least two software's running Octoshape plug-in itself and 3rd party software to play the streaming content. Octoshape was also responsible of broadcasting the Eurovision Song Contest 2006 over the Internet.

Octoshape claims that they reduce 97% of the bandwidth consumption compared to traditional unicast streaming services [9]. The total bandwidth consumption in the whole network (server-to-clients and clients-to-clients) is still about the same as in traditional unicast traffic explained in chapter 2.1.

Octoshape's website has a quite professional look, there are no advertisements or banners flashing. On the website users can find all the information required to understand what they are doing, although not technical specifications. To keep the actual service content professional and avoiding the illegal streaming etc. Octoshape does not offer free streaming services for home users, only companies and organisations can have free trials of the server application. This is good thing to keep up the image they are providing with the content offered, but this rules out the home users that want to broadcast their own streams. Octoshape is updating the website frequently, at least the pressroom-page, where all publications concerning their operations are put. [9]

3.3 TVU networks

TVU networks corporation has the least amount of information available on their website about the techniques used. Nearly all of the sub-pages are under construction. One can still find easily the software implementations available to Windows (server and client) and Linux (server). Even though their website is not finished, the Windows client offered has quite professional look and works fine.

In addition to TVU we also came across Viidoo, which seems to be some kind of TV-guide/web integration for TVU player. Although TVU is closed source software, there are no payments for broadcasting the content.

At some point there was also an advertisement “Coming Soon: TVU's global TV service” [10] on the web page, so is the current software just a small test? TVU's news and press pages are also under construction and because of that, users can not be sure how active they are, or what they have accomplished.

TVU requires WMP 9 or above and IE 6 or above to work properly. There is some information available on Wikipedia, which claims that TVU has used VLC at some point during the development [11].

3.4 RawFlow

RawFlow is a company similar to Octoshape and they offer peercasting services for a charge. Unlike Octoshape, a 30-day free trial of their broadcasting server software is publicly available, but requires registration. RawFlow claims to be the leading provider

for live P2P-streaming technology [3] although it remains unclear to which this argument is based on.

Client software is offered to at least to Windows, but we were not able to find any references to Linux usage on their website. In spite of this we managed to get a trial package that also included client software for Linux and Mac OS X . RawFlow, as many other peercasting applications, needs a 3rd party application to play their stream contents, e.g. WMP or RealPlayer. The content itself is delivered via plug-in, which can be installed at least to IE and Firefox. Server software is available for Linux and Windows platforms, as well as for Sun Solaris systems.

There is also a small (technical) review of RawFlow by Jevnaker [12], so we are able get some more specific information about it. The paper isn't too detailed, though, probably to not to interfere with the competition and give out detailed information to competing companies.

RawFlow is based on their own ICD (Intelligent Content Distribution) technology. Basic scheme is that when a peer enters the network, it performs initial buffering from the broadcasters' media server. After initial buffering is done, the peer starts to look for other peers within the network. If no peers are found, the original source server is used as last resort. [13]

Also RawFlow's website has quite professional look, but the actual content on the website is not so good as it looks. There is a lot of content in the pages, but finding what you are looking for is whole different story. For example finding out the requirements of the client software is very tedious.

3.5 PeerCast

PeerCast [14] is one of the first peercasting software on the market. Clients for Windows, Mac and Linux are available, so peercast works on every major platform. Unlike other software, this is completely free, open source and released under GPL¹ license. Because of the restrictions in GPL the PeerCast crew has also a commercial license that can be bought from them. There is also a third party GNOME interface for PeerCast called Geekast².

On average, PeerCast seems to have around a thousand users simultaneously online. This can be seen on their yellow pages, where available service channels for PeerCast are announced. Most of the content available on the Yellowpages³ is Japanese.

PeerCast itself is only software, that is installed to users computer, where it catches the links user clicks on web pages and starts media player, e.g. WinAmp, to play the media user requested. The software itself worked fine without any problems in preliminary tests.

¹ <http://www.gnu.org/copyleft/gpl.html> - GNU General Public License

² <http://home.gna.org/geekast/> - Geekast, a GNOME interface to Peercast

³ <http://yp.peercast.org/> - Yellowpages - Indexing service for PeerCast

The PeerCast website is very simple, but there is all the information needed. And because of the opensource nature, there are many media to contact the crew by, for example IRC (Internet Relay Chat) and Forum.

3.6 Tribler Streaming

Tribler streaming is a video streaming solution created by Delft University of Technology [15]. Like any other peercasting software it allows users with limited bandwidth to live stream video over large audience.

Currently Tribler streaming has clients available for Windows, Linux and Mac OS/X 10.4. Like PeerCast, Tribler Streaming is completely free and open source. It is distributed under the LGPL license. It should also be noticed that because Tribler streaming is research project it sends and receives not only video but also statistics about the stream. Still, no personal data is transmitted. [16].

One major restriction with the Tribler Streaming seems to be that the clients need to have as much upload bandwidth that download bandwidth, for example user with ADSL at 8Mbits/s download and 1Mbit/s upload can only watch streams at most 1Mbit/s [16]. This can be a major problem from the end-users point of view, when considering that most operators offer only asymmetric access network connections, so upstream bandwidth is half, or less, of the download bandwidth.

Currently Tribler streaming utilizes the Chainsaw algorithm [4], which is used to form an overlay multicast system. Notable about the Chainsaw is that it eliminates tree layouts from the overlay network completely. The Chainsaw algorithm works in a client-pull manner as described in chapter 2.4.2.

In spite of the Chainsaw algorithm currently in use, Tribler has planned to moving on using the Orchard algorithm to deal with free-riding and joining and departing the streams at high rate (churning). The means for this is to use Multiple Description Coding, a technique to split the video stream into multiple substreams (see chapter 2.4.3). Each substream is then forwarded in its own spanning tree network, and these multiple spanning trees form a "forest". Mol et al. states that "Our experiments show that Orchard is capable of providing a good quality of service to every peer, even when peers join and leave the forest at a high rate" [5].

3.7 Joost™ (The Venice Project™)

The Venice Project was introduced by the founders of Kazaa and Skype. Formerly known as The Venice Project, on 16th January 2007 product name was introduced: Joost™ [17]. (Not to be mixed with Joost (pronounced *yohst*), which is Java implementation of the Streaming Transformations for XML, abbreviated STX). Despite the new product name, Joost is still in beta phase, promised to be evolved further. Currently the Joost software version number is 0.7.3.1 and it can be downloaded on their website after applying for beta testing and the application has been approved.

Signed-in users get periodically rights to send out invites for new users to sign in for the beta testing program, but it still remains unclear how long is the time interval for getting invite credits. It also unclear if amount of testing on the software, for example amount of time spent testing, affects the interval for acquiring invite credits.

The Joost client is based on the Mozilla framework, which is the means for applying the “social networking layer” (the web-layer) to the client software. The social networking layer is based on “plug-ins”, for example chat capability and RSS reader extensions, included in Joost. For video encoding Joost uses H.264 codec licensed from a US-based CoreCodec company. The streaming library used is named Anthill (name given by Joost). The user interface of the client is based on SVG (Scalable Vector Graphics) technology. [18]

Client software is currently available for Windows XP / Vista platform, but Linux and Mac versions are under development. No server software available.

Currently the website contains pages for screenshots of the software, a short “about us” section with a short description about the people behind the project and a FAQ. There is also a subpage for Joost blog, which has some up-to-date info and contains links to Joost bloggers' (as many as 23 persons) pages, who tell more or less accurate info on Joost (also referred to as “Venice” on the blogs) on a regular basis. For beta testing community there is a dedicated section available, that requires registration and signing in. And, of course, contacts page.

Despite the professional look on the pages, it seems that currently Joost is mainly interested in getting testing personnel for their software, judging by the little amount of content on the pages themselves. But, probably this is just good: they let the product speak for itself, instead of making up marketing slogans.

3.8 Software not included in comparison

Because there are a lot of different peercasting applications available, we could not include all of these to this paper. Here are some of techniques listed what we came across during the research.

During the start of our research we tested some other software as well, which included also PPLive. PPLive seemed fine on the websites, but after installing the version 1.3.20 of the software, avast!⁴ Discovered a trojan that started already during the installation. After removing this file with avast!, the installer process had already ended, but you could not click the finish button. Only option was to cancel the whole installation.

```
C:\DOCUME~1\Saukko\LOCALS~1\Temp\nse12.tmp\EvIDPatch.exe
Win32:Trojan-gen. {Other}
Virus/Mato (worm)
```

⁴ Avast! Antivirus - <http://www.avast.com/>

After closer inspection we discovered that this actually is not trojan, but a tool that allows overriding default limit of 10 simultaneous TCP connections for Windows XP SP2 [19].

Another software we came across was IceShare, which is BitTorrent based Ogg stream distribution software [20]. Because it is currently unfinished and there has not been updates on their Wiki during 2006, we didn't include it to this document.

Because the lack of information available we also left Roxbeam⁵ and ZTE⁶ techniques outside our research. For Roxbeam, the biggest reason was that most of the information about techniques and application user interfaces were written in Chinese and therefore not readable by us. We could have used translation pages like Babel Fish by AltaVista, but then we would have had to guess meanings of some parts and overall translating would have been too tedious work.

Although the ZTE website is in english, we were not able to get any software of further information, even though contacted via email. Because the unavailability of software, ZTE was left out.

Swarmcast⁷ was ignored because it had less service content than another techniques. They claim that they have patent pending on their technology, so this might be the reason to the lack of content. Swarmcast is free to use and has no spyware in it, but it is closed source.

