Acreo
Research and development
in electronics, optics and
communication technology.

acreo

Report

Peer to Peer based Video service delivery in access network

Mozhgan Mahloo

August 2010

Peer to Peer based Video service delivery in access network

Mozhgan Mahloo

Kista • Norrköping • Hudiksvall • Jönköping
2010-08-

Report no: acr-xxxxx [Ev rev. nr]

Acreo AB
Isafjordsgatan 22
Electrum 236
SE-164 40 Kista
www.acreo.se
info@acreo.se

Bredgatan 34
SE-602 21 Norrköping

Håstaholmen 4
Box 1053
SE-824 12 Hudiksvall

Jönköping Science Park
Box 1026
SE-551 11 Jönköping

Abstract

Video rental services and TV broadcasting are encountering a revolution nowadays. Video on demand (VoD) services and IPTV are making their way into today's market. Now it's time to think of a way to fulfill the growing demand for these services, in a way that satisfies both the needs of end users and providers. Video on Demand using the pure client-server based model is a mature technology which has its own benefits and problems. The main problem with available VoD services is scalability, which limits the number of end users a provider can serve without increasing the set up cost. The alternative solution that is in use by Internet users is Peer to Peer (P2P) file sharing. It solves, the scalability problem of the client-server based model, but has other limitations such as the lack of service insurance. For these reasons, a number of researchers have proposed using a hybrid of client-server based and P2P models, to cover the shortcoming of each model with using characteristic of the other. In this work we add the P2P technology using a BitTorrent based application to an existing Video on demand service. Using P2P solves the scalability problem while the VoD system guarantees the quality of service and the seamless availability of contents.

Another novel interesting video service is the IPTV. Due to major changes in people's life styles, TV in the way it used to be with the exact timing is not always a suitable solution any more. It is hard to catch-up all your favorite TV shows on time. An available way out of this difficulty is to use a VoD server to download your favorite show after-wards, but it is not an efficient solution due to the shortcoming of client-server based model. An alternative way for the user convenient is using the set top boxes (STB) with catch-up capability. Using a combination of IPTV (multicast streaming), video servers and P2P technology; we investigate and test an efficient solution both for end user and providers.

We investigated the difference in network traffic between distribution methods for various video services by simulation. Network simulator 2 (NS2) was used to that effect, and a number of modules were written or modified to generate and receive data in a way suitable for the video services that we studied . In the former case we compare a pure VoD service with a hybrid (client-server and P2P) model in an access network with four levels of hierarchy. This part contains three simulations; in the first scenario all nodes download their favorite show directly from the VoD server, while in the two other scenarios users download the movie from their peers if there are any. The difference between the last two scenarios is the popularity of requested movies, number of available seeds, and movies. These simulations show, that the efficiency of our model is related to the amount of available data in the peers, though both scenarios are still more efficient than pure VoD model.

We also conducted a series of simulations for IPTV services. The first simulation is used as the reference and implements the existing solution: There are IPTV channels broadcasting in a regular basis, and there is a VoD server for the users who are willing to watch the show later.

This requires a high capacity server with enough upload bandwidth to serve user requests. In the second scenario, we introduce the patching method, which allows users to start watching a show after it has begun, joining the multicast stream and storing the latter part of show received via IPTV while downloading the former part from the VoD server. A considerable decrease in the server load can be seen from the result. Although the second scenario has some benefit over the first one, still there is no benefit for the users who want to watch TV show after the broadcast time, since they have to download the whole show from the server. So in the third simulation we add P2P capability to the nodes to assist the IPTV servers for catch-up. The nodes who watch TV shows on time store the content in their STB and share it with other peers later on. This distribution scheme decreases even more the load on the VoD server and keeps the traffic local, close to the end users.

Acknowledgements

It is a pleasure to thank those who made this thesis possible. In the first place I would like to record my gratitude to Stéphane Junique for his supervision, advice, and guidance from the very early stage of this research. I am indebted to him more than he knows. Special thanks also to Kjell Brunnström, who gave me the opportunity to work on this project and for the support he provided me through this work.

I gratefully thank Professor Peter Sjödin for his constructive comments on this thesis. I am thankful that in the middle of all his activity, he accepted to be my examiner.

