NTNU
Innovation and Creativity

Multimedia Distribution Networks and the Aeetes Project

Svein-Magnus Bergan Sørensen

Master of Science in Communication Technology
Submission date: May 2006
Supervisor: Yuming Jiang, ITEM

Norwegian University of Science and Technology
Department of Telematics

NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY, MATHEMATICS AND ELECTRICAL ENGINEERING
DEPARTMENT OF TELEMATICS

MASTER'S THESIS

Student's name: Svein-Magnus Bergan Sørensen

Area of study: Telematics

Title: **Multimedia Distribution Networks and the Aeetes Project**

Project description:

Multimedia applications such as music and video over IP have become an indispensable part of the Internet. When providing a multimedia service to end users, it is essential that the service is easy to use and that is provides good quality combined with low cost. This imposes a range of challenges on multimedia distribution. A multimedia distribution network (MDN) is an overlay network used to distribute multimedia content to end-users in a cost-efficient manner with good quality of service.

The objective of this thesis is to research the current state of the art in existing MDN systems, and to make a comparison of the technologies and architectures used in them focusing on the advantages and limitations of the various systems. Based on this comparison the thesis should then analyze possible MDN-architectures and suggests a design for a new and improved MDN-system.

This suggested architecture should allow for easy extensibility and simple modifications as it might become a base system upon which further work in the field will be built.

Start date: January 16, 2006
Deadline: June 12, 2006
Handed in: May 26, 2006

Department: NTNU Department of Telematics
Supervisor: Professor Yuming Jiang

Trondheim, 26th of May, 2006

Yuming Jiang
Professor

PREFACE

This report is written as a master's thesis at the Norwegian University of Science and Technology (NTNU), Department of Telematics, between January and May 2006.
The writing of a research thesis is a requirement during the final year for a student to be awarded the Master of Science degree in Telematics by NTNU.

This thesis is the first in a range of work to be performed in the field of Multimedia Distribution Networks at NTNU, in what will be known as the Aeetes Project. It contains the preliminary research into the field, and makes an architecture proposal intended to become the basis for future work and development in this field at NTNU.

I would like to thank my supervisor Professor Yuming Jiang at NTNU for all of his invaluable help and support during my work with this thesis, and I am very grateful to him for suggesting this thesis to me and for supervising my work.

Svein-Magnus Sørensen

Trondheim, Norway

May 2006

ABSTRACT

This thesis provides an introduction to the field of Multimedia Distribution Networks (MDN) and is written as the first part the Aeetes Project. It compares potential MDN architectures and technologies, and makes a proposal for a suitable IMDN architecture based on it.

The thesis defines the concepts of MDN and IMDN, and describes the current state of the art of MDN technologies and implementations. An Internet-based Multimedia Distribution Network (IMDN) is a type of overlay network, comprised of a cluster of servers on the Internet which is used to distribute multimedia content to end-users in a cost-efficient manner with better quality of service compared to regular distribution models.

It also makes a comparison of the technology and performance of various existing MDN architectures with focus on their advantages and limitations. Based on this comparison the possible architectures are evaluated and a proposal is made for an Aeetes implementation candidate. The suggested architecture is focused on low cost and simple deployment, and is designed to be used in both academic and commercial settings. It also attempts to be extensible and to allow for simple modifications as it might become the base system upon which a range of further work in the field of Multimedia Distribution Networks is built.

TABLE OF CONTENTS

TABLE OF CHARTS AND ILLUSTRATIONS

ABBREVIATIONS AND DEFINITIONS

Aeetes	(1) In Greek mythology the king of Colchis, brother of Circe and father of Medea (2) The project started with this report to design a multimedia distribution system
ALAN	Application level active networking
AVP	Active Virtual Peers
CDN	Content Distribution Network / Content Delivery Network
Clearinghouse	An institution or system that collects and distributes information.
CODEC	A matched pair of algorithms for Encoding and Decoding content, usually for compression.
container-format	A file format that allows a range of different codecs to be used in the same type of file.
DHT	Distributed Hash Tables
downloading	Getting a file from a remote location over a network to be stored on a local computer. Playback can usually not begin until the entire file is downloaded.
DRM	Digital Rights Management
EPG	Electronic Program Guide
IFC	The Independent Film Channel
IMDN	Internet-based Multimedia Distribution Network
IP	Internet Protocol
ISP	Internet Service Provider
last mile	The final bit of communications into each personal home from the nearest interchange point.
localized	Something oriented locally in relation to a physical location (either geographically or topographically in the distribution network)
MDN	Multimedia Distribution Network
OMN	Open Media Network
ON	Overlay Network
OSTG	Open Source Technology Group
overlay network	Is a virtual network of nodes and logical links that is built on top of an existing network implementing a new service that is not available in the network underneath.
P2P	Peer to Peer
PEX	Peer Exchange
PVR	Personal Video Recorder
QoS	Quality of Service
Quality of Service	The performance specification of a communications channel or service.
Quicktime	Media file container-format from Apple (Natively using the Sorenson codec)
Real	A streaming media codec and file format from RealNetworks
SCH	Spatial Content Hashing
set of data	A data set is the files that make up a single entity of content, for instance a software application or a multi-part package of video or audio like a TV-series season or a full album.
SON	Service Overlay Network
streaming	Playing content directly from a remote location with the data being served in real-time over a network as it is required with little or no waiting time for the user.
swarm	The group of peers in a P2P or Grid network currently exchanging the same set of data.
transcoding	The process of converting a media file from one format to another.
TV	Television
URI	Universal Resource Indicator
Video on Demand	The ability to select videos from a central server and receive it at a presentation device upon request, used in entertainment and videoconferencing.
viral video	Small video clips that are spread like wildfire by viewers due to interesting or funny content.
VoD	Video on Demand
WMA	Windows Media Audio – A container file-format and a series of codecs for Audio
WMP	Windows Media Player / Generic for Windows Media file-formats WMA and WMV
WMV	Windows Media Video – A container file-format and a series of codecs for Video

1. INTRODUCTION

1.1. Background

Multimedia applications such as music and video over IP have become indispensable parts of the Internet. When providing a multimedia service to end users, it is essential that the service is easy to use and has good quality combined with low cost, and this imposes a range of challenges on multimedia distribution.

An Internet-based Multimedia Distribution Network (IMDN) is a type of overlay network comprised of a computer-cluster on the Internet that is used to distribute multimedia content in a cost-efficient manner and with better quality of service compared to regular distribution methods. The work that begins with this report is the first part of a continuing effort to develop such a network, and it has been named The Aeetes Project.

The objective of the Aeetes Project is to design a platform for multimedia distribution over the Internet. The platform shall provide a simple interface between end-users and Internet multimedia services and be an efficient way to distribute multimedia to end-users.

The name "Aeetes" was picked for it origins in Greek mythology. King Aeetes of Colchis was the father of the princess Medea known from the classic story of Jason and the Argonauts. The relation to this project is that the name Medea is pronounced very much like the word "media", and delivering media is exactly what this project is all about!

1.2. Problem definition

The object of this work is to perform research into the current state of the art in MDN systems, and to write an introduction to this field for future students. In addition a comparison of the existing MDN technologies and architectures is performed, focusing on the advantages and limitations of the various systems. Based on the comparison, a range of possible MDN-architectures are analyzed and a single design is chosen and suggested to become a possible base for the new MDN-system to be developed through the Aeetes Project. For this reason the suggested architecture should be extensible and adaptable for future modification, as requirements might change while the project progresses.

1.3. Limitations in this work

The major part of this work has been to locate and study information on the current state of the art in Multimedia Distribution Networks and the theory behind the existing services, in addition to analyzing the possible architectures and making the architecture proposal.
Since this project has only been performed over a single semester, it has not been possible to perform any simulations or implementations of the suggestions proposed so they are built on an analysis of the potential architectures and not on undisputable hard facts.

1.4. Report outline

This report looks into the current state of MDN-technology and gives a thorough introduction to and overview of the existing MDN services and technologies currently in use. In addition it proposes a new architecture to be used for future work in this field within the Aeetes Project.

Chapter 2 provides a definition of what the term Multimedia Distribution System should cover and what types of MDN networks exist. Also it provides an overview of existing systems and the services they currently offer.
Chapter 3 gives a detailed description of the technologies which may be used to build a MDN, and a comparison between these to select the best one for the proposal in chapter 4.
Chapter 4 details the architecture proposal to be used in the Aeetes Project.
Chapter 5 is a summary of the suggestions and conclusions made in this report.
Chapter 6 provides and overview of the required future work to be performed in the Aeetes Project, which when completed will result in the realization of a full MDN-system.

Finally in **Chapter 7** there are a list of the references and resources that have been used for researching and writing this report. Unique words marked with [square brackets] throughout the text are references that can be looked up in this list for further information.

2. MDN DEFINITION AND STATE OF THE ART

How does a Multimedia Distribution Network operate, and what are the capabilities of the current technology and operators in this field? This chapter gives an overview over this as well as a detailed definition of the terms MDN and IMDN.

2.1. What is a multimedia distribution network

In short, a multimedia distribution network is for digital multimedia on the internet as what a grocery store is for food in the real world, meaning it is a localized shop where you can find and get products from a large range of providers in an easy manner.

2.1.1. CDN: Content Delivery Networks

In many ways a Multimedia Distribution Network is a subset of the more generic Content Delivery Network, which helps popular internet-sites distribute their traffic both geographically and over multiple servers.

[WhatIs.com] defines content delivery as follows:

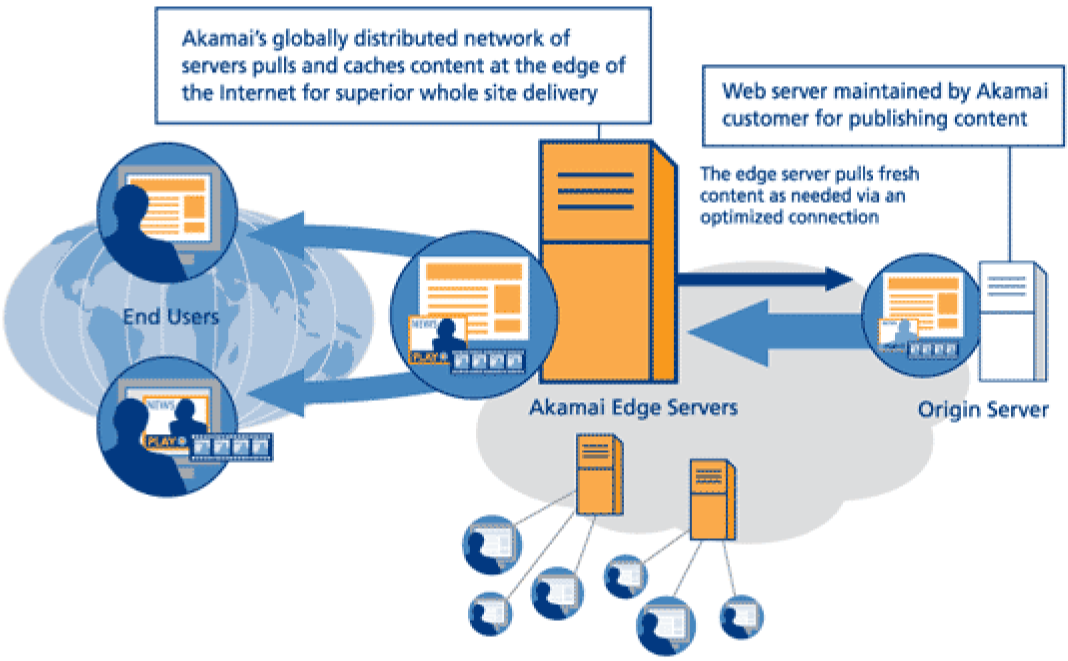
> On the Internet, content delivery (sometimes called *content distribution*, *content distribution delivery*, or *content caching*) is the service of copying the pages of a Web site to geographically dispersed servers and, when a page is requested, dynamically identifying and serving page content from the closest server to the user, enabling faster delivery.

This gives that a CDN is a network dedicated to the task of content delivery, however it does in no way change the way one accesses or uses information on the Internet. The usage of a CDN is for all intents and purposes invisible to the user and simply has the goal of providing a user with digital content in an efficient way by serving the content through a range of localized storages rather than having just a single massive computing centre.

The advantages are obvious as the CDN in addition to the improved geographical proximity to its users can provide faster and cheaper connections for everyone by spreading the traffic across a wide range of locations instead of having a single point of failure that could be prone to network congestion that would affect all users, and not to mention that a single massive computer centre would be extremely expensive, both due to power-requirements and the enormous amounts of bandwidth that would be needed. Unfortunately it is expensive to deploy a CDN as well, so instead of every provider setting up their own networks there has grown up a separate industry around providing CDN-services to serve the heavy traffic of popular websites.

There are many different approaches to create such CDNs, as are shown by the existing services from Akamai, Digital Island and others, but this report will not go into the details here as they are better described elsewhere, for instance in [CDN-PR] and [CDN-Use].

Illustration 2.1: The Akamai CDN



This is a high-level illustration of how the load distribution of the Akamai content delivery network function across the internet. Akamai's service is a type of replication network.

2.1.2. MDN: A CDN improved in relation to users

The CDN principle of distributing content geographically to bring it closer to the end-user is a big part in the theory of Multimedia Distribution Networks. However a MDN in the definition of this project is much more than just a subset of a CDN focused on multimedia. First and foremost a MDN is also a clearinghouse that provides consumers with a single point-of-presence where they can locate and retrieve multimedia content from a wide range of providers.
This way an MDN does not only give the large content-providers an efficient delivery platform for their content, but is also a localized repository with a broad range of content from multiple providers where users can retrieve anything they want through a unified interface on low-delay connections. This makes the entire transaction of multimedia consumption much simpler, faster and cheaper for both the end-users and the providers.

This also means that one common type of MDN is the much hyped Video on Demand [VoD] services that have been gaining increased popularity recently. A VoD service is a service where a user may select any desired programming from a large catalogue and either stream it directly to some kind of presentation device (TV, Computer, Mobile phone), or order it for download so that it will be ready to watch at some later time. Similar services like [iTunes] exist for music, and there is also [Audible] for other audio content, but a full MDN is usually not required to set up these services due to the lower bandwidth and storage requirements of audio compared to video.