VidTorrent⁸ is P2P-based real-time streaming software. It is opensource and currently under development. Developers have been inspired by BitTorrent, although VidTorrent's source code does not have anything to do with it.

⁵ Roxbeam - <http://www.roxbeam.com/>

⁶ ZTE - <http://www.zte.com.cn/>

⁷ Swarmcast - <http://www.swarmcast.net/>

⁸ VidTorrent - <http://viral.media.mit.edu/wiki/tiki-index.php?page=VidTorrent>

4 Reference techniques

In this chapter we introduce a few techniques for comparison and for being potential alternatives to the peer-to-peer streaming solutions available. These techniques do not cover all available techniques, but are good reference to our paper.

4.1 YouTube

YouTube media company was founded in February 2005. It offers a web page, where people can watch and share videos. The trick is that people can only watch videos without storing them on their computer. This is prevented by Flash player that is integrated to YouTube homepage.

This gives the possibility for users to share their videos so that videos can be removed later when wanted and be sure that nobody has it on his computer. Of course users can record the videos from browser with external tools, but the point is that videos can not be directly downloaded as such.

Because YouTube is open for everyone and as mentioned they are currently serving approximately 100 million videos per day [21]. This may be the cause for some of the following restrictions. Users can only upload videos with maximum size of 100MB and maximum durational length of 10min. At the server end YouTube is converting the videos to a certain quality, this is most probably because of the size and bandwidth causes. These quality settings are MPEG4 (Divx, Xvid) format, 320x240 resolution, MP3 audio and 30 frames per second frame rate [21].

There is one exception to restrictions mentioned above, which is the Director account. The account is meant for musicians, amateur filmmakers and similar groups and it allows video larger than 100MB and longer than 10min to be submitted.

4.2 Bubblare

Bubblare is YouTube-like web service, where people can share their videos. It has been translated to all primary Scandinavian languages (Swedish, Finnish, Danish and Norwegian) and is clearly aimed to that "market". There is even no English translation of their website available. The video player for Bubblare is integrated on their website and is based on Adobe Macromedia Flash and their whole server side technique is based on Java [22], which basically means that all pages and their functions should be almost platform independent. Bubblare is also willing to offer licenses of their techniques, this undoubtedly will cost some.

Because of the beta stage translations, for example Finnish, some parts of the site are not yet translated. One of the oddities in Bubblare is that every language has its own content, that is the videos. So, while browsing Finnish pages you cannot watch videos that have been uploaded to Swedish pages. Users that want to publish videos on Bubblare have to register. Registered users are allowed to upload the videos in same manner as explained above, so uploads are only valid for the translation language to which the registration is made.

As expected, Bubblare also changes the video format to more compatible format for viewing on their website. Unlike YouTube, file size limit for Bubblare is not announced on their website.

4.3 Veoh

Veoh is a video sharing service similar to YouTube and Bubblare, founded in 2004. Veoh has one difference compared to its competitors, that is the client software. Users can download full quality video files to their own computer, while lower quality previews are available on the website for online viewing. The client application is available for Windows XP/2000 and Mac. Veoh is in beta stage like Bubblare and because of that there are some inconsistencies found on their website.

An advantage with Veoh compared to YouTube and Bubblare is that there is no file length (duration) or size limit restrictions. This is because the data shared in Veoh website is poorer quality than the original media, which can be get using the client. Lower quality video on the web pages enables users viewing the video through the web page to consume much less bandwidth. They have also mentioned 10min limitation for watching the online previews in Veoh FAQ [23]. This limitation does not seem to be working at the moment (4.1.2007).

With peer-to-peer technology Veoh is avoiding busy servers and reducing the amount of bandwidth needed for actual data transfer service they provide. Thus, media published in this manner in Veoh spreads around the Internet rapidly and cannot be totally deleted afterwards because users will have copies on their own computers already.

Veoh has named its P2P technology to PeerCasting. This should not be mixed with the term peercasting in the sense that PeerCasting is not utilizing streaming. "By implementing a managed network, PeerCasting can significantly reduce video piracy, improve overall quality, and provide user-specific features not available in traditional peer-to-peer systems. In essence, PeerCasting is the next generation of peer-to-peer sharing." [23].

At the moment when a user wishes to download a video from Veoh to his own computer, he is required to register prior to downloading. After registering user can download the client application and start downloading the videos. When user clicks the "Download" button video transfers automatically into client softwares' downloads section where download-related info is presented, for example download speeds, active peers,

progress and remaining time. When a file is completed, it changes location to Library section where all playable videos are.

Because Veoh does not provide watching videos during downloading it cannot be considered as peercasting software in this sense. This feature would be indeed very nice addition to the current client, although this would bring some problems also when video bit rate is larger than available download bandwidth. In this case users would notice video lagging at times.

In addition to downloading and integrated video player the Veoh client provides also section called *Veoh TV*. Although this may sound like peercasting, it is only a full screen GUI for watching the downloaded videos.

4.4 Zudeo (Azureus 3.0)

Zudeo [24] is alias for the new Azureus 3.0 BitTorrent client. It has completely new UI and integrated content delivery system through which users can download media to their computer. To download content, users need not to contact tracker sites anymore but only start the Zudeo client.

Unlike first impression lets us assume, Zudeo does not include integrated video player, a 3rd party player is required. Zudeo uses web browser to view the pages where content is downloaded and it seems to be only a fresh user interface for Azureus. Core components used are the same as in Azureus 2, only GUI has changed a bit. The content browsing mechanism on the client application is www-based and is also available on Zudeo homepage. Because content is shared as torrent files they can be also downloaded with Azureus 2 application. Zudeo content sharing service is basically a BitTorrent tracker whit fancy appearance.

There is stream button showing already on the Zudeo client user interface, so supposable feature is to be supported: "Coming soon". So, at some point there may be the possibility for the users to watch the videos simultaneously during downloading.

5 Protocol/Program comparison

In this chapter we compare the techniques introduced in previous chapter. Comparison is centralized mainly to software, websites and things we were able to find out about the techniques.

In table 1 we have collected general information about the service providers and solutions, like starting dates (from their website) and the term that they use for their peercasting technique. Half of the techniques reference somehow to TV which indeed is most likely the target for usage. Notable is that these terms may not necessarily be substitutes to “peercasting” on developers' and service providers' websites, but these were used as reference to the subject.

Two of the six peercasting techniques, PeerCast and Tribler Streaming, are open source. In addition to these also SopCast and TVU have free broadcasting software, which need registration though. Although it is mentioned that amateur broadcasting with TVU is free, TVU sells hardware and services to professional broadcasters also [10]. All client applications are free to use.

Table 1: General information

| Protocol / Program | Starting date | Peercasting term | Opensource | Free broadcasting |
|--------------------|-------------------------|------------------------------|------------|-------------------|
| SopCast | December 2004 | P2P Internet TV | No | Yes |
| Octoshape | 2003 | Live streaming TV and Radio | No | No |
| RawFlow | 2000 | Live P2P streaming | No | No |
| TVU | 2005 | Broadband Digital TV Network | No | Yes ² |
| PeerCast | April 2002 | P2P broadcast | Yes/GPL | Yes |
| Tribler Streaming | April 2006 ¹ | P2P system for broadcasting | Yes/LGPL | Yes |
| Joost | Early 2006 | N/A | No | No |

¹ First news added to Tribler Streaming homepage on April 2006

² It is mentioned that amateur broadcasting is free

In table 2 is collected things that might be unwanted by users or might effect on using experience. First there is column for possible adware/spyware/trojans and other things that might be unwanted from the users' point of view. Addition to adware we listed to column next to it if the software has adverts flashing or popping up in the GUI. Notable is that these ads are not same as adware and are usually part of the software which is accepted also by users. The ads column does not include ads delivered with in the stream, e.g. TV-channel ads.

Table 2: General information

| Protocol / Program | Adware / Spyware | Ads in UI |
|--------------------|------------------|-----------|
| SopCast | No | Yes |
| Octoshape | No | No |
| RawFlow | No | No |
| TVU / Viidoo | No | No |
| PeerCast | No | No |
| Tribler Streaming | No | No |
| Joost | No | No |

Table 3 serves as a summary of the technologies and implementations. You can find the protocol names that they use in their products and what are the names of these protocols. The transmission protocol column lists the protocols used by the software to send and receive the stream over the Internet.

In addition to peercasting protocols that are unique in every peercasting techniques there are some additional protocols used in the softwares. Good example of this is PeerCast which uses HTTP to configuring and log and statistic viewing.

Last column tells if the client software has a built-in video viewer. In most cases the built-in video viewer is based on WMP, causing operating system specific restrictions (availability on Windows only).

Table 3: Technical information

| | Product name for the technology | Transport layer protocols | Application layer protocols | Built-in video viewer |
|-------------------|--|---------------------------|-----------------------------|-----------------------|
| SopCast | Sop technology | UDP | HTTP | Yes |
| Octoshape | Octoshape's GridCasting | UDP | ? | No |
| RawFlow | The RawFlow Intelligent Content Distribution (ICD) | TCP ¹ | ? | Yes ² |
| TVU | TVU | UDP | MMS, HTTP Streaming | Yes |
| PeerCast | Peer Cast's networking protocol | TCP | HTTP | No |
| Tribler Streaming | Tribler Streaming | ? | ? | Yes |
| Joost | N/A | UDP | ? | Yes |

¹ using UDP for NAT traversal

² When using RawFlow client, otherwise using applicable external player

In table 4 we have listed the software availability to different operation systems/platforms. It is relatively easy to see that Windows seems have the priority in clients, when development and availability is concerned. But there is also remarkable amount of clients to Linux and Mac OS also. If the software is in beta stage, it is mentioned in the table.