I also wish to express my love and gratitude to my beloved family; for their understanding and endless love, through the duration of my studies. I would also like to show my gratitude to my dear friends, without whose love, encouragement and editing assistance, I would not have finished this thesis.

Table of contents

List of figures

List of tables

I Introduction

1 Motivation

The number of people who are willing to use new internet based video services is growing rapidly due to introduction of new and user friendly computer technologies and devices such as set top boxes (STB), increasing end user bandwidth, as well as a variety of options in TV programs and movies. To satisfy this growing demand of internet users and attract new users, new services should be introduced to the market. Each of the services which are in use nowadays, such as Video on demand (VoD), IPTV, and peer to peer (P2P) file sharing has its own benefits and problems. This leads us to search for new services that meet all the user needs such as simplicity for normal users, scalability to serve an increasing number of users, quality assurance and so on. On the other hand the new solution should be attractive for the providers and give them enough motivation for switching to this new technology considering the huge amount of installation and set up expenses in the beginning, and it should keep the cost low for the end users.

The traditional method of providing services in general and video services in particular over the internet and in access networks is using the client-server based model. Depending on the type and coverage area and the number of customers, video service provider can have one server, or use a Content Delivery Network (CDN). In this method, users pay to download their favorite videos from the provider's servers. Using servers which are managed by the service providers allows enough control over the content and limits copyright problems. Downloading from a known server also guarantees the quality and authenticity of the content. Despite these advantages, the client-server model has some drawbacks. One of its major problems is the scalability. The number of customers that can be served by a provider is bound to the server capacity. Using one server limits the size of services, and CDNs are very expensive. Also, the capacity and bandwidth of the servers should be designed for the peak load of the network, which increases the set up cost of the service as well as wastes the resources during normal times. All these factors are even more critical for the live streaming video services which this report focuses on.

An alternative available solution for live video streaming via internet is P2P applications that are slightly different from file sharing methods: for live streaming, the order in which the various pieces of a file are downloaded is important, at least for the first several minutes of the video. The low cost of P2P systems is one of the interesting features of this technology. In a P2P application, each end user acts both as the server and client and there is no need to install expensive servers. Also, since every node can provide data there is no single point of failure, which makes P2P networks quite robust.

Another interesting feature of P2P distribution is using the upload bandwidth of the end users. In this way we are using available resources that is normally unused by other internet applications instead of adding resources such as server capacity or bandwidth to the network. This helps keep the cost low.

P2P data can generate very expensive traffic for internet providers: since there is no standard way of keeping the traffic local, data can be downloaded at high cost from outside the internet provider's network, even though it is available in the neighborhood. So providers are likely to try to block P2P applications, and they are likely to be worry of implementing P2P-based solutions in their networks, unless they have tight control over data routing. Another potential weakness of P2P-based solutions is that they do not play well with firewalls: end users sitting behind a home router cannot receive incoming connections from their peers without special configuration of the firewall, meaning that they cannot upload data.

IPTV [1], which is also part of our study, is a real-time video distributions service competing with cable and satellite TV. The bandwidth usage of the IPTV is easily predictable since it uses the multicast protocol to transfer content from the source (IPTV server) to the end users. The IPTV bandwidth is different depending on the quality level and the video compression method. It is worth mentioning that our study is not about the IPTV technology but rather about the catch-up capability for IPTV users: It is hard for users to manage their time in the way that always allows them to watch their favorite shows on time. So time shifting of TV programs is an interesting feature for the internet users. The methods currently available are to watch missed TV programs later using either a Video on demand service or with P2P technologies, which are not efficient.

In this thesis we are trying to present a solution for live video streaming services that is suited for both providers and users needs. Our proposal is a combination of a client-server based model and a BitTorrent-like peer to peer application with minor changes in the download strategy. Using a hybrid solution allows us to exploit the advantages of both methods without having the drawbacks. We also use this hybrid model together with an IPTV service to provide a more efficient catch-up service.

2 Problem formulation

Real-time video distribution services are attracting new customers every day, while available technologies are not enough to fulfill this growing demand. So a new solution should be proposed to serve all internet users. The solution should be based on the available technologies to use the resources already exists in the access network to keep the capital expenses as low as possible for both providers and end users. The existing technologies such as client-server or the P2P application that are used nowadays are not suitable.