Another advantage that MDNs have over general CDNs is that when the network clearinghouse is in place, and when it is where users are already going for many of their multimedia needs, an obvious addition is to give consumers the chance to provide content of their own to distribute through the network. This is something that not only empowers consumers and can allow them to publish content on an equal footing in competition with the large media-houses, but it will also expand the amount of available content greatly and thus create a positive feedback effect for the MDN as a whole, which means increased usage and thereby exposure to each of the individual content providers represented.

While this user-publishing feature will not be available within all MDNs it is clearly a feature that will have great effect in the future, especially for the democratization of multimedia publishing on the Internet.

2.1.3. Hierarchical content deployment issues

The basic mode of operation for a Content Delivery Network is to copy content from a master server onto a set of distributed delivery servers that in turn deliver the content to end-users. For a Multimedia Distribution Network this model will need to be improved somewhat due to the large storage-requirements when serving video or other high-capacity content. One of the challenges caused by this is to decide the functionality and architecture of delivery systems.

If one were to focus only on the cost of storage-capacity, the cheapest solution would be to have a single data centre with an immense amount of bandwidth that performed all required tasks, but if focusing only on the cost of bandwidth the cheapest solution would be to store the entire catalogue at every user's home-location and have their systems do any required work. However in the real world it is needed to factor in both aspects to find the optimal placement and size of servers to serve the desired amount and selection of users. [Tradeoffs]

Depending on this relation between bandwidth and storage-cost the optimal distance from each user to the nearest data centre can be very long, which is impractical because of high transmission delays and bandwidth usage which are the QoS parameters that would make an MDN desirable in the first place. A compromise can be to not store the entire content catalogue at every data centre, but instead only distribute the content that is popular in the various regions. This will reduce the storage-requirements for each location since they only need to store part of the catalogue, and therefore the centres can be moved closer to the end-users and allow for good QoS while still providing on-demand access to the entire catalogue by transferring any rarely requested files from a master storage containing the entire catalogue when a request for such content is made.

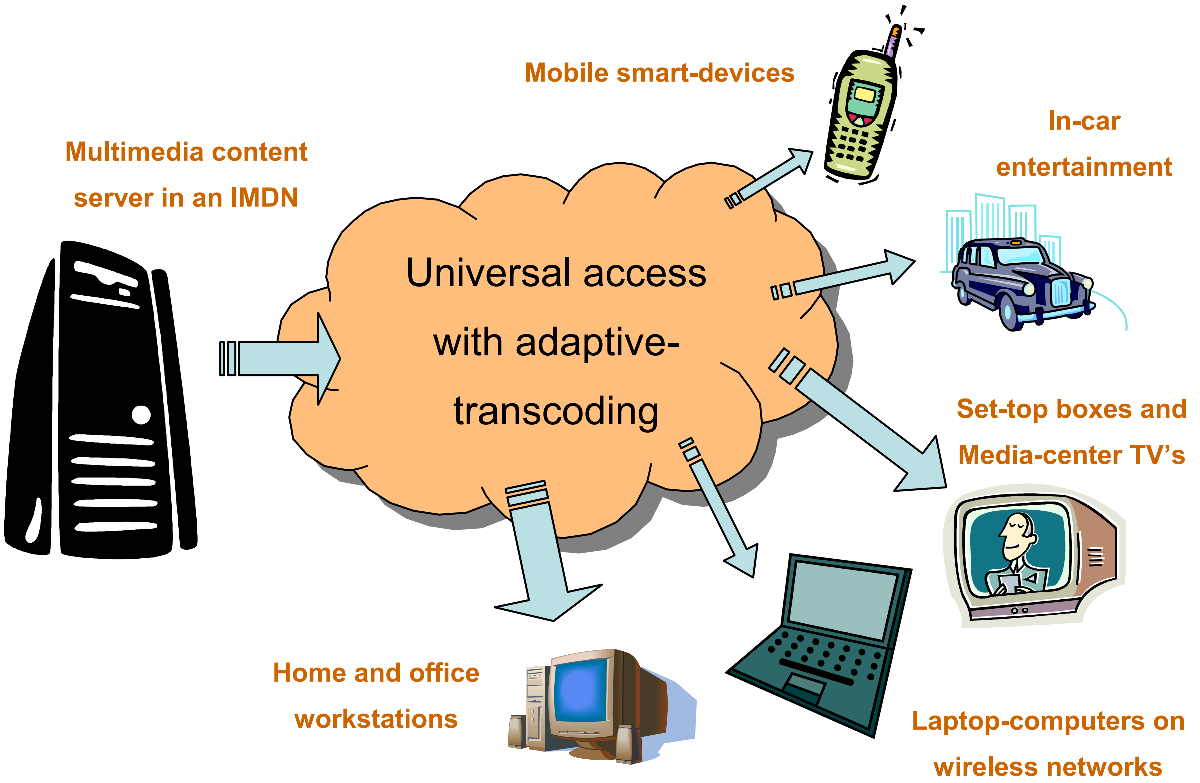
While these matters are still concerns today, hopefully future improvements in either transfer or storage technology will make them a thing of the past. They might even be already, as is shown by the Peer to Peer system called Kontiki which is used by the Open Media Network for their IMDN. [Kontiki].

2.1.4. Transcoding and adaptive streaming

Another technical aspect of Multimedia Distribution Networks that is important, but not very common yet, is transcoding of content on demand and having graceful degradation of the streaming delivery. The theory is that the MDN should be able to provide a multimedia-stream suitable for any kind of device and bandwidth based on a single "master copy" of the content in the archive. This problem has two main sides to it, firstly to provide a format suitable for any kind of device, and secondly to scale the stream to fit the available bandwidth.

Transcoding is the technique to turn one copy of a file into a different format, and doing this on demand and in real-time for every user that requests the file would be an advantage for an MDN. However this would require a great deal of the processing power in high-capacity delivery systems unless they are equipped with dedicated transcoding hardware, so it will be very costly to provide such a service with current technology. Providing the suitable format can be easily solved in other ways than transcoding however, for instance by having the various formats prepared in advance. This however will require a lot more storage-capacity from the delivery systems than storing a single copy of the file and then transcoding it, nevertheless depending on the cost-factor between this storage-capacity and the processing-power required for transcoding it can be an adequate and realistic solution. This technique is already being used in some existing MDNs like Google Video that provides all the free content in formats suitable for the Apple iPod Video and the Sony Playstation Portable in addition to their native flash based Google Video format.

Illustration 2.2: Universal access with Internet-transcoding



This illustration shows how transcoding and adaptive encoding in the Internet can allow a single media-stream to be received and viewed by a range of different terminals in various environments.

An optimal solution to the format issue is to combine these two techniques by transcoding the content whenever a new format is requested for the first time, and then storing the result either permanently or temporarily for future deliveries. This ensures that only content that is actually accessed in a given format will use storage-capacity for that format, while avoiding the major overhead caused by transcoding the content for every request by every user. This also means that adding a new format will only require the addition of the relevant codec used for transcoding, and then the content will sort itself out as requested by clients.

Scaling a video-stream to the available bandwidth is a completely different kind of problem, but much work has been done on this topic, for example by groups working to design true multicast systems. The simple solution for unicast-based MDN is to simply store or transcode a stream to fit the available bandwidth based on specifications in the request. While simple and available today this solution is very limited, especially since it does not take into account changing network conditions, and it cannot be used for multicast systems where the same stream will be received by multiple clients with varying demands.

A solution to overcome this could be to use an adaptive streaming format that degrades gracefully with the changing conditions and requirements. This means that the MDN should transmit a stream of maximum quality to all users, but with some kind of priority marking on each packet so that intermediate routers can drop them according to the marking if there is not enough capacity on a link to transmit the entire stream. The result will be that end-users on or behind a low-capacity link will receive a viewable stream in real-time independent of the network conditions, but the quality of the stream will be adjusted by the network to the level possible or required at any given time. This allows the same multicast-stream to be received by both power-users on a high-capacity wired link expecting very high-quality, and by users on a spotty low-capacity mobile link receiving the stream on their mobile phone while travelling. [Adaptive]

2.2. Types of MDN networks

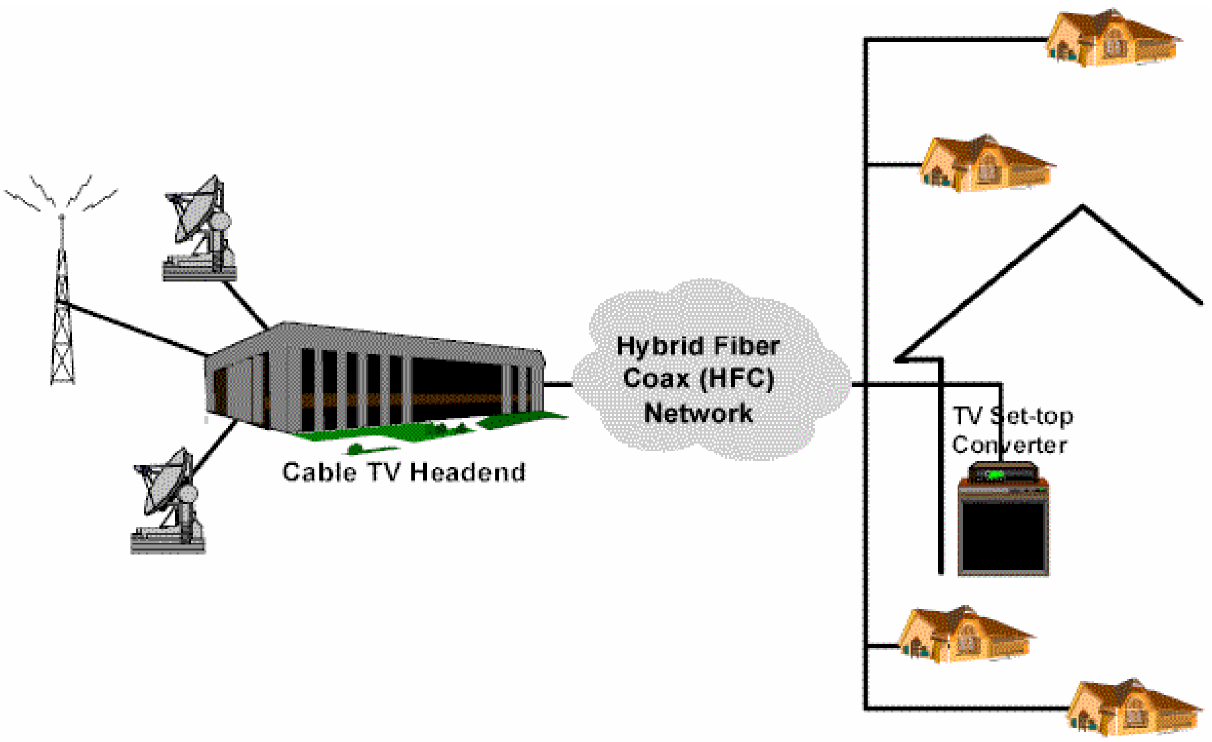
Just as with CDNs, there are many different ways to implement a Multimedia Distribution Network ranging from just setting up simple data centres at various places on the public Internet, to building a full delivery network from scratch. This chapter provides an overview of the various types of networks and their suitability for the architecture to be suggested in this thesis.

2.2.1. Dedicated end-to-end delivery network for media

The cable television networks that were first developed in 1948 can be considered to be the original kind of MDN, but since it used to be a one-way technology with programming decided by the network, its origins do not really fit into the definition of MDNs detailed above [Cable]. However the cable-networks of today has evolved to become mostly bi-directional and allow both the transmission of Internet-traffic and dedicated VoD next to the normal TV-channels. This means that most of the current cable-networks can be considered to be MDNs, which is especially true for those services that use a set-top box like those provided by HBO's OnDemand-service [HBO] allowing a user to request content, and for those using an IP-VoD service supplied over a cable-Internet connection.

Currently it is almost only set-top-box systems that belong to the class of dedicated end-to-end MDNs, since these are the only networks where both the distribution-servers and the end-users are connected to the same private network. However many of the Internet based services such as CinemaNow and Vongo have strong ties with the movie and cable industries, and while it is not yet common for these and most of the other similar services to have their own distribution centres inside the cable networks to get better proximity to users, this can become reality in the future. This is especially true if the demand for and usage of these services increase greatly, something that can make a range of distribution centres inside the various delivery-networks a necessity, making these services too into true end-to-end distribution networks where they and their partners together have control over the entire network through which the content is delivered.

Illustration 2.3: Typical cable-TV network



This is the layout of a typical cable-TV network capable of providing television, internet and phone services. The head-end station has a range of high-capacity communications links and receivers for connecting end-users to the content and services that are provided over a Hybrid-Fiber-Coax network. The households often need to use special set-top boxes or similar converters to access some or all of the services.

Another kind of dedicated end-to-end distribution networks are those built by satellite operators and the likes of Disney's MovieBeam [MB], which instead of using cable or the Internet provide regular content-updates using radio-transmissions that are picked up by advanced set-top-boxes with antenna that act as small distribution centers inside every home. The box will at any single time contain about 100 movies that the user can choose from, and the selection will change over time by deleting some movies and getting new ones over the air. While such systems do not provide its users with unlimited access to the full catalogue like those mentioned above, they do provide a large enough selection to cover the entertainment needs for most home users, and as the capacity of storage technology improves over time the amount of content that can be made available locally will increase with it.

Despite the obvious advantages from having full control of the entire delivery network, the high cost and amount of work involved in deploying a dedicated transport network for multimedia makes it very unlikely that the deployment of new networks will be attempted in competition with the existing ones. The common broadband connections to the public Internet makes such a deployment even more unlikely since they can be used to cheaply deliver any form of data directly to most households, removing the need for other networks altogether. The dedicated end-to-end delivery networks will for these reasons not be considered any further for the proposal in this report, and it will instead focus on solutions making use of the public Internet to build the Aeetes Multimedia Distribution Network.

2.2.2. Dedicated transport networks using Internet for the last-mile

For MDN operators that do not own their own cable-networks or are allowed to place servers inside existing cable-systems, their distribution architecture will differ slightly from the examples with full end-to-end control.