SopCast has its own software built on windows that includes both client and server in the same package. There is also Linux engine available with 3rd party built GUI. In contrast to other software mentioned earlier, SopCast has also a WebPlayer.

Octoshape has built its own client applications for all platforms listed in the table.

RawFlow client application is web browser -based and uses different 3rd party players for playing the streams.

TVU has its own client system built for Windows and there is server software available for both Windows and Linux. During the test period TVU updated its client and server software a couple of times. This can be seen by the user as a message when starting an older version of TVU client, when users are asked if they want to download latest TVUPlayer. TVU clients are available as both stable and beta releases, but server releases are beta only.

PeerCast has wide support of different operation systems similar to Octoshape and Tribler Streaming.

From reference techniques mentioned in chapter 4 only two have some kind of client application available. Zudeo requires user to download it, for there is no alternative way to view the content as in Veoh. Although Zudeo seems to be closest to being peercasting when the reference techniques are considered, the actual streaming functionality is still unavailable.

Table 4: Software availability

First value represents client and second server software (client/server)

| Protocol / Program | Windows | Mac | Linux | Others |
|--------------------|-------------------------|-----------------|------------------|-----------|
| SopCast | x/x | - | x/x ¹ | WebPlayer |
| Octoshape | x/x ² | x (beta)/- | x (beta)/- | - |
| RawFlow | x/x | - | x/x | - |
| TVU | x/x | - | -/x | - |
| PeerCast | x/x | x/x | x/x | - |
| Tribler Streaming | x/- | x/x | x/x | - |
| Joost | x/- | ⁴ /- | ⁴ /- | - |
| YouTube | Flash player on website | | | |
| Bubblare | Flash player on website | | | |
| Veoh | x/x* | x/x* | - | - |
| Zudeo | x/x* | ³ | ³ | - |

¹ The GUI of the Linux version is made by 3rd party not the developers.

² Server software not publicly available

³ Previous version is available to this platform

⁴ Under development

* "Server" included in client

Table 5 describes which encoding formats different P2P streaming solutions are able to use, when transmitting stream over the Internet. This information is collected from the developers' web pages and from their software. As expected, the most commonly used formats, for example MP3, are supported by all peercasting solution.

The reference techniques have more extensive support to different encoding formats than peercasting techniques, due to the different development stages of these two. For example YouTube has been out there for a long time and is using things that are not as new as peercasting and because of that implementation is much easier.

Reference techniques promise to accept all major video and audio encoding formats. As amount of formats is very extensive they are not listed here.

Table 5: Supported encoding formats

| Protocol / Program | MP3 | WMA / WMV | ASF | NSV | OGG / OGM | RM / RMVB |
|--------------------|---|-----------|-----|-----|-----------|-----------|
| SopCast | X | -/X | X | | | x/x |
| Octoshape | X | x/x | | | | |
| RawFlow | X | x/x? | | | | x/x? |
| TVU | | x/x | X | | | |
| PeerCast | X | X | X | X | X/X | |
| Tribler Streaming | N/A | | | | | |
| Joost | N/A | | | | | |
| YouTube | All major encoding formats | | | | | |
| Bubblare | All major encoding formats | | | | | |
| Veoh | All major encoding formats | | | | | |
| Zudeo | Every encoding representable in file format | | | | | |

Table 6 shows the information content of websites of peercasting solutions / service providers described earlier in this document. If the page exists there is X in the table cell. If page is under construction or similar there is O in the cell. If the content of the page mean is under some other page this is mentioned under the table.

Even though there is lots of information on each website, the reachability of the content differs greatly between sites. Good example is RawFlow who does not tell much about their client on the website, unlike for example Veoh.

Table 6: Website content

| Protocol / Program | FAQ | EULA / Terms of service / use | Contact information | Forum | Press | News |
|--------------------|-----|-------------------------------|---------------------|----------------|-------|----------------|
| SopCast | X | X ⁴ | X | X | | X ⁴ |
| Octoshape | X | X | X | | X | X |
| RawFlow | | X | X | | | X |
| TVU | | X | X | X ⁵ | O | |
| PeerCast | X | | X | X | | |
| Tribler Streaming | X | | X ⁴ | | | X |
| Joost | X | X | X | X ⁶ | X | X ² |
| YouTube | X | X | X | | X | |
| Bubblare | | X | X ³ | X ¹ | | X ² |
| Veoh | X | X | X | | X | X ² |
| Zudeo | | X | X ³ | | | |

¹ On Swedish site

⁴ On front page

² Blog

⁵ On Viidoo's website

³ About

⁶ On betatesting area

Table 7 is collection of the client hardware and software requirements for each technique found on the websites.

Table 7: Client software and hardware requirements

| Protocol / Program | Client requirements |
|--------------------|---|
| SopCast | 64MB RAM, 10MB HDD |
| Octoshape | Web browser, HTTP streaming enabled media player |
| RawFlow | N/A |
| TVU | WMP 9, IE 6, Broadband Internet connection |
| PeerCast | N/A |
| Tribler Streaming | 1GHz CPU, 128MB RAM, Bandwith 128Kbit/s |
| Joost | P4 1GHz, 512MB RAM, 32MB VRAM, 500MB HDD, broadband 1M/512kB ¹ |
| YouTube | Macromedia Flash Player 7.0+, Broadband connection with 500+ Kpbs |
| Bubblare | Macromedia Flash Player |
| Veoh | 256MB of RAM, WMP 9 or 10 (Quicktime on MAC) |
| Zudeo | Java JRE |

¹ Recommended

Table 8 is collection of server hardware and software requirements of each technique that has server software available.

Table 8: Server software and hardware requirements

| Protocol / Program | Server requirements |
|--------------------|---|
| SopCast | N/A |
| Octoshape | N/A |
| RawFlow | 400MHz CPU, 256MB RAM, stream server |
| TVU | 1GHz CPU, 128MB RAM, upload speed 2x signal bandwidth |
| PeerCast | N/A |
| Tribler Streaming | N/A |

In table 9 there is summary about file's size and quality limitations of the systems. First there are all peercasting techniques and after that all reference techniques so it is relatively easy to see the advantage that peercasting techniques provide compared to typical unicast delivery.

Even though with point-to-point technologies it could be possible to offer media with unlimited quality this is not the case in the face of great number of clients. In order to avoid congestion and server overload caused by simultaneous downloads, the amount of data must be reduced. This can be easily achieved by lowering the media quality.

With P2P technology this limitation of point-to-point delivery is reduced, because traffic load is divided in the network, thus allowing higher quality of media. However, the upload bandwidth may limit the quality that can be used.

Table 9: Video and file size limitations

| Protocol / Program | Filesize and length limitations | Video quality provided with the system |
|--------------------|---------------------------------|--|
| SopCast | No | Original |
| Octoshape | No | Original |
| RawFlow | No | Original |
| TVU | No | Original |
| PeerCast | No | Original |
| Tribler Streaming | No | Original |
| Joost | No | Original |
| YouTube | 100MB, 10min | MPEG4 (Divx, Xvid), 320x240 resolution, MP3 audio, at 30 fps |
| Bubblare | N/A | Reduced by Bubblare |
| Veoh | No | Reduced quality in previews on website |
| Zudeo | No | Original |

We did not include any measurements to this section because it is very hard to compare the techniques in this sense. However, there are some opinions on next chapter about the functionality and speed of the programs.

Because of growing number of portable devices such as laptops and handhelds (PDA), users are often travelling and not able to have connection to Internet. Because of this we listed if the different services offer ability to download or record media to users own personal device so that could be watched afterwards offline.

Table 10: Ability to watch content offline

| Protocol / Program | User can download/record media to own computer |
|--------------------|--|
| SopCast | Record |
| Octoshape | -- |
| RawFlow | -- |
| TVU | -- |
| PeerCast | -- |
| Tribler Streaming | -- |
| Joost | -- |
| YouTube | -- |
| Bubblare | -- |
| Veoh | Download |
| Zudeo | Download |

In table 11 we have collected our opinion about each technology from software usability, quality of media and homepage. The scale that is used is following Poor, Satisfactory, Good, Very Good and Excellent.

Software usability includes informativeness of the UI, visual impression and ease of use. The usability of applications differ greatly, poorest being Tribler and TVU the best. As Tribler is related to a research project, usability might not be of importance in

this sense. All other applications are more or less for commercial purposes and usability should be considered of higher priority.

Quality of media describes the perceived quality of stream, except in the case of Zudeo and Veoh, where quality of downloaded media is also evaluated. The downloaded media is dependent on the encoding originally used and therefore considered excellent.

Websites are judged by visual appearance and scope of content as well as ease of navigation on the site. None of the examined were near perfect, and some were even confusing.

Table 11: Subjective opinion

| Protocol / Program | Software usability | Quality of media | Website |
|--------------------|--------------------|--|--------------|
| SopCast | Satisfactory | Satisfactory | Satisfactory |
| Octoshape | Good | Good | Good |
| RawFlow | Satisfactory | Satisfactory | Satisfactory |
| TVU | Excellent | Satisfactory | Satisfactory |
| PeerCast | Good | Satisfactory | Satisfactory |
| Tribler Streaming | Poor | Poor | Satisfactory |
| Joost | Very good | Excellent | Satisfactory |
| YouTube | Good | Satisfactory | Poor |
| Bubblare | Satisfactory | Satisfactory | Poor |
| Veoh | Good | Satisfactory ² / Excellent ¹ | Good |
| Zudeo | Satisfactory | Excellent ¹ | Satisfactory |

¹ Original media quality

² Preview

6 Testing the software

In this chapter we provide information about the clients and server softwares which we introduced earlier in this document. During the test period, most of the providers updated their software.