The main motivation of this project is to find a suitable method for real-time video services using a hybrid of IPTV, VoD and P2P technologies and test it using simulations. Our study is focused on access networks, partly because they are sensitive to the cost, and partly because the network operator has full control over which technologies are implemented.

During this report we propose our solution based of our literature study using available techniques. We then show the effectiveness of the proposed solution by running several simulations using the NS2 network simulator. We evaluate the efficiency, scalability and resource saving of our solution with the simulations. We present a quantitative comparison between available methods and the proposed one to highlight the effectiveness of the new method.

3 My Contribution

The thesis is focused on defining and testing a solution for live video streaming, and the concept presented here was validated by simulation. The author's main contributions to this project are as follows:

1. Definition of the proper peering strategy: how to build an efficient and resource-saving swarm, which communication messages to pass between network nodes, and so on.
2. Development of application-level modules of the network simulator in C++, including full implementation of the IPTV server module, substantial modifications to the P2P tracker module (implementation of the peering strategy), and a number of bug fixes.
3. Determine and implement user profiles that represent end user download behavior.
4. Design the network topology and build it using the Tcl scripting language in NS2.
5. Define scenarios which cover different aspect of the experiments. Run the simulations and analysis the result.

4 Outline

Chapter 2 summarizes the ideas which are used as the basic information of the project. The chapter starts by a short review of client-server model, followed by a description of the peer to peer technology. Different solutions proposed in the literature are presented. Also some information is provided about IPTV and its connection to our work. We end chapter 2 by presenting the result of a study about user behavior that is used to define our user profiles for the simulations.

Chapter 3 presents the tools we are using in this project, and discusses our implementation and the physical topology of the access network in more details.

Chapter 4 introduces the scenarios that were implemented in the simulation. The reasons why we chose each scenario and what we aimed of each specific scenario are discussed. This chapter also includes the simulations result and their analysis.

Finally, chapter 5 concludes the work, and some suggestions are provided for the continuation of this project as future work.

II Literature study

This chapter summarizes the state of the art in terms of deployed solutions and published ideas and concepts related to the distribution of video material by Internet. The first section presents the state of the art for the client-server architecture. Section 2 presents the P2P architecture. Section 3 presents the so-called hybrid architecture, which combines the client-server and the P2P architectures. Section 4 presents IPTV. Finally, section 5 presents a user behaviour study that we have used to create end user profiles.

1 Client-Server model

Nowadays Media streaming is mostly done using the client-server design. In a large streaming system where user requests arrive at a high rate, a server has to support a large number of concurrent streaming sessions [2]. When the maximum number of users that can be served by the provider is reached, more investment is in order: both operational and capital costs increase with the number of concurrent users. One of the available solutions which address the scalability problem is Content distributions networks (CDN). Multiple servers or proxies can be deployed to increase the total system capacity. In this design, media content is replicated on these proxies and clients receive streaming data from the closest proxy [2]. Figure 1 presents an example of client-server model with local proxies.

CDNs such as Akamai, Limelight, Kontiki, and SAVVIS provide video delivery bandwidth and local proximity caching of rich media for websites. The CDN’s ensure that the actual video that consumers want to watch is close to their actual physical location [3].