One can through the use of leased lines relatively cheaply design a dedicated network to provide each server or data centre with the content required for that area, but the big problem is getting the content out to consumers through what is known as the "last mile". As explained above it is not viable to roll out a new wired network to every house, and the radio-networks used by services such as Disney's MovieBeam can only provide consumers with a limited selection of content, which partially defeats the purpose of building a MDN in the first place.

A simple solution to overcome this problem is to make use of the broadband Internet-connections and 3rd Generation mobile networks that are getting increasingly common everywhere.

This will allow the provider to potentially have full control of all the distribution internally in the MDN. Also by having a large number of data centres, any local Internet or mobile cell outages will only affect the small portion of users connecting to the affected servers. In addition, even the affected users will not always end up with black screens due to such outages as they might be able to connect to other data centres instead, though often at reduced speeds or with lower quality than what their local data centre would be able to provide.

While this solution is potentially viable and possibly very good for commercial MDN-operators, the solution to be suggested in this thesis should also be available for low-cost deployment in an academic setting, meaning that even using leased-lines are too expensive to be realistically considered. Therefore also this architecture will not be further considered in this thesis.

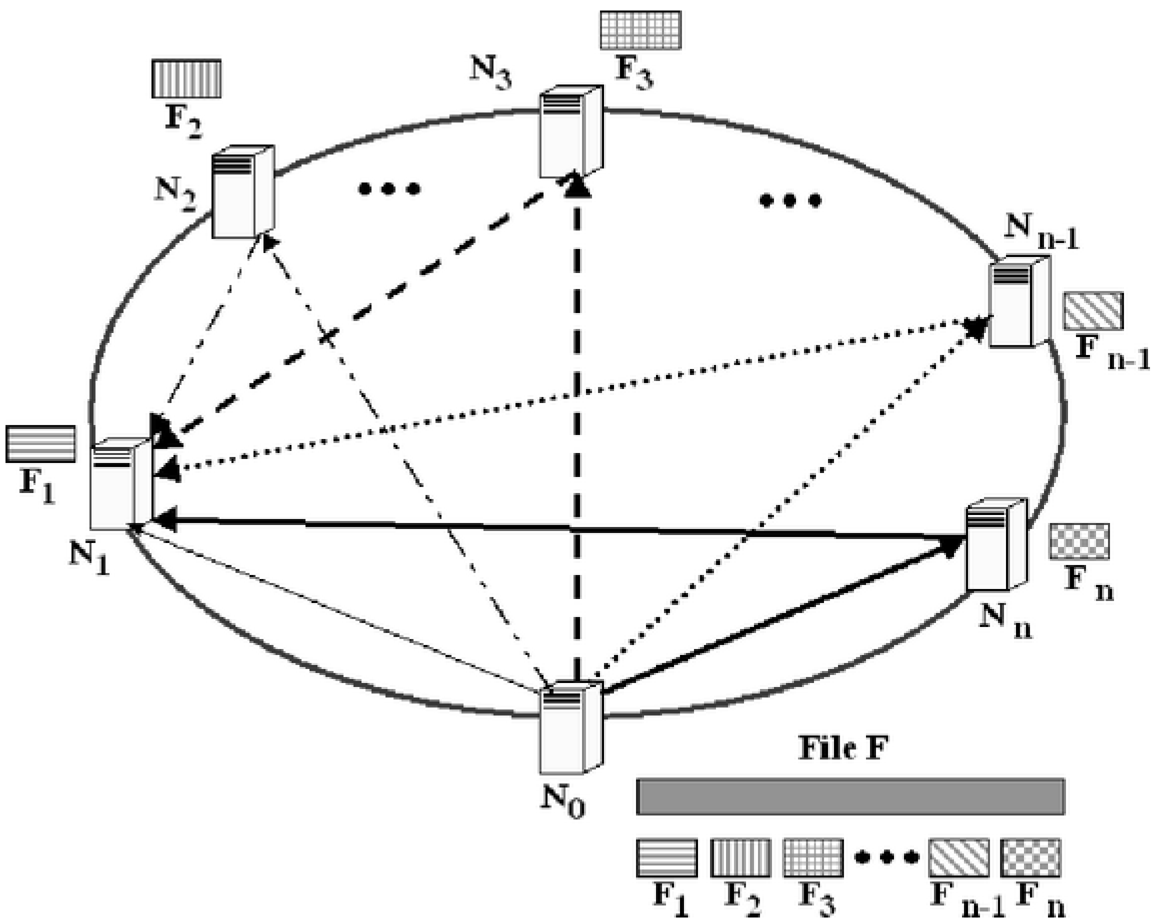
2.2.3. Using the public Internet only – Internet-based MDNs

A different but much easier and cheaper way to implement a Multimedia Distribution Network is to spread out a series of server-farms close to the end users in what is known as an overlay network, and simply use the public Internet to update content in the data centres and for delivering it to end-users. However due to congestion and other matters that can cause problems with the public Internet, the distribution part of this design requires a bit more thought than the equivalents in other solutions where full control over the entire network is possible. Nevertheless the problems are not insurmountable. It is for instance possible to use the FastReplica algorithm described in [FR] as this will distribute the bandwidth requirements evenly across all the servers in a distribution group. IP multicast is another technology that can aid both distribution and delivery to make this kind of MDN possible. Even a form of P2P can be used both for distribution and delivery as is shown by the Kontiki-architecture that is introduced later in this report.

Both for the Internet-based type of MDN and for the dedicated transport network type described in Chapter 2.2.2, it is very important that the data centres are located at or near important network hubs and interchanges. Smaller countries could in such a way be served with just a single large data centre by placing it at the national network interchange. In larger countries one might want additional data centres at other important interconnects, and one could also locate data centres for the most popular content directly on the backbone networks of the larger ISPs as this will improve service for the largest groups of users.

This solution is most interesting for the Aeetes Project since it can simultaneously allow for very low-cost internal deployment on a single LAN and for a major worldwide distribution network based on the same general architecture. This type of overlay network will be referred to as an *Internet-based Multimedia Distribution Network* (IMDN), but the term is in this report used somewhat interchangeably with the more general form *Multimedia Distribution Network* (MDN) since the other types of MDNs are not discussed in detail.

Illustration 2.4: An Internet-based CDN using FastReplica



This illustration show how multimedia distribution using the FastReplica protocol can efficiently distribute content from node N_0 to node N_1 by using intermediary servers on the public Internet. The content will through such distribution be accessible to end-users in the all the areas covered by each of these servers.

2.2.4. Types of end-user clients

While the distribution of the data to the local sites is a very important aspect of a MDN, it is also relevant what a user must do to receive the media. Currently there are only the following practical interfaces available and in use.

2.2.4.1. Public Internet websites

Most Internet-based services provide the primary interface to access content at a public website. Some of them require registration to use, but often they allow free access to browse and search through the content to, after which the content may be streamed or downloaded for viewing at the local terminal.

The major advantage of websites is that almost every Internet user in the world already has access to the necessary applications, i.e. a web-browser and a media player. This means that the threshold for checking out the service is very low since it does not require one to actively do anything other than locating the website in the first place. However for very high-traffic sites a separate content distribution network for the website may be needed in addition to the MDN that delivers the actual content, which complicates the implementation a bit.

Illustration 2.5: The web interface of Google Video

This is a screenshot from the web interface of Google Video. The flash based media-player allows the same kinds of controls that people expect when playing a downloaded film. On the top of the page are the account settings, category selections and a search field. Information on the currently playing clip can be found in the right panel and just below it is a selection of related clips.

2.2.4.2. Local computer applications

A few services like the Open Media Network [OMN] only provide access to its content through a standalone application that must be downloaded and installed on the local computer. Through this application one can search a range of different TV-channels and shows for streaming or downloading. Such an application can take any shape and form and there are no limits to which options it can provide.

Other services like Google Video provide both the web interface described in Chapter 2.2.4.1 for searching and previewing video, but give the option to use a local application for playback of downloaded content. In addition they do along with many others require a local application in addition to the website for content uploading and transcoding. [GV]

Illustration 2.6: The Open Media Network application

This screenshot shows the Program-guide in the Open Media Network application. From this menu a user can search for and select shows for downloading. Viewing the download-status or playing these shows can be done through the My Downloads tab on top of the screen, where one also finds options for publishing new content.

To allow downloading in the background and scheduled downloads or subscriptions the application minimizes to the tray bar when it is not being used.

2.2.4.3. Set-top boxes

A kind of user interface that is very popular with MDNs using an end-to-end network is the so-called set-top boxes, which are simply small digital devices similar to DVD-players designed to be placed on top of your existing television. Their purpose is to extend the capabilities of the TV-set by providing two-way communications and a user interface, and their tasks may be to enable things like content descrambling, Video on Demand and potentially a host of other services.

Descrambling devices for satellite-TV reception are possibly the most common set-top boxes today. However Personal Video Recorders (PVR) like [TiVo] and even full MediaCenter computers are also getting popular, in addition to a range of hybrid devices that can perform several tasks simultaneously.

The set-top boxes used for MDNs can either be just a simple menu-system that initiates streaming and viewing of selected content, or it can be combined with PVR functionality to store content and even record from other media as well. Many current systems do not allow direct streaming but requires the users to initiate downloading or recording of the content prior to watching.

Illustration 2.7: A set-top box from TeleClub

This is a set-top box from the German cable-TV operator TeleClub AG. Its primary purpose is decoding digital television signals for use with a regular TV. In addition it provides an Electronic Program Guide (EPG) where users can browse the TV-schedule for the channels they can receive by using the remote control. It also supports the image-formats 4:3 and 16:9 as well as output of Dolby Digital AC3 sound.

2.2.4.4. Mobile terminals and smart-devices

With the deployment of the UMTS 3rd generation mobile networks, many new mobile phones and other devices are getting the capability to stream, play and store multimedia-content. This spells that a large market for multimedia content on mobile devices is bound to develop over the next few years, meaning that many MDNs and their end-user clients will be required to operate over cellular networks as well. Some of the end-user devices will have the capability to access ordinary web services or computer software directly, while others will require specially created client side applications to access the MDN services. While the future is promising, this is still an emerging technology, so it is uncertain which options will actually be available on the various kinds of devices. A MDN operator might even be required to provide all the mentioned kinds of services to support the full range of available devices in the future.

Illustration 2.8: A Nokia 7710 video enabled Smartphone

Copyright © 2005 Nokia. Made available as a public press photo at [*http://europe.nokia.com/nokia/0,,73864,00.html?name=7710*](http://europe.nokia.com/nokia/0,,73864,00.html?name=7710)

The Nokia 7710 Smartphone is an example of a modern mobile terminal that can make use of a Multimedia Distribution Network to receive data.
The Smartphone is shown here while streaming video from a BBC World news broadcast.

2.3. Currently existing Internet MDN services

While this report is attempting to design and describe a new architecture for Internet-based Multimedia Distribution Networks, there are already a host of existing services available. Some of these services do not fully qualify as an MDN based on the definition in this report, but others do everything that is described here and more.

Here is an overview of some of the most important services currently available.

2.3.1. Google Video

The largest and currently most well known IMDN service is this beta offering from the search-corporation Google. The service is primarily based on a standard web interface where one may search for, preview, and view streaming movies in a flash-player, and there is also a standalone video player available to view downloaded clips. Any users can upload their own videos to the service and choose to provide them for free or set a price. The files can also be protected by Google DRM if the provider wishes them to be.

In addition to user-provided files the service is also indexing most American TV-programming as well as other premium content from various sources.

All content is natively available in a special flash-based Google Video format, but unprotected files are also available in formats suitable for the Apple iPod Video and Sony Playstation Portable devices. How content distribution happens is not known, but it is most likely performed through Google's own computing grid and CDN that is also used for their other search-related services.

The service can be found at http://video.google.com/.

2.3.2. The Open Media Network

The Open Media Network is an IMDN based on a Grid CDN called Kontiki. The system is a free public service containing a broad selection of free public programs and movies, in addition to user provided content.

Unlike most IMDN services the OMN only delivers its programming through a standalone application that must be downloaded and installed on a local computer. From this application the user can search an extensive program guide and even schedule content downloads for future viewing.

The content is mostly in standardized formats like Apple QuickTime or Windows Media, and DRM may be used at the content-providers discretion. This allows some content to easily be converted and used with iPods, TiVo and WMP-devices, while other content can be protected or in some obscure format.

The OMN can be found at http://www.omn.org/.

2.3.3. YouTube

YouTube is a well-known video-service similar to Google Video, and it is the market leader in user-uploaded content. It too is using a web-based interface where one can search and view streaming videos in a browser through a flash based player. The service is based on self-publishing, meaning that the users themselves must create and upload all the content that is to become available. For this reason there is no premium content available in the service at all. Unlike the other services YouTube also make provisions for users to share videos privately with their friends and family in addition to the open publishing. Videos are not available for download and can therefore not be easily used with other devices, but because of its focus on user-generated content the clips are usually very short (avg. 5 minutes), so there is not much reason to bring them along on a portable player as with the longer TV-shows and movies available elsewhere.

No information is available on how the videos are distributed to end-users, but it is most likely done through a regular server farm or a normal CDN due to the small size of most of the content provided.

YouTube can be found at http://www.youtube.com/.

2.3.4. iFilm

One of the leading short-film distribution networks and one with strong ties to the existing content industry is iFilm. The company is currently a part of MTV Networks and boasts an extensive catalogue of both professional and user-uploaded content which is available through its web-based interface. It is the only service to have a section for viral video clips, and it is also the only one to let users select a format for viewing. WMP, QuickTime and Real are available but each requires the respective players to be installed on your system.

However there are no easy ways to download any content from iFilm and you are forced to view a short advertisement before your selected content will play.

Also the free user-uploading feature does not guarantee that your video will be featured on the site, as this privilege is limited to those who pay a subscription fee per video. The site is therefore more geared towards filmmakers looking to be discovered rather than the amateur who just want to publish some clips. In addition the site is primarily focused on short films and clips rather than longer features.

There is no official information available on the distribution network used by iFilm, but a quote on the website of Akamai Technologies states that MTV Networks make use of their services for content delivery on the Internet.

You can find the service at http://www.ifilm.com/.

2.3.5. IFC Uncut On Demand & the IFC Media Lab

The Independent Film Channel is a regular cable-TV station in the United States that also provides online on demand services and a Media Lab where users can upload content. The Uncut On Demand channel is a web-based MDN from IFC providing free independent media content through a flash-player. In addition one can also view user-created content uploaded to the IFC Media Lab from which the best clips are being featured on regular IFC television.