6.1 Testing environments

Test were done in several different computers and with different Internet connections. This was done to make sure that it is possible to use these techniques on different platforms and connections.

Computers used in tests are listed in table 12. All tests are done under Windows XP Pro SP2 unless mentioned otherwise. Tested software in table 12 does not include reference techniques.

Table 12: Environment(s)

| PC # | Hardware | Operating system | Connection | Tested software |
|------|------------------------|-------------------------|---------------|---|
| 1 | A64 3500+, 2GB | Win | 8/1M ADSL | All |
| 2 | AMD XP 2400+, 1GB | Win | 1M/512kB ADSL | All |
| 3 | Intel P4 1.7GHz, 512MB | Win | 100MB LAN | All |
| 4 | AMD Duron 1GHz, 256MB | Linux (Debian) | 100MB LAN | PeerCast, RawFlow, Octoshape |
| 5 | P3 500MHz, 256MB | Win | 100MB LAN | SopCast, PeerCast, TVU, RawFlow |
| 6 | Intel P4 1.7GHz, 1GB | Linux (Debian) | 100MB LAN | RawFlow, Tribler (server), TVU (server) |
| 7 | AMD Sempron 2800+, 1GB | Win / Linux (Suse 10.1) | 100MB LAN | Joost, Tribler, SopCast, TVU |

6.2 Clients

This chapter includes basic practical tests of the client softwares.

6.2.1 SopCast

SopCast had quite slow buffering in our tests, but in comparison it did not stand out much from others. SopCast allows anonymous usage from clients that only watch the streams so it does not need much configuration and is ready to use almost instantly.

We had some problems starting SopCast tests on the first test computer (PC 1) on which the software did not work at all. On the second computer SopCast worked without any problems and finally we got it working on three of the four computers we tested SopCast with.

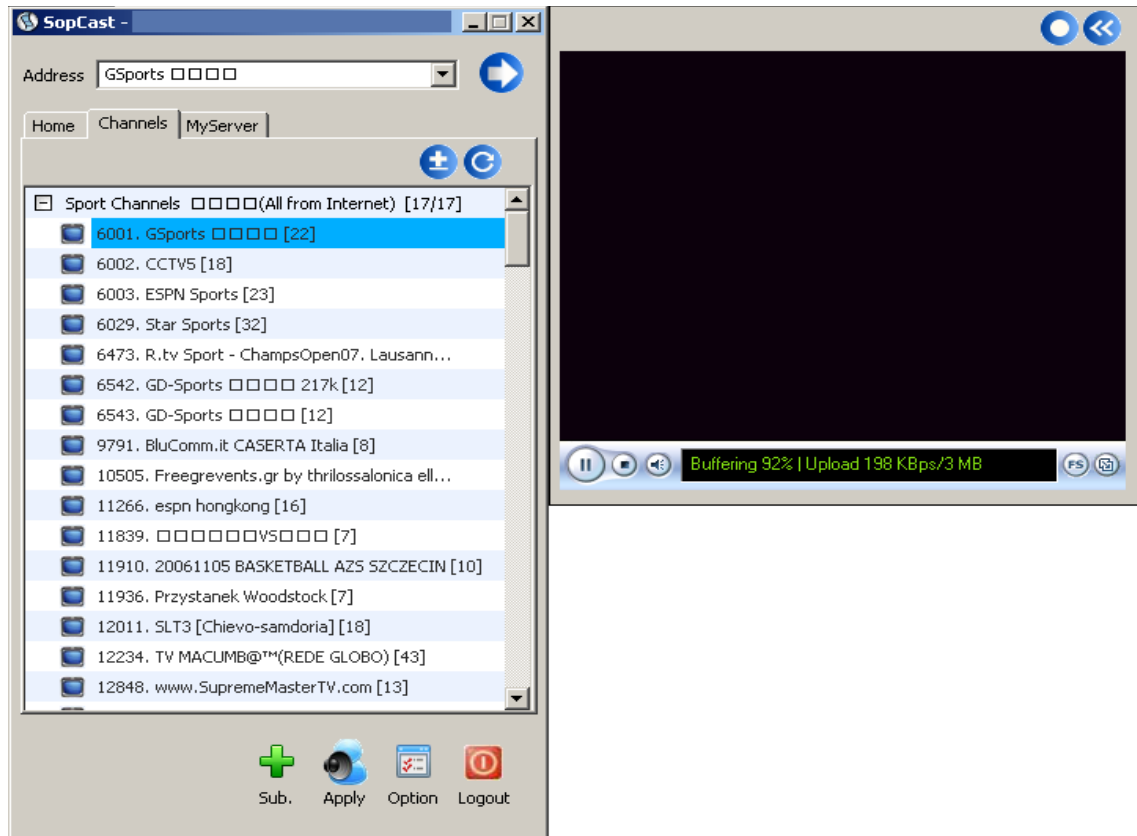


Figure 5: SopCast client

SopCast has quite complex GUI (Figure 5) compared to other techniques, for example TVU or Joost, that might push users away from using it. There is also some inconsistencies, e.g. when browsing the channel list it might automatically update and take your cursor to the top of the list. This is merely a usability issue, but a rather remarkable one. The list is also quite hard to use because users can add their channel to many categories at the same time which is both good and bad thing. It enables broadcasters to add their channel to every matching category but when a user is browsing the list you find same channel over and over again, but now. This does not matter if you want to watch and “surf” between sports channels, which are on the top of the list.

Most of the content in SopCast is sports related, which is mentioned in their websites title “live football, NBA, cricket” [7], but there are also channels in entertainment and music categories. The legality of some of the channels is questionable, but as always the software producers do not take the responsibility.

Also when browsing the files in SopCast's installation folder on the computer, we found references to VLC media player 0.8.1. This is probably only used for streaming the content, because WMP is required when using SopCast on Windows.

Some inconsistencies were also found, for example License.txt files mentioned version 0.9.5 even though the application version we used was 1.0.1. One notable thing is that SopCast's forum was hacked by Turkis-Hacker Sinaritx for at least a month (noticed 26. December 2006), why do not they do anything to that?

6.2.2 PeerCast

PeerCast worked fine on Windows XP. After installing the software all `peerCast://` links were redirected to the client software, which started a media player, in our case WinAmp, for playing after the media type was recognized. It was very pleasant to notice that media started to play almost instantly after media players started.

PeerCast has very simple UI, mostly this is because there is not integrated media player of any kind in same software. PeerCast has only basic stuff available in content menu which opens from the PeerCast icon when it is running. About half of the menu items open the default web browser on users computer.

PeerCast also has a web browser-based administration interface (Figure 12), which can be reached through `http://localhost:7144` by default. This interface shows information about users current broadcasts, bandwidth usage, opened connections, logs etc. It also allows remote administration, for it can be reached using web browser. By default, remote access is disabled, but it can be enabled by entering user password on local host Settings-page using the same browser interface.

During usage PeerCast informs user using pop-up messages. Messages are divided into four categories which are PeerCast, Broadcasts, Track info and Upgrade alerts. User can choose from these which messages he wants to be shown. In addition to this there is also a small GUI built for broadcasters, where user can set the local port through which the data is transmitted (by default 7144). When broadcasting Shoutcast, also one additional port is used (7145 by default).

PeerCast is very simple to use, although the browser-based settings and log-viewing might be surprise for some of the users. Disadvantage of this type of web-UI is that user is required to have a web browser. There are also advantages in this. For example when dividing software into smaller parts, it is more easy to control and less likely to crash. If for example a web browser or media player crashes, the PeerCast software itself keeps on going and does not cause any interrupts to stream. Also the size of the PeerCast software is reduced because of this.

When we used PeerCast the first time we were surprised that it did not need any configuration before client-end usage. Users with less expertise can run PeerCast using default configuration by just installing it on their computers yet more options can be offered to advanced users. The ease of use applies only to Windows, though. Linux client needs extra configuration in order to enable the redirection from `peerCast://` links.

User can find content to PeerCast easily on the Yellow Pages. These are simple by appearance and contains some information on different streams, for example listener statistics. User can search streams by genre, bit rate and type or enter keywords to search for a wanted channel. When the wanted channel is found, he can click the Play button next to stream on left and start viewing/listening the stream. There is also chat functionality for each created media channel in the yellow pages where people can talk with each other about the channel.

When user is changing from music stream to another there is nearly no "wait time" at all. This is because PeerCast is playing the earlier stream until new one is found.

Overall, PeerCast is good package, it is easy to use and does not need much expertise.

6.2.3 Octoshape

The beginning of this chapter is mainly about Windows client. Linux specific test results can be found at the end of the chapter.

The Octoshape plug-in was easiest to install from the client applications we tested. After installing, Octoshape URIs (octoshape:) were recognized and the plug-in started the playback on a media player, in our case WMP. The Octoshape URIs are used for identifying the media at server end.

User can find content from Octoshape's homepage⁹. The content is provided by companies like TV and Radio stations and has good stream quality. Quality of these streams vary from 64kbps audio to 800kbps video. Linux playlists reside on their own page.

Octoshape does not offer their server solution for home users, even no trial version is available. Although this makes the content of Octoshape very professional-like, it is major setback for the "home DJ" not being able to broadcast own content freely.