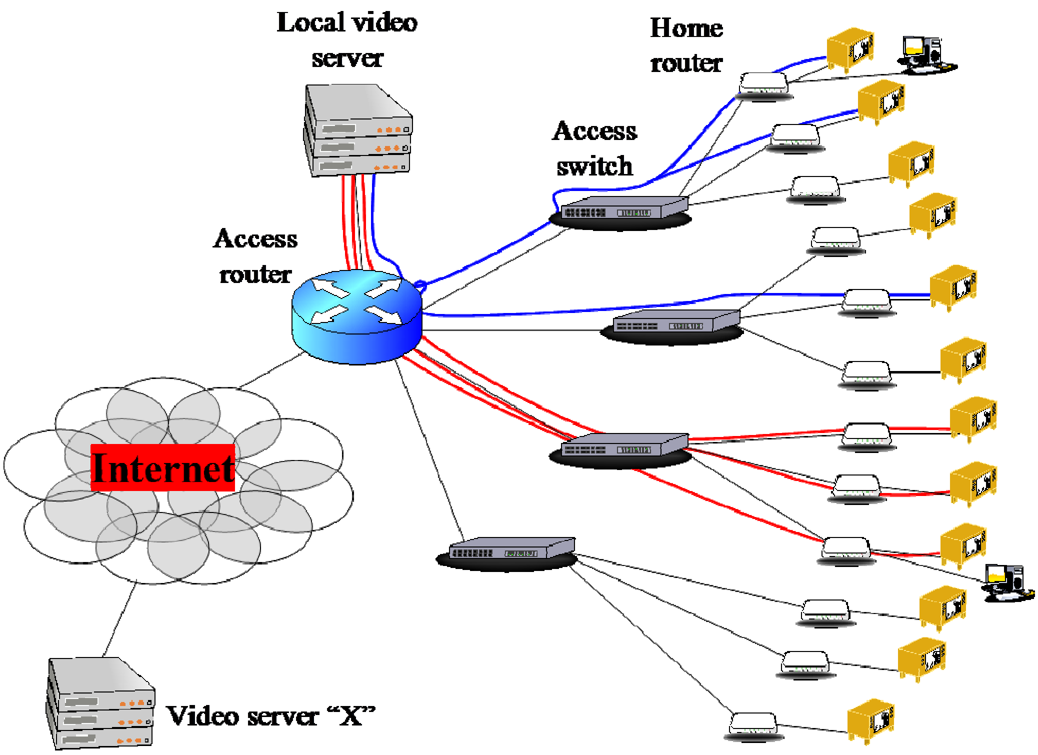

Figure 1: Illustration of the CDN network.

Download from an official address of the CDN servers guarantees the authenticity and the quality of content. By using caches in the network we keep this benefit and reduce network load while increase the provider capacity to serve more end users [4].

But there are some drawbacks with CDN solution: the cost of maintaining a CDN is high considering the massive CPU power, storage space and bandwidth needed [4] that stimulate researchers to search for more efficient solutions. One approach to solve the above problem is motivated by the concept of peer-to-peer (P2P) distribution combined with client-server based systems [2].

2 Peer-to-peer architecture

Peer-to-peer (P2P) architectures for file sharing are generating an increasing fraction of the traffic on today's Internet and are reshaping the way new network applications are designed. Statistics show that P2P applications use 55% of the downlink capacity of links [5].

P2P was designed with symmetry in mind. The idea is that participating hosts play dual roles as servers and clients [6]. By this mean the upload bandwidth of the users can be used to support more nodes instead of having a server with higher storage capacity and upload bandwidth in client-server architecture. In a research published by the "World Internet Institute" in 2009, statistics show that P2P applications use around the 70% of the uplink traffics of end users in Sweden [5].

Even though many P2P file-sharing systems have been proposed and implemented, only very few have stood the test of intensive daily use by a very large user community. The BitTorrent file-sharing protocol is one such system, and it is the most common P2P protocol implemented in file sharing applications [7].

A BitTorrent file sharing systems is a combination of different components including a tracker, a torrent file, peers, seeds and torrent application. Each file is broken up into the constant number of the pieces called chunks that client application downloads them. A torrent is a file containing IP of tracker, meta-data about main file including file name, size, number of chunks, and checksum of all pieces that are needed for reassembling the chunks later. Tracker is a node that has a list of available peers with their content and IP address. Each peer contacts the tracker to ask about the seeds (source of download) IP addresses that have the requested files by node, and then try to establish a TCP (Transmission Control Protocol) connection to the peers and start downloading the chunks. When a peer has finished downloading a file, it may become a seed by staying online for a while and sharing the file for free, or it may leave the swarm (the group of peers sharing the chunks of a given file), as soon as it has finished downloading [7].

Since P2P applications do not have information about the physical topology of network, data exchange does not take peer locality into account and P2P customers are regarded as expensive customers [8]. That is why some of the network provider tries to filter P2P application or control them. Locality aware content fetching is hardly accessible without knowing the physical topology of the network. A close collaboration between the network provider and application developers is needed to achieve this factor. This cooperation can be beneficial for both providers and users. Providers can have more control over the P2P traffic to decrease the cost of the traffic to the other providers, and users can enjoy a more reliable and faster service [9].