While being more community based than the other services reviewed, IFC like iFilm is mostly focused on aspiring film-makers and those with special interest in independent films rather than at the general population. There are no details available on which kind of distribution network they are using.

The IFC Media Lab service can be found at http://medialab.ifc.com/.

2.3.6. CinemaNow

This is the leading internet distributor of feature-length films and other content from professional content producers. The CinemaNow network is like most of the others based on a web interface from where users can locate, buy and download the desired films. There is no user generated and uploaded content available, but instead the service provides a large amount of long-playing movies and programs that are not available with other services. In cooperation with WatchMusicHere.com the service also provides various kinds of audio content, and CinemaNow is also the only service to include adult-content in their lineup.

There is a small selection of free content available, but most of CinemaNow's features have an individual price tag or require signing up for a subscription plan. All of the content is distributed to end-users in the Windows Media 10 format through PatchBay, a proprietary content-on-demand distribution and DRM system that runs the CinemaNow Web site and its sister sites.

The CinemaNow service is available from http://www.cinemanow.com.

2.3.7. Other sites and services

In addition to the services mentioned above there are also a range of competitors rolling out various kinds of multimedia distribution systems. A few amongst these are Vongo (vongo.com) and Movielink (movielink.com) that are both focused on providing movies and television programming, but as both of these services require a user to be located in the United States and are similar to CinemaNow, their operation has not been considered in detail here. Also the focus on social-sharing of video by YouTube has inspired a wide range of competitors. Most of these are merely clones of one or more existing services, but some provide a degree of innovation as well. EyeSpot (eyespot.com) does for instance provide editing facilities that allow the user to combine, cut, splice and mix their videos online in the flash format, and with JumpCut (jumpcut.com) users can even do such editing with videos from other people.

Another service that is not reviewed is [Audible], a kind of MDN that indexes audio books, radio shows and other audio-only content. It is left out because by limiting itself to audio it does not have comparable size-requirements to video-services and therefore does not have to address the same storage and bandwidth problems and costs that this thesis is focusing on. The same applies to the wide range of other music-download services on the market, but the findings in this report will be applicable for these kinds of services as well.

3. Internet MDN implementation

As discussed in the previous chapter, this thesis will focus on techniques for implementing a Multimedia Distribution Network using only the public Internet for end to end data transfer, a so called IMDN. This chapter makes a comparative analysis of known techniques for IMDN implementations as a factual base for deciding which techniques are best suited for the MDN architecture to be proposed in this work.

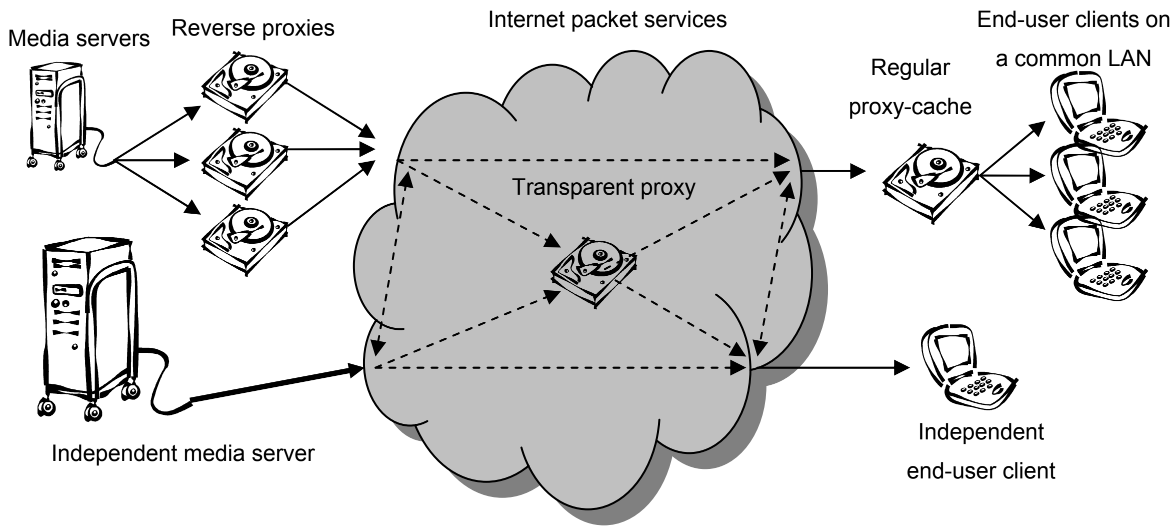
3.1. Potential architectures

3.1.1. Proxy-caching

The simplest way of providing content distribution is through proxy-caching, where one has one or more intermediate systems to which the user connect to retrieve all kinds of data. If the primary proxy-cache does not contain the requested content it in turn connects to the next cache or to the origin server to retrieve the data before passing it on to the requesting user. When this happens the proxy also stores a copy of the content for an amount of time since it is expected that more users will request the same data relatively soon according to the principle of temporal locality. This way, the next time someone requests the same data, it can be quickly provided by the proxy instead of burdening the origin server. Each proxy will usually just serve users within the autonomous system of a single network operator, or even just a subset of such a system.

Proxy-caching has been in use on the Internet for a long time and is a proven way to provide simple content distribution. Proxies are currently in use by thousands of organizations world-wide to improve the Internet service for their clients.
The most common type is called a forward-proxy and has the advantage that it does not put any administrative overhead on the content providers, as each cache is administered by the various network providers for their own users only.

Illustration 3.1: Types of Internet proxy-caches



Various types of proxy caches are used in very different environments despite their nearly identical mode of operation. Regular proxy-caches are usually located at the user's edge of the Internet to provide faster access to common content for the clients on the private network behind it. Reverse proxies on the other hand are for instance used by server-hosting providers to do load-distribution of popular content for one or more servers located in their server farm, while transparent proxies are used by network providers that seek to reduce their-own bandwidth requirements while being invisible to the communicating parties on both sides of the network. All three types of proxy-caches can even be in use simultaneously depending on the route through the network.

However they do put some extra overhead on users since every application must be configured to use the proxy instead of connecting to the origin servers for some or all of the requests they make. Another downside is that the original content provider has no control over how many users are accessing the content through a proxy-cache, and they also have no control of the freshness of cached content. These problems can be reversed by using what is known as a backward or reverse-proxy instead.

The backward-proxy is often called a server accelerator as it is usually a cluster of proxies installed in the network-neighbourhood of a content server instead of close to the end-user. It is often a requirement that all connections to content servers go through the proxies, thereby both protecting the server from direct access and providing load distribution in various ways, like caching static content

or performing encryption for secure sites. Proxy-selection and load distribution can be performed by simple methods such as round-robin DNS look-ups. Many types of basic CDNs are based on a variation of the reverse-proxy design to provide their service.
A third variation of the proxy-cache is what is called a transparent cache, a proxy that some network operators install in their network transit points. The transparent-proxy examines all passing traffic, and if possible serves the requested content from its cache to reduce network traffic and improve response times for end-users. When installed the transparent cache does this automatically for all supported traffic and it doesn't require any active configuration by neither end-users nor the content providers. This makes these kinds of proxies practically invisible in the network. [Proxy]

The most important part required to create an efficient cache is selecting the replacement policy that decides which items should be purged from the cache when the cache is nearing the limits of its storage capacity. Popular policies are ones that removal of the least recently used (LRU) item, and also removal of the least frequently used (LFU) item. The efficiency of such replacement policies is usually measured against Belady's Algorithm, which will remove the item that won't be needed for the longest time. Belady will always give the optimal selection, but since it requires knowledge of the future it cannot be used for anything other than an optimal comparison baseline. [Cache]

While proxies are currently very popular and effective for basic web-contents, they do not currently perform very well for multimedia content due to the large file-sizes and storage requirements involved. Also proxy-caches use the store-and-forward model based on someone requesting the content, and they are therefore not very well suited to provide real-time streaming multimedia from for instance live events. In addition the Internet-oriented model of proxy-caches means that they cannot usually be managed from a central location. For these reasons this report will not give proxy-caches any further consideration when suggesting the MDN-architecture.

3.1.2. Replication

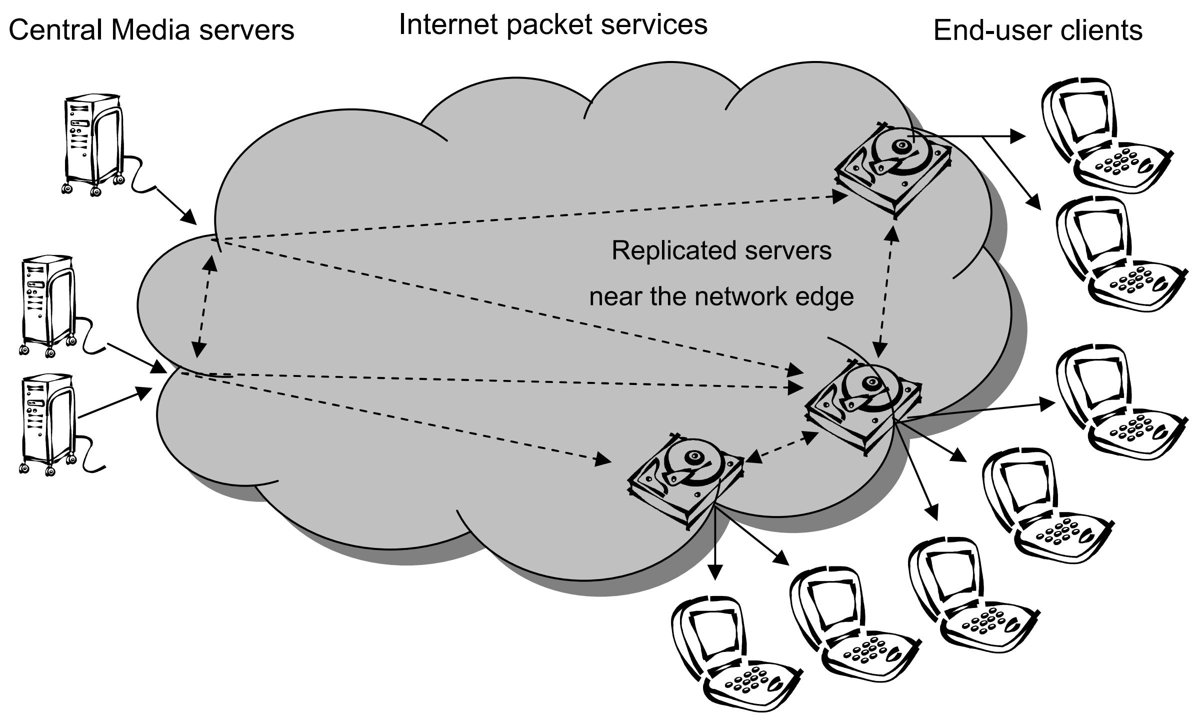
A more advanced but still simple way to provide content distribution is to replicate the data set across a series of servers for redundancy and increased capacity, much like the operation of backward-proxies. However in addition the servers in a replicated network can be widely distributed geographically to improve the response times experienced by users all over the world, compared to a backward-proxy that are usually collocated with the origin server. Similar to backward-proxies this technique allows content providers to retain full control of the published content, including the option to keep accurate usage statistics, but in exchange the content providers must carry the costs associated with the distribution. [Replication]

The replication itself can either be static (non-adaptive) where the full content is available from a series of predetermined sites, or it can be adaptive and dynamically distributed by demand. Like caching, adaptive distribution requires a good replacement policy.
Replication is the technique used by most regular CDNs such as those operated by Akamai and LocalMirror, and also by a range of other high traffic websites such as C|NET Download.com and the Open Source Technology Group (OSTG).
In addition to the problems with selecting a replacement policy a replication network also needs policies for amongst other things server placement and request routing, meaning the selection of where you need to place the replicas and which site is best suited to serve a particular user. Deciding where to put the servers can for instance be done with the hot-spot algorithm that picks sites close to the clients that are generating the most requests. [Replication]

For request-based routing, the high-traffic websites usually just give a default selection based on the originating IP-address, or simply give the user a list of choices for a manual selection that can be remembered through browser-cookies. Neither solution is very good since both will likely provide the user with a suboptimal site in a wide range of cases. In addition the selection will at best be at a very course granularity, something that can be a problem in large areas with dense populations of both Internet-users and server-farms, like for instance most of the continental United States. In such areas there is often no correlation between the IP-address and a user's geographical location, which is even less correlated with the network-topographical location, making IP-based selections worthless and manual-selection hopeless since very few users will have any knowledge of the network-topographically best site.

The major CDNs usually try to avoid these problems by using a range of different algorithms to make informed decisions for selecting a site, but analysis [CDN-Use] shows that their attempts are far from perfect even though they usually provide a better service than random selection [CDN-Perf].