When installing the software there is an End User License Agreement (EULA) that user needs to agree with before continuing the installation. In the EULA one part that caught our attention: *"You may not collect any information about communication in the network of computers that are operating the Software or about the other users of the Software by monitoring, interdicting or intercepting any process of the Software."* [25]. So, they practically deny usage of applications like Wireshark and Netlimiter that monitor the network usage on users own computer. This is understandable if they wish to prevent backward engineering of their software. Still it might be too much for some users wanting to monitor traffic on their host, for example because of security reasons.

The client application itself is very easy to use and practically needs no user actions to work. After installing the client is ready to use and catches all the links with octoshape: prefix. It is very independent and does not need user interaction even later when newer versions are released, because of auto-update feature. This auto-update can be turned off if users do not wish to use it, but it still offers much assistance to users because they do not need to check periodically Octoshape website or notice in some point that their client does not work anymore.

Client offers only five menu items which are Play, Help, Settings, About and Exit. From these Play and Help starts web browser and Settings GUI (Figure 6). In this GUI user can change five things language, player, browser, startup and idle. In the GUI there is two choices, to user Octoshape recommended player/browser or system default browser,

⁹ <http://www.octoshape.com/play/play.asp> – Octoshape channel list.

what is the recommended one is not said there, but after inspection as expected choices were IE and WMP.

Octoshape streams that we tested worked mostly fine. Only problems that we encountered were when no stream was found at all and error, i.e. "You can not get this stream now. You may try again later", was returned by the Octoshape client. When this kind of error occurred also WMP returned error, i.e. "Windows Media Player cannot find the file.", that it can not play the stream. Besides these errors streams worked fine with exception that 800kbps video streams had sometimes small lag in video which did not cause much reduction in viewing experience.

Linux client was also tested and like Windows client it worked very well. But unlike the Windows client, it needed some configuration. Needed steps to configuring were told in a detailed manner and were also very easy to follow. One thing that might be hard for beginners is that Octoshape client to Linux is command-line based, so no GUI was available. There was no automatic link following either. In fact the URIs did not contain the octoshape: prefix, and were not presented as links. Users must take certain link name with copy/paste-manner from Octoshape Linux playlist and start the command line application with that.

Overall impression of Octoshape is very good. It works like a charm requires no configuration in Windows and very little in Linux, that is, Java executable location entered in setup file.

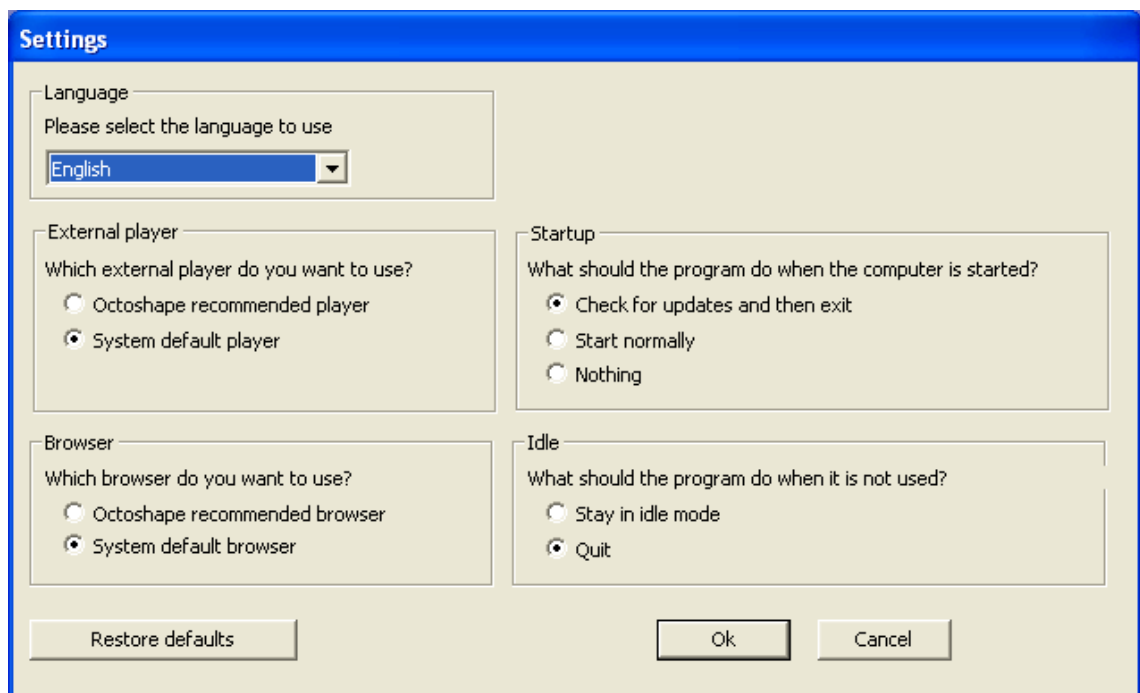


Figure 6: Octoshape settings dialog

6.2.4 TVU networks

TVU's player is most integrated from the software we tested. It has integrated media player which is WMP based. The player works fine and has quite clean and TV-like look. There is not specific information about streams shown, for example statistics, source and peer amounts, but instead a meter that indicates signal quality. With this meter they try to give an impression of TV rather than P2P application.

Although TVU provides all its features in same GUI, i.e. client and media player, it is very easy to use. TVU is using WMP internally for playing the stream, so there is only one application running.

First thought about this software that came to mind is Home Theater PC (HTPC) software, that provides simple GUI (Figure 7) to use, for example to DVB-cards and other media elements in your PC. To view content, user just needs to start the client and it is all there. There are no extra tasks needed, for example to search a web page for a stream. During the tests we started to wonder if TVU has a centralized server solution that contains all statistical data in once place. This is because Earthquakes in coastal China in October 2006 caused huge amounts of problems and the whole channel list was unreachable as were the web pages also. Later, TVU informed about these problems on their website.

After closer look to the files in directory of TVUplayer on Windows we found also a file named wvc1dmod.dll. This is Windows Media Video 9 Advanced Profile Decoder which they are most likely using for decoding video streams.

There is also an automatic updater which checks that user is running the latest version of client application. This updater was very slow during the tests, so we did the updates by manually reinstalling software.

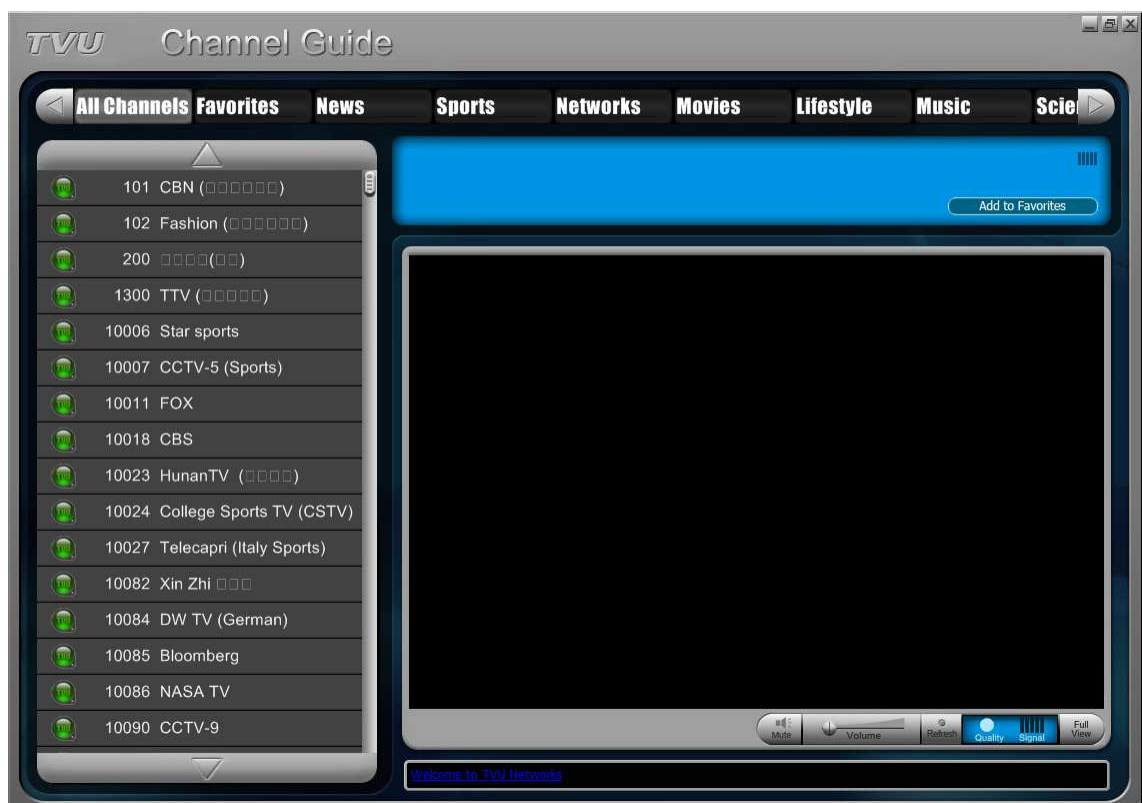


Figure 7: TVU client

TVU has also age limit mentioned in their Terms of Service: *“Eligibility. the TVU Networks Platform IS not available to persons under the age of 18 or to any Users previously suspended or removed from the TVU Networks Platform OR NETWORK by TVU Networks.”* [10]. This may be due to differences in legislation in different countries. In some countries there might be legal restrictions for viewing adult content, for example. So, by defining this kind of limitations in TVU terms of service, they can avoid dealing with this kind of details. If this is the case, there could be better ways of communicating such a restriction, for example EULA of Octoshape: “... not intended for use by or availability to persons under the age limit of any jurisdiction that restricts the use of Internet-based applications ... “ [25].