Studies show that locality awareness content fetching can increase the efficiency of P2P networks. Dividing end users to different group regarding their physical location can lead to a faster exchange of chunks in networks [10] [11]. But there are other factors which should be considered for peer selection in addition to the location. For example the end users may be grouped based on their interest which may optimize the search time for finding the peers [12].

Some people have worked to create locality-aware P2P applications. The most famous proposal is called "Provider Portal for P2P (P4P)". This method enables better cooperation between P2P and network providers through explicit communications. The P4P framework consist of a control plane include an itracker and a data-plane (Figure 2). The itracker node facilitates communication between the P2P applications and the network providers and normally belongs to the service provider. This node maintains the information about network status and topology, provider policies and the network capabilities that can be used by P2P application to optimize P2P traffics. In the data plane, routers give a fine-grained feedback to the P2P and allow more efficient usage of internet resources. For example end users can adjust its TCP flow rate regarding the routers feedback about the network capacity [9].

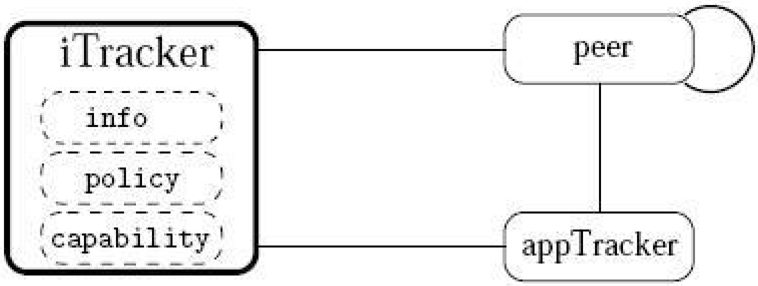

Figure 2: iTracker interfaces and information flow [9].

Since the order of download is not important in available P2P file sharing applications, they are not suitable for the live streaming applications. In the real-time services, the chunks should be downloaded in order at least for the first several seconds of the video to start playback with minimum start-up delay. Several live streaming proposals for P2P application are introduced in [13] [14] using the multicast streaming in overlay networks, which is not our concern.

There are some factors that should be considered in the design of a P2P network for live video streaming. The number of peers for each swarm is an important factor that is different depending on the protocol and applications. Experiments show that having more than 15-20 peers, has no benefit regarding efficiency of live streaming P2P application. Also the number of chunks of a file will affect the efficiency of the application; the more chunks available, the more efficiency and more reduction in the server load, though there is a limit on the maximum number of chunks for each file. Increasing the fragment number will increase the lag which user will experience in live streaming [8].

One of the benefits of P2P applications is using the upload bandwidth of the users, which is wasted in normal internet usage. But it makes P2P application sensitive to the upload capacity of the links. By increasing the upload bandwidth of end users, each node can seed several other nodes, increasing the efficiency of P2P systems [15].

In general despite benefits of pure P2P applications such as their low cost, scalability of the system and being fault tolerance, it is hard to reach all the above specification with a pure P2P system. Here are some of the drawbacks of pure P2P systems, which drive researchers toward a hybrid method.

a) There is no guarantee for content delivery [4].
b) Clients with high bandwidth normally leave the system shortly after completing their download [4].
c) Clients which are unable to accept incoming connection request, such as clients behind firewall, will suffer from a reduction in download speed [4].
d) There is no mechanism for source authentication which makes P2P network unsecure [16].

P2P set top boxes are a new development to the market that makes it easier to combine technologies like P2P, online video distribution and TV. Introducing P2P-set-top-boxes to the living rooms offers an enhanced ease of use and limits the perception of P2P being a computer related file sharing activity, due to the small device size, which does not remind of computers anymore. STBs are offered by P2P-providers such as Sopcast, Vudu, Vatata, and TVU so far. To become a mass market phenomenon a couple of premises still have to be met or proven, like ease of use, legality, support by known manufacturers brands and seamless integration in the home entertainment environment. Certainly standardization will help to become relevant to increased numbers of users [3].