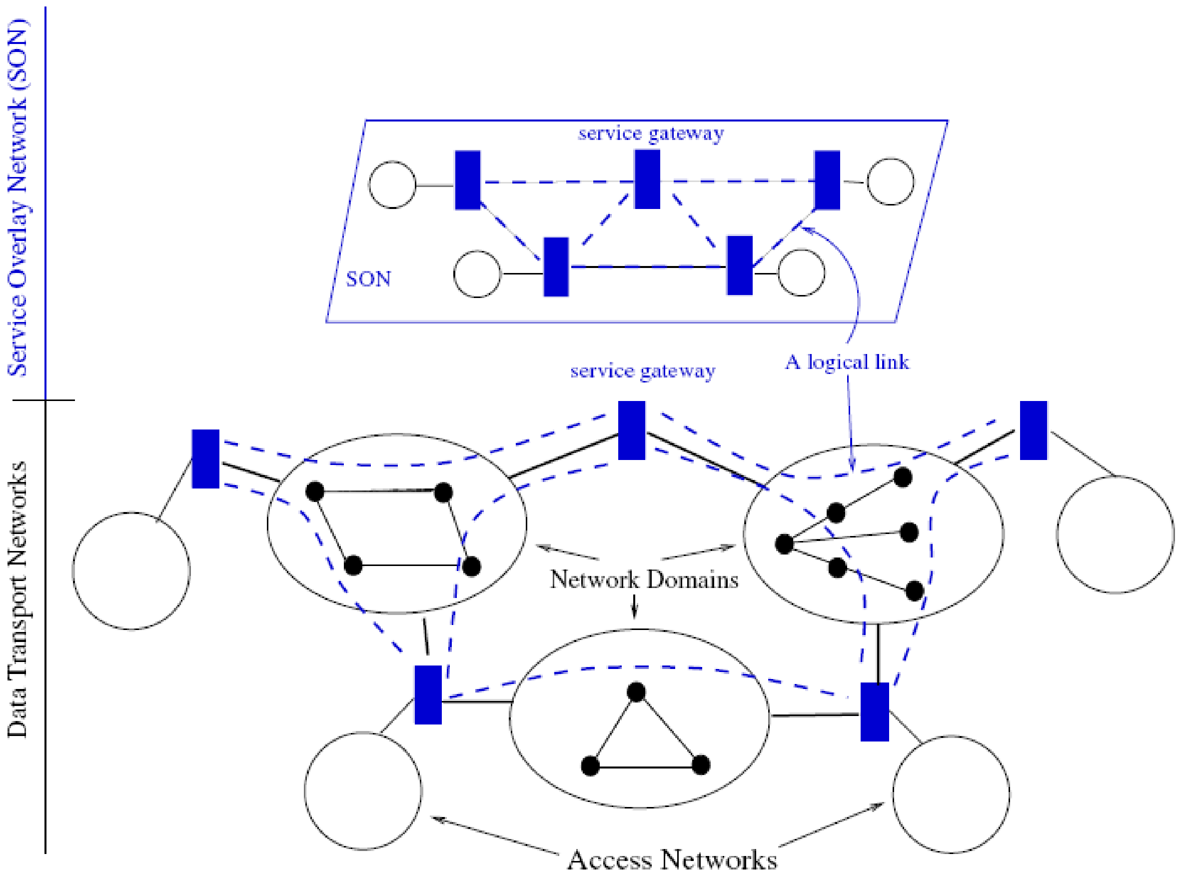
Illustration 3.2: Replication of distribution servers



Replication servers are a type of backward-caches that are located near the network edge close to end users so that the replicated content can be more effectively delivered to the recipients while the owners can still retain control of the replicated content. Each replica will ordinarily only serve a defined set of users, for instance the subscribers connected to a specific ISP or in a specific country. Which replica serves a particular user is often decided dynamically on request.

A variation of the replication network architecture for real-time multimedia streaming is the networks described in [Reflector]. This kind of reflector-networks makes use of IP-Multicast to distribute a media-stream from the originating host to a series of servers running Reflector-software. The reflectors receiving these streams can then re-reflect the streams onward to other distant reflectors or to end-users in the vicinity, but the same problems that plague replication networks also apply to reflectors. This type of basic architecture should also be able to support multimedia streaming in addition to static file replication. Another service very similar to this one is described in [Prism].

Illustration 3.3: A Service Overlay Network



This is the basic layout of a Service Overlay Network as presented in [SON]. A SON is able to provide QoS-guarantees over its logical links by signing SLAs with each operator of the independent underlying physical networks that perform the actual packet-transportation.

Copyright © 2002 Duan, Zhang & Hou. Figure 1 from the paper [SON].

3.1.3. Service Overlay Network

A Service Overlay Network (SON) is a new way to provide improved end-to-end Quality of Service (QoS) over the Internet as proposed in [SON].

A SON is like most other overlay networks created by designing a logical architecture that functions on top of an existing physical network. However, providing a multimedia delivery infrastructure with strict QoS requirements over such an overlay network without support from the underlying operators is very difficult, so in addition to designing and deploying the logical architecture a SON operator must also enter into Service Level Agreements (SLA) with each of the operators of the underlying physical networks. Through these agreements the SON can then be based upon a physical network with known and guaranteed QoS-parameters on which the logical service can be built. This allows the SON to make its own guarantees to its customers and to be able to deliver real-time streaming video in a timely fashion to end-users.

Despite a SON being a good way to provide bandwidth guarantees to ensure a certain quality level when streaming multimedia, the method has a large administrative and financial overhead since it requires the MDN operator to make agreements with all relevant network providers. In addition to requiring a lot of work to make such deals, it might not be possible to make such agreements with all operators, and the agreements one does secure will usually come at a substantial economic cost. For these reasons a SON solution is not a suitable base for the architecture suggested in this report, but for comparative reasons it will still be considered in the comparison in the next chapter.

3.1.4. Pure peer-to-peer

A peer-to-peer (P2P) based architecture literally means that the data is distributed directly between peers, meaning the set of hosts that are actually requesting the data, instead of from a range of designated distribution servers to end-user systems. This means that the end-users requesting the data in question are the ones supporting the distribution to the rest of the network by donating some of their own system resources according to a predetermined set of rules.
A very desirable property that arises from this kind of architecture is that as the popularity of the network increases, so does the resources available to distribute contents since there are more users donating resources.* [P2P]

The peer to peer distribution model was developed in the years of the Internet-boom and popularized by the music-downloading service called Napster, followed by Gnutella and many others. These first generation P2P architectures weren't pure P2P since they relied on central servers for indexing. Also they did not handle additions and removals of active users very well, and while the frequent disruptions only caused some problems for normal downloads they made efficient transmission of a continuous stream impossible across such networks. Even though Napster was shut down by legal action in July 2001, its success, and the continuing success of the services that followed has led to much interest in the field by researchers. [Napster]
A result of this is that a range of theories on how P2P-streaming can be used for stable multimedia delivery under the inherently changing conditions of P2P have been developed, including a technique called swarming used by among others the FastTrack network and popularized by BitTorrent. Using this technique every shared file is split into pieces that can be downloaded simultaneously from different peers. This eliminates the problems caused by the source of a file disconnecting or having limited bandwidth, and it also increases the capacity of the network since the receiver can share the completed pieces with the swarm

* Note that this depends upon the selected rule set and its implementation into the system and clients.

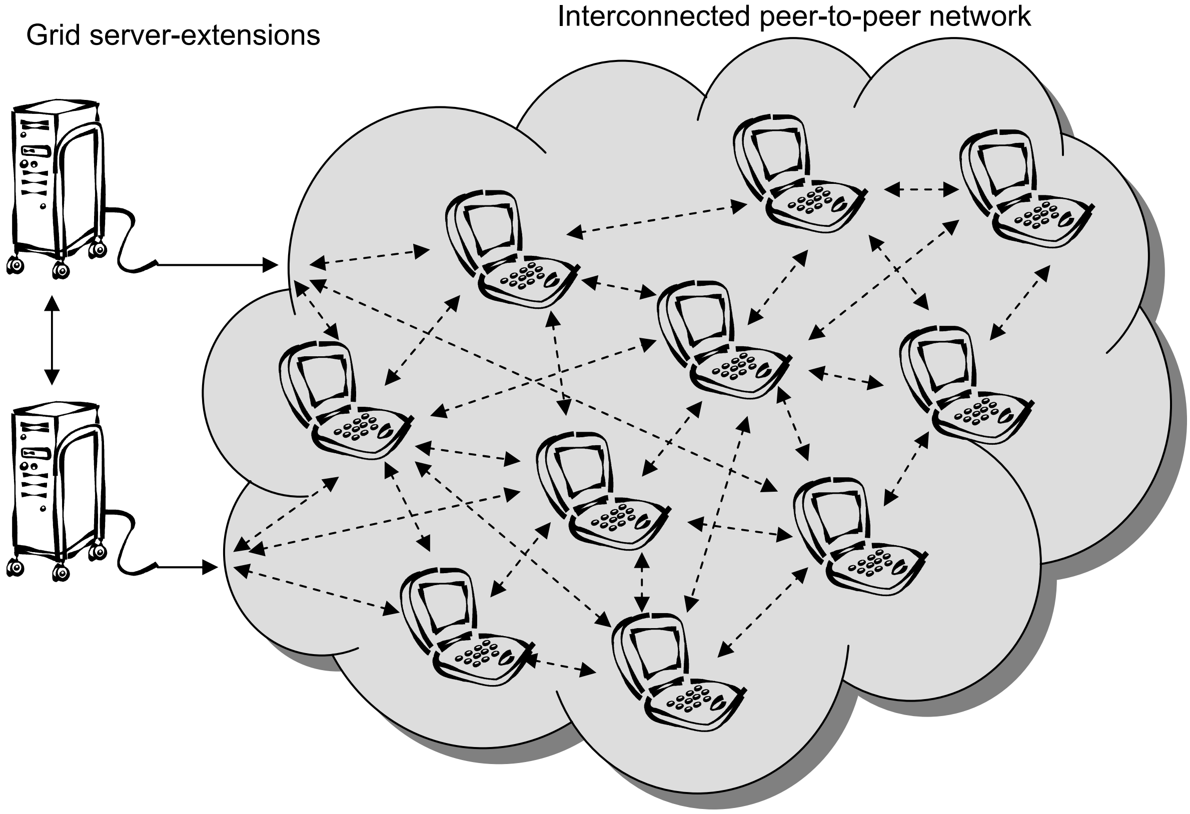
right away instead of waiting for the completion of the entire file like in the older systems, but since the pieces are randomly downloaded they cannot be used for streaming media.

This can be improved with a range of new developments like the [ZigZag] and [CoopNet] architectures. If these developments could be combined with for instance the theories presented in [mOverlay] to support better locality in the trees, the result would be a very efficient and high-capacity streaming network that could be a suitable base for an IMDN.

A particular point with pure P2P networks is that besides from the originating server of the content, the network is designed to be completely independent. This means that the network will function without any supporting servers or other mechanisms that can become a point of failure or target for legal action.

Despite this, none of the currently popular peer-to-peer protocols are completely pure as most either rely on some kinds of centralized servers or on super-peers that are more important to the network than the average user. The protocol closest to being pure is BitTorrent, but since it still requires the use of external websites to locate torrents and most users need central trackers to manage them it has not yet reached the point where equal peers participate exclusively.

Illustration 3.4: Grid/P2P network layout



A pure P2P network is a large amount of interconnected clients that exchange pieces of a data-set with other peers with the goal of distributing the entire data-set to all the connected peers. A Grid network is simply an extension of this system where a set of servers are introduced to aid the grid with common tasks like indexing and also to provide a new set of options like access control and stability.

3.1.5. Grid delivery network

The grid delivery network is a relatively new kind of content distribution facility. It is based on many of the ideas from peer-to-peer networks, similar to replicating systems being an improvement of the proxy-cache. However there are still a range of differences between these two kinds of systems. Grid delivery networks require all the participants in the grid to run application that controls the transfer of files from servers and between peers, just like in a P2P solution. Current grid delivery systems make use of BitTorrent-like swarms for bandwidth-harvesting for efficient delivery of popular content, but a new feature of the combined system is the ability to time-shift scheduled downloads away from peak-hours when the network is most congested. Grid networks also use a set of selection algorithms to get the data from the least congested peers, unlike the current systems that often make no distinctions regarding which peer a request should be sent to.

The authors of [Hybrid] discovered in 2003 that only 5% of the peers using the P2P networks Napster and Gnutella were powerful enough to actually relay streaming multimedia, which shows that a pure P2P system would not be very efficient. Based on this they suggested using a hybrid architecture where a regular P2P architecture is complemented with streaming servers, and thereby effectively creating what is currently known as a grid network. While both the bandwidth availability and processing power with home users have increased greatly since 2003, so has the quality of multimedia that users expect to be able to stream flawlessly, so their statements still remain valid to a degree.

Current commercial grids also rely on complementing directory-services and other host-servers for performing advanced tasks in addition to what is provided by the peers of the network. While this do require more administration than it would take to operate an independent peer-to-peer network, it does allow the grid to add services like access control, user authentication, virus-scanning and other features even while permitting content to be published by any user connected and authenticated to the grid. This is actually somewhat similar to the design used for the earlier P2P networks starting with Napster, which also required a centralized directory, but in those networks the directories were essential to operations and they usually did not provide any additional services.

Among the most well known grid delivery systems today are those by [Kontiki] and [Avaki] upon which this description is based, but there are a range of competitors on the horizon as well. To appeal to businesses, these Grid-systems provide a lot of extra services such as grid computing and collaboration tools in addition to the media delivery that this report focuses on.

3.2. Evaluation of architectures

To decide which architecture will be best suited for the proposal to the Aeetes Project it is necessary to perform a detailed comparison of the costs, strengths and weaknesses of the combinations available. The range of features in these architectures that could be compared is very broad, but most features are not relevant for this comparison and will therefore be omitted. This comparison will focus on a few main points that are believed to be of considerable importance for the future implementation and deployment of the system.

Table 3.1: Criteria for evaluation of architectures

Architecture cost	The type of supporting architecture behind the MDN is important in regards to the resources required to deploy the system. A network that requires expensive dedicated hardware will be less suitable than a software-only solution, and dependency on a third-party server-hosting or network company will add an extra layer of expensive administration.
Architecture management	To allow for simple deployment for non-commercial use it is best if the architecture does not require a separate staff in one or more locations to manage hardware or network settings. Managing server and network operations or a large amount of service agreements can be a toll on a small non-commercial system.
Complexity of system	An overly complex system would mean that the development time will be long and it is also more likely to contain serious bugs or design flaws. A very simple system on the other hand might not be adaptable enough to evolve with changing requirements, so this too should be avoided. Since the Aeetes Project will be developed primarily through student projects it is extra important that the complexity of the system is not overwhelming.

Delivery performance	Besides the details of the architecture itself, it is essential that the system will be able to provide high performance during the load scenarios that can be expected to occur during normal use. This means that the architecture should not contain any artificial limitations that could lead to transmission problems, but rather have extra features that would allow it to sustain high performance even under difficult network conditions.
Control potential	Since it is expected that this system will be deployed by some party that has a need to transmit data to a group of users, and not primarily to allow users to share data between themselves, administrators should be able to monitor and preferably also control both the data- streams and the connected clients
Ease of manageability	It is important that the system is easily manageable from the point of view of both administrators and end-users. Client installation should therefore be a simple task not requiring any fiddling with computer or network settings, and the available control functions should be easily accessible and not require any special skills to make use of.

Now that we have the relevant comparable features defined, we need to do a comparison of these features for each of the different architectures that are mentioned above. Since there are no authoritative sources available that can provide an analysis of the kinds of features we require, the assessment below had to be based on the author's specialist knowledge of both the Internet and of the compared architectures instead.