6.2.5 RawFlow

The RawFlow client application a lot differs from other peercasting softwares. It actually does not have a separate client application that users have to download and install. When a user enters website containing the ICD technology, the site offers to install a plug-in if it recognizes users web browser and operating system. But, if recognition cannot be done, user is redirected to location where the original data stream is delivered from. RawFlow refers to this server as “Fallback server”. The fallback server is also used for injecting the stream for delivery into the ICD network.

When the ICD technology is enabled on users computer, the website opens an embedded media player, e.g. in Windows WMP, and starts to play the content in a web browser window. For IE there is an ActiveX component. A Firefox a plug-in is available also.

Any P2P delivery was not discovered during the tests, this may be due to relying on the original stream source, which is characteristic for ICD. A bigger testing environment network with multiple routers is probably needed to activate the P2P functionality. This may be due to ICD LAN intelligence, see [13].

We also tested the stream in two different browser windows running simultaneously on the same computer. In this case we had over one minute delay between these two transmissions.

The client application removal is hard, because it is plug-in-based. The plug-in must be located and removed in order to remove the client. At least in IE7 this was done under “Internet Options -> Programs -> Manage Add-ons”, where the ActiveX component. On Firefox (Linux version) the plug-in can be found when typing “about:plugins” to the address bar, but it must be manually removed under Firefox profile directory.

6.2.6 Tribler streaming

Tribler streaming was the only application having their own video player built-in and integrated to their software. Because they are using integrated player, the software may be more easily portable to various platforms than if using already available solutions.

The GUI (Figure 8) on Tribler is very simple and has nothing extra. Only selections for server and stream quality existed. Users can only choose buffer size and do they want to use half or full quality of the stream.

Because Tribler streaming is a research project, the software is not very professionally built and decorated. It does the job, nothing else. Tribler has its own dedicated streams available on their website and these can be tested by users. These streams were not good in quality, which might be because they are just testing the mechanisms behind the Tribler Streaming. These are also mentioned to be “test streams” on their FAQ, so stream quality is intentionally poor.



Figure 8: Tribler streaming client

6.2.7 Joost™ (The Venice Project™)

From the Venice project we tested the software beta versions 0.7.2 and 0.7.3.1. There are no major differences between the two, except 0.7.3.1 was the first version released under Joost. The main effect caused by this was on the UI, where references to Venice project were removed. Also some minor bug fixes had been done related to RSS and chat functionalities.

The application appearance is the most polished from all the software tested in this context, even though it is still in beta phase. This alone gives rather strong impression that there is money involved in this project. The UI itself is transparent, floating on top of the screen so that video running underneath is shown through the UI elements.

Unlike any of the other applications tested, Joost client contains integrated “social networking” features like RSS feed reader and chat functionality. User can add his own RSS feeds to Joost, and feed titles will be shown as a running banner on top of the running video. Chat is channel-based, reminiscent to IRC. It enables people viewing a channel to chat with other people on the same channel.

Video can be viewed either in full screen or window mode, so that desktop actions can be performed during watching. The actual viewing of the video acts as a ordinary TV with recording: the stream can be watched from the beginning to the end without fast forwarding. If watching is interrupted, the program being viewed can be resumed from that point later.

Some minor problems appeared while testing. Less popular channels had a bit longer initial buffering delay when starting to watch. Sometimes the stream got a bit jerky when the UI elements were shown on top of the video. This occurred with both tested versions.



Figure 9: Joost™ client [17]

6.3 Server (broadcasting) software

This chapter is about the server (broadcasting) software that we were able to test at least some extent.

6.3.1 SopServer

SopServer was tested on Windows XP (Figure 10). Linux command line and GUI versions of SopCast are also available, but these were not tested. The Linux GUI version is available as an RPM package (Red Hat Package Management), but was not installable in our test environment (Suse Linux 10.1), probably due to invalid package format. This may well be only operating system dependent, so it may be possible to install the GUI version on some other Linux distribution using RPM.

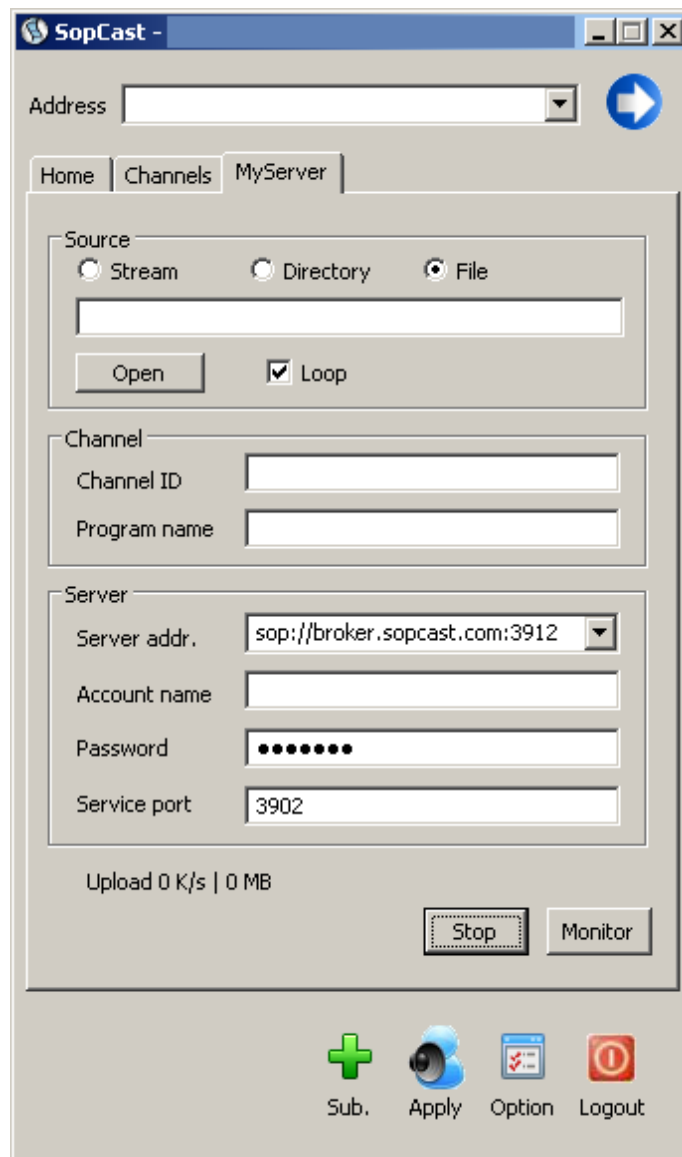


Figure 10: SopCast server

SopServer supports both streaming local media files and mms:// streams from an external source. Local files can be streamed as a single file or as a batch from a directory. Before broadcasting content, user is required to register to SopCast. Also the channel must be registered and added to one channel group in the SopCast service, in order for the viewers to locate the channel. SopCast allows single user to broadcast multiple channels, after each channel is registered, in contrast to TVU (see chapter 6.3.4).

SopCast web page offers a nice walk-through for starting the stream, so there is nearly no threshold for user to start streaming. After the channel is registered, the combined SopCast client / broadcasting application is used for broadcasting. Setting up a broadcast is very easy: user must enter a channel ID obtained during the channel registration, then choose the content (file, directory or external stream) and start streaming. After starting the broadcasting the channel was showing on the client channel list a couple of minutes later.

6.3.2 RawFlow

We managed to get 60-day trial license for RawFlow client and server software upon request, but the server / full system testing was not possible when considering restricted resources and the amount of time associated with the project. The RawFlow ICD testing would require larger environment consisting of several IP subnets divided by routers in order to provide better understanding of ICD technology.

The RawFlow server was tested on Windows XP, but also Linux version test was attempted. The Windows version installation was rather straightforward using the installation wizard. Linux version is actually a server daemon that needs to be run as a system service.

The server software is actually a mechanism to enable ICD technology for media distribution, that is, activates the actual P2P functionality. This means that an external media server needs to be used to provide the stream to the network. External stream server needs to provide mms:// or http:// stream.