3 Hybrid method for Video on Demand services

As was said before, neither the client-server nor the P2P models are good enough to be used on its own for modern video services. Some may think there are other options such as using application level multicast instead of the hybrid system, but this method has its own drawbacks: it does not scale very well, and more importantly, it is not very robust, as when a peer disconnects in the middle of the tree, all the branches under it lose their streams [8]. The Figure 3 illustrates an application level multicast system. As can be seen, when node B1 leaves the network, all nodes below it are disconnected from the stream. Some methods such as SplitStream [17] or Probabilistic Resilient Multicast [18] mitigate this problem.

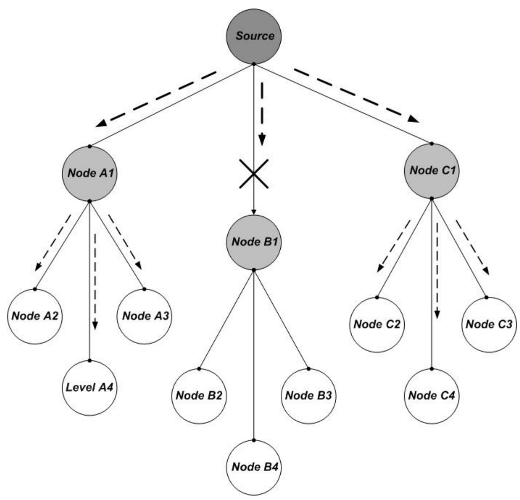

Figure 3: Application level multicast tree.

In this part of the report we are trying to find an optimal solution combining a client-server model and a P2P application. But before we start the solution lets define the specification we need for a live video streaming services. One of the important criteria is to have a low start up delay, since users normally do not like to wait too long before they can watch the video they are requesting for.

Another interesting characteristic for the users is the ability of going forward and backward through the video during playback. So having the VCR functionality with minimum delay is necessary for a good video streaming service. On the other hand, as it is said, providers are not interested in P2P application since they do not have control over the routes traffic taken from source to destination. This makes the P2P traffic so expensive for the internet providers. As a result the locality aware content fetching is interesting to attract the support of the internet providers in our solution [19].

At the very basic level our solution contains a P2P application working in parallel to the video on demand server. Using P2P within a community increases the number of concurrent viewers served by a local server. Regarding the access network parameters that influences the solution, it is noticeable that by increasing the upload bandwidth of end users, the provider can serve more concurrent peers with P2P, though increasing the upload bandwidth more than encoding rate has no benefit on the efficiency of application[20]. In some cases where increasing the upload bandwidth is not economical, it is possible to compensate its effect on the service by increasing the storage time after completion of the download, to support the same number of end users [19].

At the user level, each end user has a STB containing the application needed for our service. As the seed time is an important factor in the efficiency of our services, we suppose that STBs have ample storage and are continuously available in the network, even if the owner is not watching a movie [19]. An interesting implementation for the STB application can be found in paper [19] where the authors propose that each STB have two modules: a sliding window and a pre-fetching module. The proposed model is depicted in Figure 4.

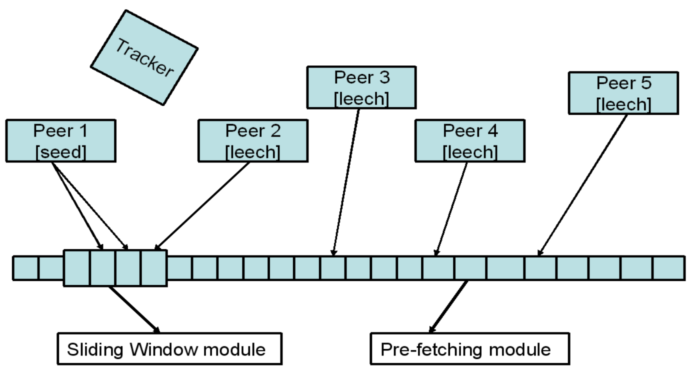

Figure 4: An example of Video-On-Demand architecture [19].

The sliding window is responsible for downloading the chunks near the playback time while the pre-fetching module downloads the rest of the file using the fact that the download bandwidth is more than the video streaming rate. Both modules store data in the same buffer which is accessed by the playback application. The sliding window first checks the local cache for the chunks. If they are not available, it will try to retrieve them from peers, and if the chunk deadline is near, it is downloaded from the VoD server. A sliding window of 300 seconds is a proposed [19].