3.2.1. Evaluation of the replication network architecture

Replication networks are very costly to deploy since they require a distributed set of servers in all areas that need coverage, and to supply these servers a large amount of bandwidth must also be deployed or rented. In addition to this a great deal of administration is required to successfully operate such a distribution network and to keep it running through the hurdles that can crop up.

Achieving good performance from a replication type distribution network does unfortunately require very complex software, both to perform the actual replication and keep it up to date and for making content decisions. But, this is at least possible and when implemented it will provide an excellent service.

On the upside, a replication network can provide a very high level of control by administrators, and controls are also easy to implement since the operator has full and potentially direct control over all the parts of the entire network. Also using replication will often not require an end-user to make any client-side modifications as the network will operate hidden from direct interaction with the user.

3.2.2. Evaluation of the Service Overlay Network architecture

This architecture does also come with quite large costs since one must sign Service Level Agreements with all network providers through the entire part of the Internet where coverage is desired to achieve the full potential of the system. However, once set up, the network operators will handle the practical details of the connections and a relatively low number of servers can potentially provide service for large areas. To make efficient use of the SLAs the server-software need to be quite complex, and system performance will be limited by the low number of servers and high costs for increased bandwidth despite using the available resources in a near optimal fashion.

As with a replication network, a wide array of controls can be made easily available as all servers and networks are controlled by the operator, but it might also require end-users to actively connect to the SON to be able to use it.

3.2.3. Evaluation of the peer-to-peer architecture

Since a peer-to-peer network is deployed only as a software application that donates some of the local computing resources to the network, there is very few costs involved for the operator and there is also very little management required since there are no distribution servers or network links that must be administered.

The client application can be based on simple principles of operation, and is therefore not very complex to develop after the appropriate underlying protocols have been decided on and integrated in the system. This solution also provides good performance to end-users since the capacity available for a given object will increase with its popularity; however there can be some problems with objects that are only popular in a limited area or not popular at all.

Because of the distributed approach, it is very hard to implement any system controls for the managers other than what can be placed at the origin servers of the data-streams. Even though the controls can be implemented on the user-side and may not bring any complications other than the installation of the client itself, they unfortunately could lead to some troubles if firewall and NAT-traversal facilities are not implemented properly.

3.2.4. Evaluation of the Grid architecture

The improvements made to produce a Grid delivery network include adding a set of dedicated servers for distribution and indexing. This naturally adds more costs to the deployment and requires more management than a pure P2P. However the number of servers that are added may be very few and it is not required that all network operations are integrated. The addition of these servers does however add some extra complexity to the software, but it has the benefit of ensuring good performance under any circumstance.

Since there are centralized servers under operator-control available in a Grid network, it is much easier to add management controls than in a P2P, and from the users point of view there are no differences between a Grid and a P2P network other than the available feature set and performance experienced.

3.3. Architecture comparison

To compare the results of the architecture evaluations in the previous chapter, each of the architectures is assigned a suitability value according to the scale in Table 3.2 for each of the criteria that have been evaluated. Table 3.3 show the suitability assignments, and an arithmetic sum of the suitability values for all the architectures.

Table 3.2: Suitability scale

Suitability for project	***Suitability value***
Very Good	5
Good	4
Medium	3
Bad	2
Very Bad	1

It can be argued that the features that were evaluated in this analysis should not be assigned equal weight, and also that the assigned values are hardly better than random as they are not based on any hard facts or determined through a defined algorithm.

While it would have been preferable to assign relevant weights for each of the features, this would have only complicated matters unduly since there is no good or fair way to decide exactly how these weights should have been distributed. It is for this reason that equal weights have been used, and because of this it has been taken extra care to select criteria that are close to equally relevant. A side effect of this however is that the variation in the results might not be representative for the actual differences between the architectures, but the summary value should still give a good indication on the suitability of the architectures relative to the others that was evaluated.

Likewise are the rough manual assignments of values based on the assumption that it is better to make a decision on an educated evaluation than on nothing at all when there is an absence of hard facts that can be used to make a better point.

The result of these approximations is that the values calculated in the comparison table can only be taken as general guidelines, as they are not indisputable facts stating which architecture is best suited. Despite this there is still a range of interesting knowledge to be gathered from the results shown in the table.

First of all we can easily see that all of the reviewed architectures have some features that make them suited for the project, but also that all of them have some other unsuitable features. Another notable thing is that solutions based on the P2P and Grid architectures do stand out as being somewhat more suitable for this project than a solution based on a SON or Replication architecture. We also see that P2P appears to be marginally better suited than using a Grid network, and if we analyze the actual comparisons we find that this makes sense as a Grid network would both be more costly and require more administration due to the centralized servers. It can be argued that the advantages gained from this do not negate the added complexity and cost of deploying a Grid network compared to a P2P network.

Table 3.3: Architecture suitability comparison results

Network type	**Replication**	**SON**	**P2P**	**Grid**
Cost	Very bad	Bad	Very Good	Medium
Management	Very Bad	Medium	Very Good	Good
Complexity	Very bad	Bad	Good	Medium
Performance	Good	Medium	Good	Very Good
Controls	Very Good	Very Good	Very Bad	Medium
Manageability	Very Good	Good	Good	Good
Summary	16	19	23	22

All of this combines to show that based on the expected usage and defined requirements for the design of the Aeetes Multimedia Distribution Network, it will most likely be the best choice to use the peer-to-peer architecture.

However it should be noted that this decision is based purely on informed speculation, so the appearance of any kinds of hard facts supporting one of the other architectures should be cause for reconsideration of this decision.

4. The Aeetes MDN

This chapter contains a proposal for an architecture that can be used as a base for the Multimedia Distribution Network that will be developed through the Aeetes Project.

4.1. Properties of the proposed architecture

Based on the evaluations and comparisons in the previous chapter, it appears that a peer-to-peer based architecture is the best choice for building a Multimedia Distribution Network according to the expected usage and requirements of Aeetes. This chapter will therefore review the qualities of a P2P network for such use in more detail. Most of the chapter will be referring to the BitTorrent protocol for examples and comparisons, since it is currently the most successful pure P2P system that attempts to avoid the use of central servers and super-nodes.

For readers who are not very familiar with peer-to-peer content distribution systems, it is recommended that the paper [P2P-survey] be studied in detail as it provides a broad overview of the various common P2P techniques and technologies currently in use.

4.1.1. Equality of peers

A peer-to-peer based architecture is as previously described an architecture where every peer is equal and each peer donates some computing resources to participate in distributing the requested data to every other peer in the swarm while at the same time receiving the data for themselves.

There will always be one particularly important peer, namely the original source that was providing the data to the swarm in the first place. This original peer will always be a "seed" or a "mature peer" that does not download anything but only transmits data to the other peers. For finite file-sizes most of the other immature peers will eventually have received an entire copy of the file and should then as well switch to seeding the file for some time to increase the total efficiency of the swarm. [BitTorrent]

The above will not be the case for streaming media however. Since a stream in theory is infinite there will never be any other seeds than the original source peer from where the stream is originating. This also means that any receivers of continuous stream(s) can be only expected to donate resources to the swarm as long as they are actively receiving the stream(s) carried in that swarm for themselves, since they otherwise would not have any current data to distribute.

4.1.2. Accessing the network

A challenge with a peer-to-peer based system is the design of the architecture that allows new users to locate and join an existing swarm. The original version of the BitTorrent protocol solved this part by having a dedicated tracker keep track of all the clients in each swarm, and left locating the tracker out of the protocol entirely. All downloading by the BitTorrent protocol therefore require users to manually locate a .torrent file containing the data and tracker information from an external source, usually a web-based search engine of some sort, before the desired data could actually be retrieved from the swarm. However, since the trackers were not distributed they could become a single point of failure that might be attacked through litigation or denial of service.

Because of this, the BitTorrent protocol has recently been updated to include support for both Distributed Hash-Tables (DHT), the technique used in the distributed approach of the eDonkey network, and Peer Exchange (PEX) which is a gossip protocol for peer discovery in a swarm. This allows for the creation of trackerless torrents without single points of failure. [BitTorrent]

Despite such innovations most users are still required to locate the necessary torrent files manually, and while amongst others the Magnet URI Project attempts to address these problems, it is as of yet not commonly used among file sharers. [Magnet]

.

4.1.3. The problem of freeloaders

A common problem with a peer-to-peer system is freeloaders that retrieve content without donating any resources to the swarm. If such behavior is widespread the advantages of using a P2P-network will disappear as in the worst case the originating peer will remain as the only one transmitting data, meaning that the network effectively has degenerated to a regular client-server system.
This is actually a well known problem from game theory similar to the famous Prisoner's dilemma where the optimal solution can only be reached by all parties cooperating. One party can receive an advantage paid for by the other through acting selfishly, but this advantage materializes only if the other party doesn't act selfishly as well. [PD]

In the case of BitTorrent, it is for this reason customary to at least seed until one has sent out as much data as one has received, since stopping any earlier could have adverse effects on the efficiency of the swarm if it is done by many users. This is understood by programmers so most client-applications are designed to promote this behaviour, for instance by not allowing automatic halting of an upload until it reaches a 1:1 share ratio or better. Also the protocol itself encourages such behaviour by limiting the possible download speed to match the promoted uploads, and in addition these effects are partly countered by the fact that the original provider of a file will often be seeding until the content of the file gets dated and no longer is of any interest to other users. Providers do this since it is usually in their self-interest to distribute the file as widely as possible, so combined this means that the swarm will stay efficient as long as there is a critical mass of downloading users. [BitTorrent]

For streaming media the seed/peer conflict is non-existent since a stream in theory is infinite, which means that there will never be any other seeds than the providing peer where the stream is originating. However freeloaders can still cause problems if it is not ensured that subscribers to a stream also contribute by transmitting as much as they are retrieving. This is especially important since the volatility of the users in a swarm is much greater for a stream than for finite files since they cannot keep seeding without downloading themselves.

Also it should be noted that freeloading is not the only kind of malicious behaviour that can be damaging to a peer to peer network. In [Resilient] there is an analysis of various attack vectors and weaknesses in P2P networks, and there are some suggestions on how the networks can be reinforced and become more resilient against such attacks and problems.

Note however that in most cases the freeloaders are not a large problem since some restraints can be implemented in the clients that the large amount of non-technical users run. These non-technical users often lack the skill required to bypass the restraints, and those that do often sympathize with the network operators and also understand the necessity of donating resources to maintain the network. In addition, most users are not averse to sharing their resources when it comes at no direct cost for the individual user.
However, if there are powerful external forces at work, this balance can be skewed causing the majority of users to avoid donating resources and thus greatly limiting the efficiency of the network. An example of such forces can be the fear of legal action that may result in hefty fines or imprisonment for the individual users if the content in question is distributed in violation of copyright.

4.1.4. Control and management issues

A big issue with using P2P for an MDN has been identified in the Suitability comparison chart, namely that it is very hard for a content provider or distributor to retain any kind of control of the distribution and accessibility of the content.

For BitTorrent, the shift to using DHTs caused trouble for private trackers which require a user to be registered to access premium content. They do this to be able to gather statistics which is used to identify and ban "leechers" and other misbehaving users. However the DHT and PEX-functionality enabled unregistered users to bypass the tracker by connecting directly to the other peers in a swarm, and thereby get access to the content. To prevent this leak, a 'private' flag was added to disallow the use of DHT for such torrents, but some clients like BitComet ignore this flag and also misbehave in other ways to gain advantages

for its users on the expense of the swarm as a whole. This has caused users of the BitComet client to be entirely banned from using many trackers. [BitComet]
The result of this is that for a content provider to be able to truly limit distribution to only the intended recipients, the provider must depend upon some kind of external DRM-solution as this is not possible within the P2P network itself.

Neither do the developers and providers of a pure P2P architecture have any control over what the network are being used for. This too is clearly illustrated by the BitTorrent example where creator Bram Cohen has been spared from legal attacks from the music and movie industries as he himself claims to have never violated copyright law. The creators of Napster and Gnutella who operated server-based P2P networks have been forced to shut down their networks after such litigation, but because of the nearly pure P2P nature of BitTorrent it would not be possible to shut it down even if Bram Cohen himself wanted or were being forced to try. As both the protocol and application are open source this is made even less possible since anyone can keep improving the software independently of the original creator. While this in many cases is considered to be a desirable property of the system, it is sometimes required that the creators and providers of an MDN are able to leverage some form of control over the system. [P2P]

4.1.5. Single vs. multi-set swarming

Current P2P protocols are using so called single-set swarms for file distribution. A single-set swarm is defined as a swarm that is sharing a single set of data, and only that set of data. Note that this does not preclude membership in multiple swarms. The clearest example of a single-set swarm protocol is BitTorrent where each set of files are described by a single .torrent. This torrent provides a client with the address of the tracker that manages the swarm dedicated to exchanging the data-set described in the torrent. Other file-sharing networks like eDonkey have native simultaneous support of multiple single-set swarms within the same client, but each swarm still operates independently and is dedicated to only exchanging their single set of data. This separation between swarms also applies to third-party BitTorrent-clients like Azureus and BitComet that support the use multiple active torrents simultaneously.

A multi-set swarm is when a client connected to a swarm aids in the exchange of other data than what the client itself is interested in for itself. This is performed through the use of a temporary cache where data passing through the client is stored, from which it can be distributed to other clients. This means that in addition to processor time and network bandwidth, a multi-set swarm also requires the participating clients to donate additional storage capacity.

Multi-set swarms are currently very uncommon in ordinary P2P networks for file-distribution, but they are being used in some anonymity services like Freenet. This use is mainly due to the anonymizing qualities provided by the multi-set swarm, and it is not clear whether they can provide other advantages over the regular single-set swarms. In fact when studying Freenet, it even appears to operate less efficiently, but this might very well be due to the very high overhead that is required when providing anonymity. [Freenet]

.

.

.

4.2. System design and technology

This chapter suggests the overall system design for the implementation of a peer-to-peer architecture with focus on the qualities required by a MDN network.

4.2.1. The peer-to-peer protocol

The most important selection to make when creating the architecture behind this MDN system is the choice of underlying peer-to-peer protocols that will handle data transfers and peer interactions. To be able to pick a good protocol we need to define which capabilities are required to support the MDN as outlined earlier, and identify which protocols can provide the necessary features.