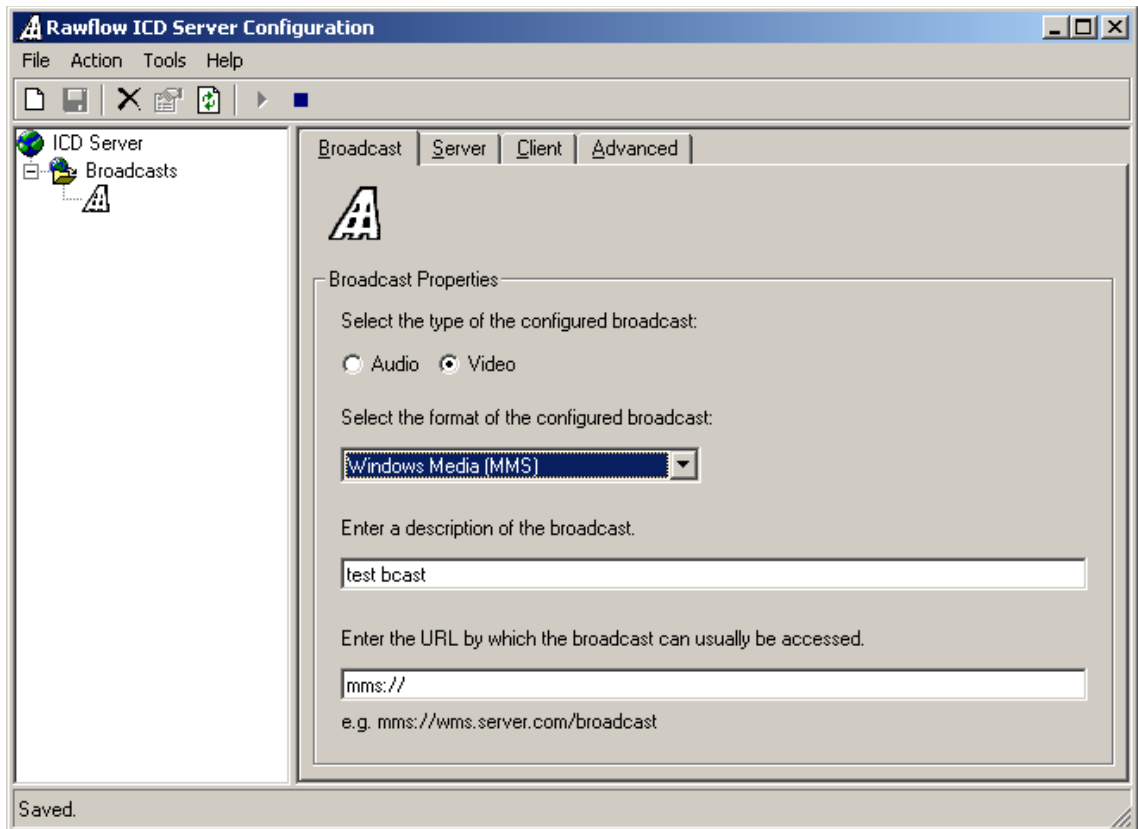


Figure 11: RawFlow server

While Windows version has a desktop GUI (Figure 11), the Linux and Mac versions can either be run from command line or via web interface, which was not tested due to insufficient time and instructions. The manual configuration is rather time-consuming. It may well be that the web interface allows similar usage than the Windows GUI.

The Linux command line version requires manual modification of some XML (Extensible Markup Language) files before broadcasting. On Windows version the XML files are “edited” via the GUI, and the files are created automatically on user actions. These XML files are used for service configuration and need to be published on service provider web page, in order to deliver them to clients.

The GUI is rather easy to use, after figuring out what goes where. Basic usage is as follows: user needs to input the source stream address and address for the web content to publish, as well as the local content directory where web content is published.

In our tests we were not able to observe the P2P capability of RawFlow. This was probably due to the LAN-intelligence of RawFlow [13]. Because the stream source resided in same LAN as our clients, both clients were able to get the stream from the original stream source instead of relying on peer-traffic.

6.3.3 PeerCast

PeerCast broadcasting is not so easy compared to receiving and listening a stream, because of several applications are needed. For example to broadcast MP3 stream you need to install WinAmp and Shoutcast DSP in addition to PeerCast application itself.

Amount of different softwares required for broadcasting might be the biggest challenge to some users. To ease this pain there are quite good 3rd party instructions¹⁰ available in addition to instructions by PeerCast.

During tests and when trying to broadcast behind NAT our channel was shown on the Yellow Pages¹¹, but people could not join in. NAT requires correct port forwarding in order to enable this. There are no warnings or messages shown regarding this.

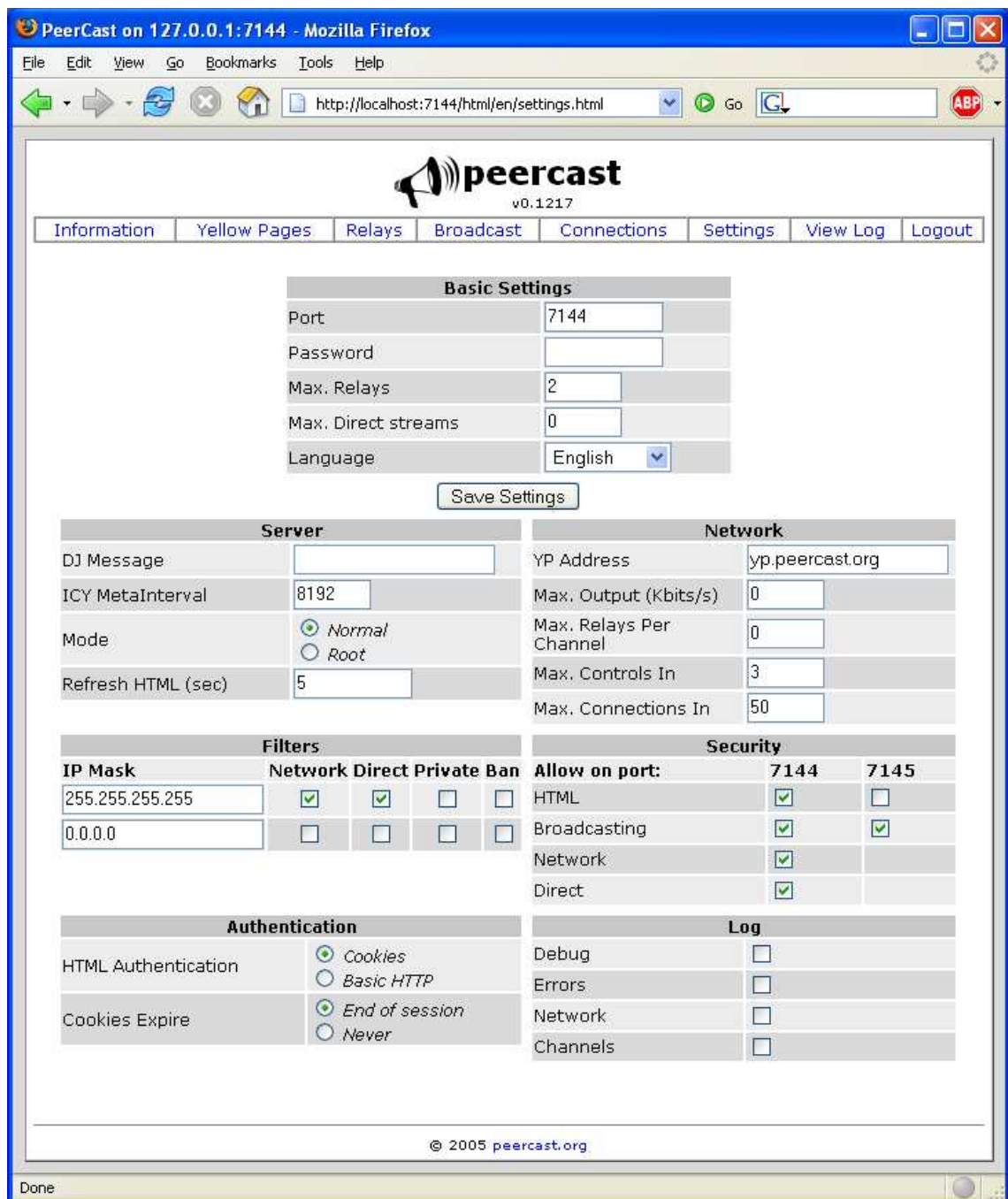


Figure 12: PeerCast browser UI

From the yellow pages you can see last played tracks and listener amounts by hour. You really feel like you had your own radio station, because of all the application windows open. On other tests we were also able to advertise our broadcast on Yellow Pages, but the actual broadcast could not be received.

6.3.4 TVUNetworks

The TVU server / broadcasting software acts as actual content streaming server. On Windows platform usage is straightforward and easy: using the GUI user must choose a file to stream and a network interface to stream the content to.

Prior to broadcasting, user is required to register. After registration a channel ID is allocated for broadcasting the content. As a restriction only one broadcast per channel ID is possible. So if wishing to broadcast several channels, a user account must be created for every channel. Whether this is only the case for amateur broadcasting, remains unresolved.

Whereas the Windows server is used for broadcasting users own content, the Linux server software is used for only relaying the stream from an external source (a http:// or mms:// stream server). The software does not have any user interface, but required parameters, which are address of the stream server and local interface to send the stream to. These are entered on the command line and server software is started. User was required to calculate a hex value of the channel ID before broadcasting, because the software required channel ID to be entered in this form for sending. The purpose for this transformation instead of directly using the decimal value acquired as channel ID remained unclear to us. This is probably a “feature” in the software.