4.2.1.1. Required capabilities

Most of the required capabilities in the protocol to be used for Aeetes follow inherently from the definition of what an MDN is, and from the capabilities that the Aeetes MDN will be expected to provide.
These capabilities include:

- Searching and/or browsing a catalogue to locate desired content
- Downloading of static content files like finite media clips
- Live streaming of both static content and continuous media

In addition there are additional compatibility features that are of some importance to ensure that the protocol can be used everywhere and on any system. Among them one will for instance find these capabilities:

- The protocol should function independently of IP multicast.
- Adaptive encoding of content for variable bandwidth is preferred.
- System requirements should allow for use on old and very slow devices.

4.2.1.2. Potential protocols

During the research phase of this thesis a number of potentially suitable protocols for Aeetes were identified. The most notable of them are:

Table 4.1: Details of potential protocols

ZigZag	A protocol to handle live streaming media over P2P. ZigZag is optimized for short end-to-end delays and low overhead, meaning it handles flash crowds and live streaming very well. [ZigZag]
CoopNet	This is a centralized protocol using P2P to distribute bandwidth for both live streaming and downloads. The focus is on client join/parts to stay functional also under flash crowds. The content-server is expected to be powerful with high-bandwidth. In addition it supports MDC for adaptive-rate streaming. [CoopNet]
Osmosis	Using a cache-and-relay application level multicast this protocol allows asynchronous live streaming from an origin server to users in P2P clusters to minimize total network link cost. [Osmosis]
Scribe	A scaleable decentralized application level multicast system built on top of Pastry that provides routing and locating services. A multimedia streaming service can be built using Scribe as a transport layer service. [Scribe]
MarconiNet	Based on IETF standards to provide signaling, control functions and media delivery this CDN architecture is designed to deliver streaming media over 3G/4G wireless networks with client mobility. In addition it also provides a security model for content with integrated payment models for commercial use. [MN]

4.2.1.3. Protocol comparisons

To provide an overview of the potential protocols in relation to the required capabilities described, they are combined in a mesh in Table 4.2.

A "No" value in any field means that the protocol as described does not have integrated support for that capability, but it will in most cases be possible to implement the capability as either an external or a third-party add-on module. Scribe is a special case as it does not detail the streaming functionality itself but only provides the underlying application level multicast support. Note also that Osmosis, unlike the others, supports asynchronous live streaming. This means that end-users can follow the same live stream from its beginning even when some of them start the subscription to the stream at a later time.

Table 4.2: Capabilities of P2P protocols

Networks: / **Capabilities:**	***ZigZag***	***CoopNet***	***Osmosis***	***Scribe***	***MarconiNet***
Content lookup	No	On server	On server	Yes	Yes
Static download	No	Yes	Yes	(Possible)	No
Live streaming	Yes	Yes	Asynch.	(Possible)	Yes
No IP multicast	Yes	Yes	Yes	Yes	Yes
Low system req.	Yes	No	Yes	(Depends)	Yes
Adaptive stream	No	Yes	No	(Possible)	No

Based on the current knowledge it is likely that any of these protocols could be used successfully as a base for the Aeetes architecture. To make an informed selection it is necessary to perform a detailed analysis of each in relation to a detailed requirement specification. As the necessary requirement specification was not available when working on this thesis no conclusions are made here to suggest which protocol is best suited based on the capability chart in Table 4.2. Such a specification should be created before a final decision is made.

4.2.2. Ensuring quality of service

While the chosen protocol might perform just fine by itself, it might not be optimal in all circumstances if implemented as described. One thing that might easily take a performance hit under difficult network conditions or an unevenly distributed user-base is the quality of service experienced by end-users.

There are no golden solutions that will fix everything causing this, but there is a range of separate research performed on some of these problems, that could be valuable if it was to be integrated into the chosen protocol.

One such solution can for instance be the locality-awareness protocols presented in [mOverlay] where one through the use of dynamic landmarks can ensure that a peer-to-peer overlay network is not unnecessarily burdened by a lower level inefficient physical routing that could have been avoided.

Another way to improve the system is by designing better replication algorithms used to distribute the content between hosts in an efficient way. These improvements can for instance be combined in as described in [P2PMDN].

Note that some of the protocols suggested in the previous chapter already support solutions similar to those mentioned here, so which ones of these that are required will depend on the protocol that is chosen and also on the requirements specification for the system.

4.2.3. Providing control functions

One of the primary drawbacks to basing Aeetes on a peer-to-peer architecture is the limited potential offered by the distribution protocols themselves for controlling clients and media. This is due to the fact that a P2P based network doesn't usually have any central servers that can enforce the desired controls, so the commands can be ignored by the clients without reprisals.

The easiest way to deal with the lack of content controls would be to simply accept that there are no controls available in the distribution network, and instead provide these controls through other external means if they really are necessary.
This is probably also the best solution due to the uncertain nature and usefulness of the currently available content controls, as is discussed in the articles [Darknet], [Piracy] and [DRM]. For these reasons this matter will not be discussed in any more detail in this report, even if something useful possibly could be integrated into the architecture.

Client control on the other hand is a whole other matter. As mentioned above most of the drawbacks with a P2P-based architecture is the potential for clients to act as freeloaders or cause other kinds of havoc, so if administrators could have the ability to block or restrict access to the network for certain users, and also to enforce other kinds of client controls, then it would be very useful in a range of different cases to ensure efficient operation of the network.
A major problem with implementing such controls is that while they can be easily implemented in the clients provided by the network operator, those controls can also easily be bypassed by third party clients or client-patches. Depending on the expected use and popularity of the system this might not become much of a problem in reality, but having an efficient control-system in place would still be preferable for the system to be more adaptable.
While it is unclear how such controls can be implemented most efficiently, some potential solutions to alleviate the problems are presented below. More detailed research into this issue will be required to decide on the best combination of solutions for an MDN type network like the Aeetes.

- *Secure client authentication and communication*

  Having clients authenticate through a secure challenge-response mechanism and thereafter use encrypted communication could prevent unauthorized clients that do not adhere to control commands from joining a swarm. However this will not prevent third-party modifications of the official client from being implemented unless the entire client is also secured through cryptography and secure hashing. While this of course is possible it would lead the system to having a massive security-overhead that may render it impossible to use on low-capacity or older computer-systems and devices.

- *BitTorrent style capacity matching*

  For static files each client by itself can monitor the capacity it receives from another client and match it by providing as much in return. While attractive and self-regulating, this will not work for streaming media since a user will rarely receive anything in return from those clients it is forwarding a stream to.

- *Reputation-based systems*

  A solution that accepts the fact that an administrator is hard pressed to leverage direct control over end-users is to use a reputation based system where good behaviour will provide advantages to those users. This also has the desirable property of scaling with the network depending on how it is implemented. Both [XRep] and [Trust] are among the papers that detail potential implementations of reputation based trust-systems.

- *Active Virtual Peers*

  The paper [AVP] proposes the use of Active virtual peers to implement various control functions in P2P networks. AVPs are based on the concept of Application level active networking (ALAN) and allow a range of various algorithms to control different aspects of a P2P network to be implemented on a flexible and adaptive network.

4.2.4. Locating content

As became apparent from the review of existing MDN services, most of the services use a regular web-based interface to provide both content-lookup and playback functionality. While this is a simple and efficient way to perform this task it does unfortunately require some form of centralized server that can host the website and the content indexes, and in addition to both being expensive and requiring administration it does not fit naturally into a P2P based system.
However the Open Media Network does notably make use of a stand-alone application that connects to the underlying Kontiki grid-network and provides both search and browsing functionality in addition to an integrated viewer for the downloaded content. As Grid-networks are essentially an advanced variation of the P2P-architecture, a similar solution would be suitable as the end-user interface for the Aeetes MDN as well, but this requires an underlying protocol that supports distributed content discovery.

The content-discovery functionally is not required to be an integrated part of the P2P-protocol, but may also be a plugged-in third-party or independent solution. There are currently just a few such solutions in common use:

- *Request flooding*
  The most primitive way to locate content in a distributed peer-to-peer network is simply to flood requests for the desired content to all the peers the user is aware of. Those peers then flood the request further to all the peers they know of and so on. This solution is used in Gnutella and while it is easy to use it does not scale to a global network and it also consumes large amounts of time and network resources.

- *Distributed Hash Tables (DHT)*
  A major improvement over the basic flooding technique is using DHT, a variant of consistent hashing that is being used to map unique keys to all the nodes in the network. The nodes can then be "searched" by calculating where the desired content should be located in the keyspace.

- *External content indexing*
  Content descriptors like the .torrent files of BitTorrent are located externally from web-site indexes which are completely independent of the network itself. While this removes the issue from the protocol it is not suitable for reasons that have been discussed previously. Another external solution is the universal content identification system developed by the Magnet-URI project, but it is not yet widely used.

Nevertheless there has been proposed a range of novel methods to perform content searching and lookups in recent scientific literature, and some of them may be well suited to perform the tasks required by the Aeetes MDN.

- *Semantic based content search*
  [Semantic] proposes a solution to search relevant content through semantics using a Hierarchical Summary Structure and High-dimensional indexing in a Super-peer P2P network.

- *Gossiping and random replication*
  The [PlanetP] system allows for efficient search and retrieval of content through collaboration for replication of unpopular content and a gossiping algorithm that distributes content summaries of members.

- *Spatial Content Hashing*
  [SCH] provides a way to extend regular key-word searches in Distributed Hash Tables with spatial coordinates making each node responsible for the control points that hash to that node.

The suggestions above consider the case of static files that need to be located, but do not provide solutions for continuous media streams. The reason for this is that research in that area is currently lacking. However it should be possible to modify most of the suggested solutions to also work with media-streams as both kinds would require a content descriptor that can be indexed and searched.

4.3. Content provisioning

So far Chapter 4 has only been concerned with the underlying protocols that will provide the Aeetes functionality to end-users that are consumers of data. Here we will consider how the providers of content fit into the equation and how they and the content they provide can be controlled or managed.

4.3.1. Who shall be able to provide content

Related to which control functions are made available in the system there is the matter of deciding who shall be allowed to provide content through the system and also how this shall be decided and by whom.

If some kinds of centralized systems are used for allowing access to the MDN, such policies can be easily enforced by the operators at the central site as they see fit. Only authenticated content-providers will then be allowed to register the descriptors for their data-sets with the centralized system, and thus only those providers' data will be available through the system.

The opposite case is where there is no centralized system available to enforce such kinds of controls, like in the architecture proposed for Aeetes. In this case any user with the ability to create valid content descriptors will be able to publish content through the network if the descriptors can be distributed to interested parties. Since there are no centralized systems available, the provision of descriptors must also be through a distributed protocol, and thereby outside of the control of any single party.

However the existence of such data in the network will in most cases be an advantage as it increases its popularity and usage without bringing any ill effects toward the operators of the network other than increased bandwidth requirements for providing the end-user client downloads. For this reason there need not be any direct limitations on which parties will be allowed to publish content. The case where the published content is undesirable on the other hand means that there might be a need to control publishing on a per-content level as well, so this case will be discussed below in more detail.

4.3.2. What content is desirable to provide

One problem with the above conclusion is that one should not limit the availability of content from interested third-parties is that all the content they provide might not be desirable to have accessible in the network. For a network used academically this can be content containing pornography or excessive violence. In the case where this content is also illegal the matter seizes to be a technical one and as long as the network does not provide obscurity the perpetrators can easily be handled by local authorities and brought to justice. The problem is with content that cannot be handled this way because it is only politically incorrect or otherwise undesirable and not illegal in any way.

To deal with such content there should preferably be a way to limit it from being distributed through the network, or to prevent malignant users from being able to distribute anything on the basis that it is most likely undesirable. Achieving this without any kind of centralized control and content review is very hard, but there are some solutions available that might be effective.

In Chapter 4.2.3 a range of solutions for the implementation of control functions in a distributed network is proposed, most notably the use of reputation-based systems as described in [Xrep] and [Trust]. Such solutions can be designed in a range of different ways depending on how strict the designers wish that the controls should be. A very authoritarian system could require users to acquire more than a set reputation score before being able to publish anything at all, or filter out content from users with a low reputation by default, while a more forgiving system might assign reputation on a per-content level and let each data-set must build its own credibility through the reputation system, where a higher reputation for instance could allow the data-set to gain higher positions in search results. This is of course in addition to the other control functions that can be supported by a reputation based system for ensuring the operation of the network.

.

4.3.3. How the content is made available

The final part of content provisioning is defining the systems that allow users other than the content provider to identify interesting content and to locate and join a swarm where this content is available.

A unified content descriptor format that can be used to identify and access any kinds of content through the network should be identified or developed. The Magnet URI project mentioned in Chapter 4.1.2 is one solution that might be useable as a generic content descriptor.
The descriptor format should at least support these usages:

- It must allow any user to identify an exact or approximate address that may be used to connect to the swarm distributing the data-set.
- It should allow for efficient indexing so that the available content can be searched and/or browsed either through the network or externally.
- It should be content-independent so that it can be used equally well for identifying both static files of any type and streaming media content.
- It should be extensible to support future additions and changes.
- If a reputation based system is selected for network management the descriptor should allow for per-content reputation handling.

When the provided content has a descriptor that can be indexed, the system should provide some way to browse or search through the available content. The simplest solutions are either leaving it out of the P2P protocol entirely, like BitTorrent that instead makes use of an external system provide this functionality, or they are able to do this inside the client application by getting data from the P2P network itself as the Open Media Network does.
This matter too has been covered before in Chapter 4.2.4 where a range of protocols for searching through distributed content is proposed. Note that for efficiency the selection of a location protocol must be seen in conjunction with the content descriptor format being used so that they are compatible.

5. SUMMARY OF FINDINGS

This chapter contains a brief summary of the findings, conclusions and other details that have been discussed and reported throughout the previous sections of this thesis.

5.1. The definition of MDN and IMDN in short

A Multimedia Distribution Network (MDN) is a type of Content Distribution Network (CDN) that is focused on delivering streaming and static multimedia-content from a multitude of sources through a single unified interface. Many MDNs also provide facilities for self-publishing by its users, and uses transcoding and adaptive streaming to deliver live content to match the device and bandwidth available for each receiver.

There are several different types of such Multimedia Distribution Networks ranging from the dedicated end-to-end delivery networks used by current Cable-TV companies and radio-broadcasters like Disney's MovieBeam, to combinations where some signalling and transmission are done using the public Internet or mobile networks, and further to the other extreme where the public Internet is used for everything.

An Internet-based Multimedia Distribution Network (IMDN) is a system with the abovementioned functionality that is implemented using the public Internet as the sole transport system for both the media and control signals. Existing services that can be classified as IMDNs include those provided by Google Video, YouTube, the Open Media Network and CinemaNow. Because of the high costs associated with deploying or leasing separate networks to build an MDN, the only viable solution for Aeetes is to make use of the public Internet.

5.2. Overview of the potential architectures

Behind the unified front-end of an Internet-based Multimedia Distribution Network is usually a set of CDN-architectures and communication protocols that ensure delivery of the requested content to each user.

The old theory of proxy caching is a simple solution, but it is inherently not suitable for streaming media and the large storage capacities involved. For this reason proxy-caching is not considered any further for the Aeetes proposal. A replication network is similar to proxy caches but is as a somewhat different architecture better suited for multimedia delivery. Through recent suggestions such as Reflector-networks and Prism, a replication network can even support streaming media, however both of these network types require the deployment of multiple server farms which would incur large costs for both set up and operation, making them unsuitable for Aeetes.

A Service Overlay Network is a novel solution where Quality of Service can be guaranteed through making Service Level Agreements with all the network providers along the distribution chain. However this too requires dedicated server farms and has a large administrative overhead due to all the agreements and its related costs.

The only two architectures that are viable as a base for the Aeetes system is peer-to-peer networks and the evolved variant called Grid networks.
Peer-to-peer networks are based on the communal notion that all the users interested in retrieving a given type of content must donate some of their own computing resources to the cause of distributing the content further to other users. This allows peer-to-peer networks to scale very well and removes the bottlenecks caused by single servers, in addition to being free and easy to deploy. The main drawback is its limited potential for leveraging control over users. A Grid network is essentially a P2P network that also provides some additional services through the use of centralized servers, which both increases the cost and the efficiency of such networks.

5.3. Evaluation of a peer-to-peer based system

Based on the analysis performed the peer-to-peer architecture appears to be best suited to function as a base for the Aeetes system. The major advantages it brings are low cost, high scalability and simple deployment which all are very important considerations in this case, so it is a good choice despite the limited control functions.

One of the major problems with P2P networks is that since there is no simple controls that may be leveraged on individual users, it is prone to exploitation by freeloaders and other malicious people. Many P2P-networks rely on the combination of some limitations in the clients used by most non-technical users and on the good graces of the users, but in many cases this is not enough, especially when there are external forces that affect the willingness to donate resources like a threat of legal action. Such problems are caused partially by the relative anonymity and unaccountability of individual users, but it can be remedied by the use of reputation based systems or active virtual peers (AVP).

A set of requirements for the protocols to be selected have been identified from the MDN definition and the expected usage of the system when deployed. They include: Searching and browsing for content, downloading static content, streaming live and static content, independence of IP-multicast, and adaptive encoding for compatibility and heterogeneous use on old and slow devices.
Based on these requirements a range of potential protocols has been suggested and evaluated, but lacking a requirement specification a recommendation cannot be given. In addition a number of other protocols have been suggested than may be used for improving Quality of Service, providing control functions and locating content.

The chapter on peer-to-peer systems concludes with an evaluation of content provisioning challenges and attempts to answer the questions of who shall be able to provide content, which content is desirable and can be distributed and how this content will be made available to end-users. These are broad questions and a short definite answer cannot be provided in this paper, but some suggestions are made nevertheless.

6. FUTURE WORK

This report has described a proposal for a Multimedia Distribution Network. The proposal is the first stage in the work to create a new and functional platform for multimedia distribution over the Internet. The work as a whole is called the Aeetes Project and involves the completion of a number of tasks that are needed to bring the work to fruition.
These are the overall tasks that Prof. Yuming Jiang set forth at the beginning of this project:

i) Review literature efforts/solutions and explore their problems/limitations.
ii) Investigate and suggest architecture(s) for multimedia distribution.
iii) Study and propose mechanisms for routing, caching, scheduling and distribution.
iv) Investigate and design mechanisms for user identification and mobility support.
v) Design service management for the multimedia distribution network.
vi) Develop and test the platform (e.g. for IP Video broadcasting).

The work performed in relation to and presented in this thesis-report comprises a possible solution to the tasks labelled I and II, and throughout it has also given a range of pointers and ideas for the future work with part III. If the solution proposed here is suitable for the future work on the project, this leaves the completion of tasks III and above before the Aeetes Project can be considered completed and the MDN can be deployed as a useable system.

7. REFERENCES

Some references are for entries in the free encyclopaedia Wikipedia. Note that these entries may be modified at any time by any user, so refer to a version valid on the 25th of May 2006. In this report such entries were used primarily for overview information on a given topic.

[Adaptive] R.S.Ramanujan, J.A.Newhouse, M.N.Kaddoura, A.Ahamad, E.R.Chartier, K.J.Thurber, "*Adaptive Streaming of MPEG Video over IP Networks*", Proceedings of the 22nd IEEE Conference on Computer Networks (LCN'97), November 1997.

[Audible] Audible.com public website, http://www.audible.com

[Avaki] Avaki Corporation, "*AVAKI Grid Software: Concepts and Architecture*", White paper, March 2002

[AVP] T. Koulouris, R. Henjes, K. Tutschku, H. deMeer, "*Implementation of Adaptive Control for P2P Overlays*", Fifth Annual International Working Conference on Active Networks IWAN 2003, Kyoto - Japan, December 2003

[BitTorrent] Wikipedia entry on "*BitTorrent*", http://en.wikipedia.org/wiki/Bittorrent

[Cable] Wikipedia entry on "*Cable Television*", http://en.wikipedia.org/wiki/Cable_television

[Cache] Wikipedia entry on "*Cache Algorithms*", http://en.wikipedia.org/wiki/Cache_algorithms

[CDN-Perf] K.L.Johnson, J.F.Carr, M.S.Day, M.F.Kaashoek, "*The Measured Performance of Content Distribution Networks*", Proceedings of The 5th International Web Caching and Content Delivery Workshop, Lisbon, Portugal, May 2000.

[CDN-PR] Christos Gkantsidis, "*Content Distribution Networks*", Project Report, 29.10.2001

[CDN-Use] B.Krishnamurthy & C.Wills & Y.Zhang, "*On the Use and Performance of Content Distribution Networks*", AT&T Technical Document TD-52AMHL, 2001

[CoopNet] V.N.Padmanabhan, H.J.Wang, P.A.Chou, K.Sripanidkulchai, "*Distributing streaming media content using cooperative networking*", Proceedings of ACM/IEEE NOSSDAV, Miami, FL, May 12–14, 2002, pp.177–186.

[Darknet] P.Biddle, P.England, M.Peinado, B.Willman, "*The Darknet and the future of content protection*", In Digital Rights ManagementTechnological, Economic, Legal and Political Aspects. LNCS 2770, Springer, 2003, pp 344-365.

[DHT] Wikipedia entry on "*Distributed Hash Tables*", http://en.wikipedia.org/wiki/Distributed_hash_tables

[DRM] D.K.Mulligan, J.Han, A.J.Burstein, "*How DRM-based content delivery systems disrupt expectations of "personal use"*", Proceedings of ACM Workshop Digital Rights Management, 2003, pp. 77--89.

[FR] Cherkasova, Ludmila & Lee, Jangwon, "*FastReplica: Efficient fast file replication within Content Delivery Networks*", 4th Usenix Symposium on Internet Technologies and Systems, 26.03.2003

[Freenet] I.Clarke, O.Sandberg, B.Wiley, T.W.Hong, "*Freenet: A Distributed Anonymous Information Storage and Retrieval System*", In Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability. Springer, New York, 2001.

[GV] Google Video public website, http://video.google.com/

[HBO] HBO OnDemand public website, http://www.hbo.com/hboondemand/

[Hybrid] Y.Shan, S.Kalyanaraman, "*Hybrid Video Downloading/Streaming Over Peer-to-Peer Networks*", Proceedings of International Conference on Multimedia and Expo (ICME), Vol II, Pages 665 - 668, Baltimore, MD, July 2003.

[Tradeoffs] F.Schaffa & J.-P. Nussbaumer, "*On Bandwidth and Storage Tradeoffs in Multimedia Distribution Networks*", Proceedings of Infocom'95, 1995, pp. 1020-1026.

[iTunes] Apple iTunes public website, http://www.apple.com/itunes/overview/

[Kontiki] Kontiki Inc, "*The Kontiki Grid Delivery Advantage*", White Paper, 2005. http://www.kontiki.com/technology/print/Kontiki_Digital_Video-On-Demand_White_Paper.pdf

[Magnet] Gordon Mohr, "*Magnet URI Draft Overview v0.1*", 17.06.2002, http://magnet-uri.sourceforge.net/magnet-draft-overview.txt

[MB] Disney MovieBeam public website, http://www.moviebeam.com/

[MN] A.Dutta, H.Schulzrinne, "*MarconiNet: Overlay Mobile Content Distribution Network*", IEEE Communication Magazine, February 2004

[mOverlay] Q.Zhang, F.Yang, W.Zhu, Y.Q.Zhang, "*A Construction of Locality-Aware Overlay Network: mOverlay and its performance*", IEEE JSAC Special Issue on Recent Advances on Service Overlay Networks, Jan. 2004.

[Napster] Wikipedia entry on "*Napster*", http://en.wikipedia.org/wiki/Napster

[Osmosis] S.Jin & A.Bestavros. "*OSMOSIS: Scalable Delivery of Real-Time Streaming Media in Ad-Hoc Overlay Networks*", Proceedings of IEEE ICDCS'03 Workshop on Data Distribution in Real-Time Systems, 2003.

[P2P] Wikipedia entry on "*Peer-to-peer*", http://en.wikipedia.org/wiki/Peer-to-peer

[P2PMDN] Z. Xiang, Q. Zhang, and W. Zhu, "*Peer-to-Peer Based Multimedia Distribution Service*", IEEE Transactions On Multimedia, Vol. 6, No. 2, April 2004

[P2P-survey] S. Androutsellis-Theotokis & D. Spinellis, "*A Survey of Peer-to-Peer Content Distribution Technologies*", ACM Computing Surveys, Vol. 36, No. 4, December 2004, pp. 335–371.

[PD] Wikipedia entry on "*Prisoner's Dilemma*", http://en.wikipedia.org/wiki/Prisoner's_dilemma

[Piracy] S.Haber, B.Horne, J.Pato, T.Sander, R.E.Tarjan, "*If Piracy Is the Problem, Is DRM the Answer?*", Digital Rights Management — Technological, Economic, Legal and Political Aspects, Eberhard Becker, Willms Buhse, Dirk Günnewig, Niels Rump (Ed.), Lecture Notes in Computer Science, Springer-Verlag, Berlin, Germany, Lecture Notes in Computer Science, Vol. 2770, November 2003

[PlanetP] F.M.Cuenca-Acuna, R.P.Martin, T.D.Nguyen, "*PlanetP: Using Gossiping and Random Replication to Support Reliable Peer-to-Peer Content Search and Retrieval*", Technical Report DCS-TR-494, Department of Computer Science, Rutgers University, 2002.

[Prism] C.Cranor, M.Green, C.Kalmanek, D.Shur, S.Sibal, Cormac J.Sreenan, "*PRISM Architecture: Supporting Enhanced Streaming Services in a Content Distribution Network*", IEEE Internet Computing, pp. 66–75, IEEE Computer Society Press, 2001

[Proxy] Wikipedia entry on "*Proxy server*", http://en.wikipedia.org/wiki/Proxy_server

[Reflector] F.Kon, R.H.Campbell, K.Nahrstedt, "*Using dynamic configuration to manage a scalable multimedia distribution system*", Computer Communications 24 (1) (2001).

[Replication] S.Sivasubramanian, M.Szymaniak, G.Pierre, M. van Steen, "*Replication for web hosting systems*", ACM Computing Surveys, 36(3):291--334, Sept. 2004.

[Resilient] V.N.Padmanabhan, H.J.Wang, P.A.Chou, "*Resilient peer-to-peer streaming*"**,** IEEE Int'l Conf. Network Protocols, Atlanta, November 2003.

[SCH] A.Harwood & E.Tanin. "*Hashing spatial content over peer-to-peer networks*", Proceedings of the Australian Telecommunications, Networks, and Applications Conference-ATNAC, Melbourne, VIC, December 2003.

[Scribe] M.Castro, P.Druschel, A.M.Kermarrec, A.Rowstron, "*SCRIBE: A large-scale and decentralized application-level multicast infrastructure*", IEEE JSAC, 20(8), October 2002.

[Semantic] H.T.Shen, Y.Shu, B.Yu, "*Efficient Semantic-Based Content Search in P2P Network*", IEEE Transactions on Knowledge and Data Engineering, Vol.16, no.7, pp. 813-826, July 2004

[SON] Z.Duany & Z.-L.Zhangy & Y.T.Hou, "*Service Overlay Networks: SLAs, QoS and Bandwidth Provisioning*", Proceedings of the 10th IEEE International Conference on Network Protocols, 2002

[TiVo] Wikipedia entry on "*TiVo*", http://en.wikipedia.org/wiki/TiVo

[Trust] A.Selcuk, E.Uzun, M.R.Pariente, "*A reputation-based trust management system for P2P networks*", Fourth International Workshop on Global and Peer-to-Peer Computing, April 20-21 2004.

[VoD] Wikipedia entry on "*Video On Demand*", http://en.wikipedia.org/wiki/Video_on_demand

[WhatIs.com] Whatis.com definition for "*Content Delivery*", http://whatis.techtarget.com/definition/0,,sid9_gci214325,00.html

[XRep] E.Damiani, D.C. di Vimercati, S.Paraboschi, P.Samarati, and F.Violante, "*A reputation-based approach for choosing reliable resources in peer to peer networks*", Proc. 9th conference on computer and communications security 2002, pp. 207-216.

[ZigZag] D.A.Tran, K.A.Hua, T.T.Do, "*A peer-to-peer architecture for media streaming*", IEEE JSAC Special Issue on Advances in Overlay Networks, 2003.