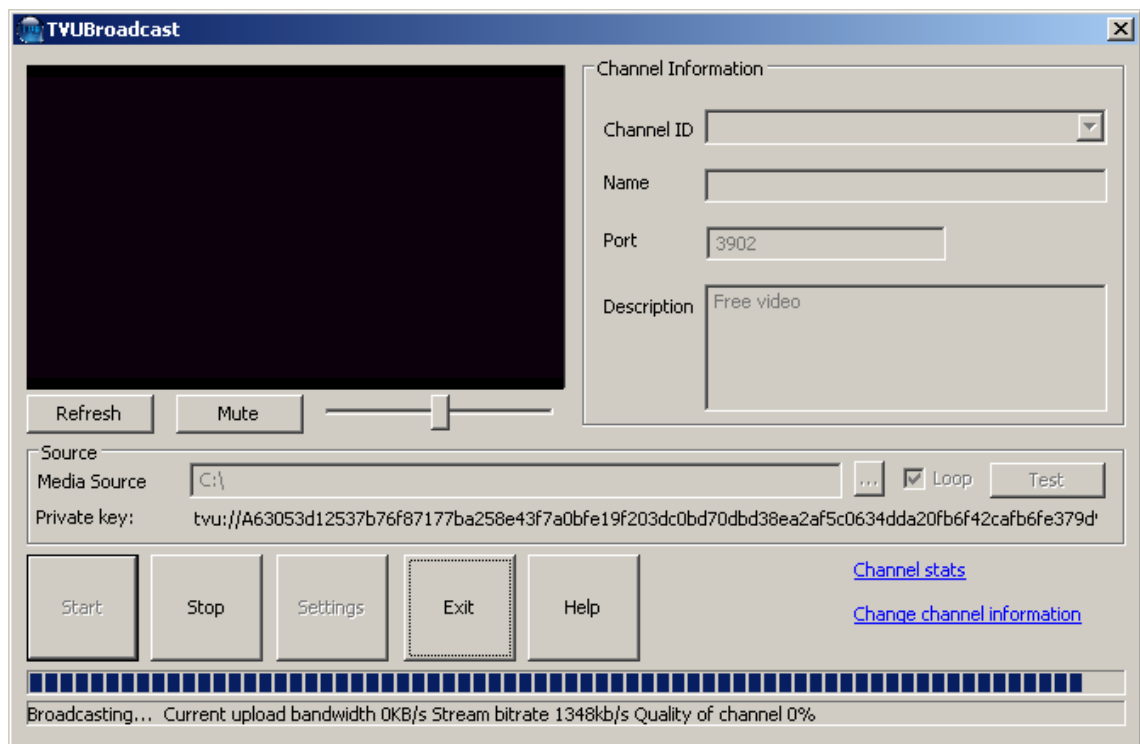


Figure 13: TVU server

By default the broadcasting software was using TCP port 80, so basically stream sent out is HTTP. Using a web browser during the streaming on the server end cuts the stream so dedicated server should probably be used for broadcasting.

6.3.5 Octoshape

We did not have possibility to test Octoshape server software, because of software unavailability.

6.3.6 Tribler Streaming

Linux server had to be built from source code. After a successful build we were not able to use the server because the software crashed on every start.

6.4 Reference techniques

This chapter includes test results for the reference techniques that have been listed in chapter 4. Because these techniques does not have clear separation between client and server solutions we included them both in the same subchapter. In this context testing the reference techniques were considered of lesser importance than peercasting techniques. This is because references techniques testing is lot easier to perform by a single user and users may be more familiar with these than the peercasting software tested.

6.4.1 YouTube

During the tests YouTube worked fine without any problems. Because it compresses all of the videos to same format the quality of the video does not get any better from the videos that are in it, even though you upload one with higher quality.

Because of the resizing and its effects to the quality YouTube recommends that users would upload videos with the same quality that they use in playing, i.e. MPEG4 (Divx,Xvid) at 320x240 resolution with MP3 audio, 30 frames per second [21].

6.4.2 Bubblare

As mentioned earlier Bubblare does not have a size limit for uploaded files. On our tests we managed to upload a 800MB video clip, which is eight times larger than the limit on YouTube. There might be some kind of limitation intended, because pages state after uploading that reformatting video takes from 0,25 to 2 minutes, depending on server load. For our 800MB test file it took around 40 minutes. Uploaded files cannot be currently

removed by user, but there is a “mark as incompatible” functionality available for all users, so it may be possible to remove videos using this.

File uploading is done using HTTP and actual content seems to be located in Sweden, according to route tracing performed with traceroute. This is the case for all content, regardless of service language.

Overall experience on Bubblare was very amateur due to bad translations, for example Swedish language has been left on Finnish pages. This was expected because Bubblare is in beta stage.

6.4.3 Veoh

As mentioned before in introduction chapter Veoh do not have any limitations in file size, length or video resolution, which is achieved with P2P technology. Veoh website worked equally good as other reference technique websites with integrated Flash player to preview the content.

Recently automatic syndication functionality with YouTube, Google Video, and MySpace was added to Veoh. This enables user to only do a single upload to Veoh and the uploaded media is automatically submitted to the other services mentioned. Restrictions set by for example YouTube still remain, so this is not a way to bypass these restrictions. Syndication feature requires existing user accounts on each of these communities.

6.4.4 Zudeo

Zudeo worked fine, as expected. Why this was expected is that it uses the same engine Azureus 2 engine that have been proven to work fine. The UI addition compared to normal Azureus is very nice and takes the UI more closer to less expertized users.

7 Conclusions

At the moment peercasting is securing its position among users and the implementations are more or less unfinished or oriented to small group of broadcasters. Comparing to traditional unicast media transfer peercasting offers a lot of improvements concerning the bandwidth usage and server requirements. Although the improvements are great and help small organizations to build for example Internet radio to large audience rather easily, people probably unaware of the opportunities.

Compared to YouTube and other reference techniques peercasting solution is still far from being mature. A lot of work needs to be done for example in usability, in order to gain massive public support. People do not always want to or even can not download client or share their bandwidth for security reasons which prevent effectively spreading of these techniques. From the user point of view, JoostTM seems to be closest to achieving public acceptance.

For the casual user, the reference techniques examined allow easy to access and easy to use services to spread users content and share experiences in a community-like environment. But if wishing to professionally broadcast high quality content over the Internet to large-scale audiences, peercasting is the way to go.

References

- [1] Akamai, Akamai Media Delivery, 2007. Available http://www.akamai.com/html/solutions/media_delivery.html. Accessed
- [2] Nazanin Magharei, Yang Guo, Reza Rejaie, Issues in Offering Live P2P Streaming Service to Residential Users, 2006. Available <http://mirage.cs.uoregon.edu/pub/ccnc07.pdf>. Accessed 14.1.2007
- [3] RawFlow Ltd., Homepage, 2006. Available <http://www.rawflow.com/>. Accessed 21.11.2006
- [4] Pai, V., Kumar, K., Tamilmani, K., Sambamurthy, V., Mohr, A. E., Chainsaw: Eliminating Trees from Overlay Multicast, . Available http://iptps05.cs.cornell.edu/PDFs/CameraReady_164.pdf. Accessed 4.1.2007
- [5] Mol, J., J., D., Epema D., H., J., Sips H., J., The Orchard Algorithm: P2P Multicast without Free-riding, 2006. Available <http://pds.twi.tudelft.nl/pubs/papers/p2p2006.pdf>. Accessed 3.1.2007
- [6] Pai, V., Mohr, A., E., Improving Robustness of Peer-to-Peer Streaming with Incentives, 2006. Available <http://www.cs.duke.edu/nicl/netecon06/papers/ne06-streaming.pdf>. Accessed 14.1.2007
- [7] SopCast, Homepage, 2006. Available <http://www.sopcast.org/>. Accessed 27.11.2006
- [8] Alstrup, S., Rauhe, T., Octoshape – a new technology for large-scale streaming over the Internet, 2005. Available http://www.ebu.ch/en/technical/trev/trev_303-octoshape.pdf. Accessed 27.11.2006
- [9] Octoshape, Homepage, 2006. Available <http://www.octoshape.com/>. Accessed 10.12.2006
- [10] TVU Networks Corporation, Homepage, 2007. Available <http://www.tvunetworks.com/>. Accessed 14.1.2007
- [11] Wikipedia, TVUnetworks, modified 16:05, 31 December 2006, 2006. Available <http://en.wikipedia.org/wiki/TVUnetworks>. Accessed 4.1.2007
- [12] Jevnaker, I. S., RawFlow – using P2P to create virtual ”multicasts”, 2006. Available http://www.ebu.ch/en/technical/trev/trev_308-rawflow.pdf. Accessed 27.11.2006
- [13] RawFlow Ltd., ICD Technology Whitepaper, 2006. Available http://www.rawflow.com/pdfs/white_paper.pdf. Accessed
- [14] PeerCast, Homepage, 2006. Available <http://www.peercast.org/>. Accessed 14.1.2007
- [15] Tribler Streaming, Homepage, 2006. Available http://www.tribler.org/test_streaming/. Accessed 3.1.2007

- [16] Delft University of Technology, Tribler Streaming – A Quick Overviev v1.0.3, 2006. Available http://www.tribler.org/ibc/tribler_streaming_explained.pdf. Accessed 13.12.2006
- [17] Joost (TM) (Baaima N.V.), Homepage, 2006. Available <http://www.joost.com/>. Accessed 25.1.2007
- [18] Malik, O., Exclusive: Inside The Venice Project, Built On Mozilla, 2007. Available <http://www.netscape.com/viewstory/2007/01/13/inside-the-venice-project-built-on-mozilla>. Accessed
- [19] LvLLord, Tips, Tricks & Utilities, 2006. Available <http://www.lvllord.de/>. Accessed 14.1.2006
- [20] IceShare, Homepage, 2006. Available <http://wiki.xiph.org/index.php/IceShare>. Accessed 10.12.2006
- [21] YouTube, Inc., Homepage, 2006. Available <http://www.youtube.com/>. Accessed 14.1.2007
- [22] Bubblare, Bubblare, 2006. Available <http://bubblare.fi/>. Accessed 30.11.2006
- [23] Veoh Networks, Inc., Homepage, 2007. Available <http://www.veoh.com/>. Accessed 14.1.2007
- [24] Azureus Inc., Zudeo homepage, 2006. Available <http://www.zudeo.com/>. Accessed 14.1.2007
- [25] Octoshape, End User License Agreement, 2005. Available <http://www.octoshape.com/files/EULA.pdf>. Accessed 17.12.2006