TOAST [21] and BASS [22] are two methods that proposed a download policy in P2P networks for video distribution. In the BitTorrent protocol, peers tend to download the rarest chunks first, to increase the availability of the chunks in the network. This is not necessary in a hybrid system like toast: since the VoD server has all the chunks, there is no need to worry availability.

In this method, the downloading application asks about the playback position and divides the reminder of file in three parts: the first part is very early slices which are downloaded from the VoD server directly; this part can be considered equivalent to the sliding window in the previous method. The second part downloads from peers but in order; and in the third part slices are requested randomly to the peers [21]. The download strategy of TOAST method is shown in Figure 5.

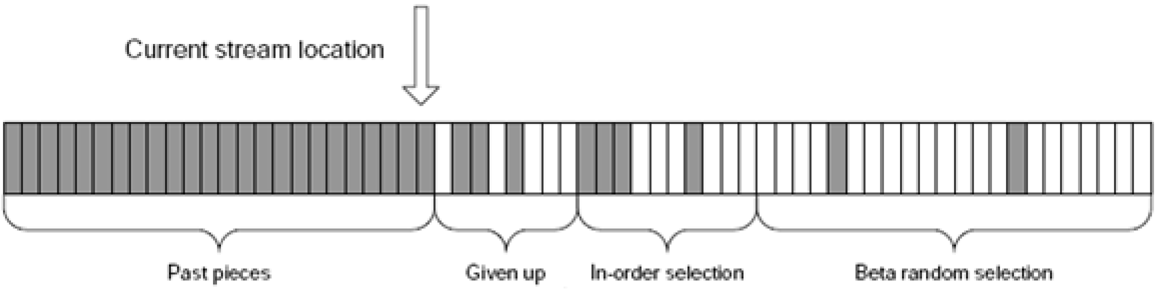

Figure 5: Illustration of the download strategy in Toast method [21].

As was mentioned before, an important functionality of live streaming is the VCR functionality during playback. One way to achieve this functionality is described in [19]. The idea is to download small chunks of video (5 seconds every minute) with a high priority or during the set up time at the beginning. If the user fast forwards the show, the application stops in the nearest 5 seconds that already has been downloaded, starts to play back these 5 seconds while it downloads the rest of the show from that point [19].

There are some implementation of hybrid model such as VoDdler, Vudu, Spotify (Audio), that work mostly as Local or regional providers. VoDdler is a commercial VoD service and uses hybrid P2P system. It was developed by the Swedish corporation VoDdler Inc [23].

The VoDdler player connects to the user's Internet connection to deliver movies, users select the movie that they want to watch then a few advertisements clips start to play before the selected movie starts playing. The first several seconds of the movies available to users are pre stored on the VoDdler player on user hard disk to decrease the time which user has to wait before show starts. The rest of the movie is downloaded via a special P2P network from other user's players in the network [23].

4 Multicast streaming and IPTV

Another remarkable video service on the internet is IPTV. There are lots of the TV channels broadcasting both in their traditional way and over the internet. IPTV uses network layer multicast [24] to reach the end users, which introduce a fixed load to the network. Because it uses multicast at the network protocol layer, IPTV in the way it exists today is the most bandwidth economical video service.

Furthermore, the multicast stream of IPTV channels can be used by catch-up services where users come late to a show that they want to watch from the beginning. As IPTV shows are sent by a service provider using a multicast stream for each channel, users can start watching shows at different times. But a user may arrive late and still want to watch a show from the beginning. In some cases, shows are available as a VoD service shortly after the show is over, but the user may not want to wait either. What he can do is subscribe to the multicast stream and store the latter part of the movie, in the STB local cache. The former part of the show is downloaded from the peers (mostly its neighbors), who are also watching the show and have a local copy of it.

The concept of retrieving the latter part of the show from the IPTV multicast stream, and the former part by other means is called patching [25][26]. It is a very bandwidth-efficient solution for catch-up TV. Patching (Figure 6) can be used for the VoD services also. If a show becomes very popular, then server can start sending it as the multicast traffic by receiving first user request and later users use patching method to download the whole movie [25] [27].

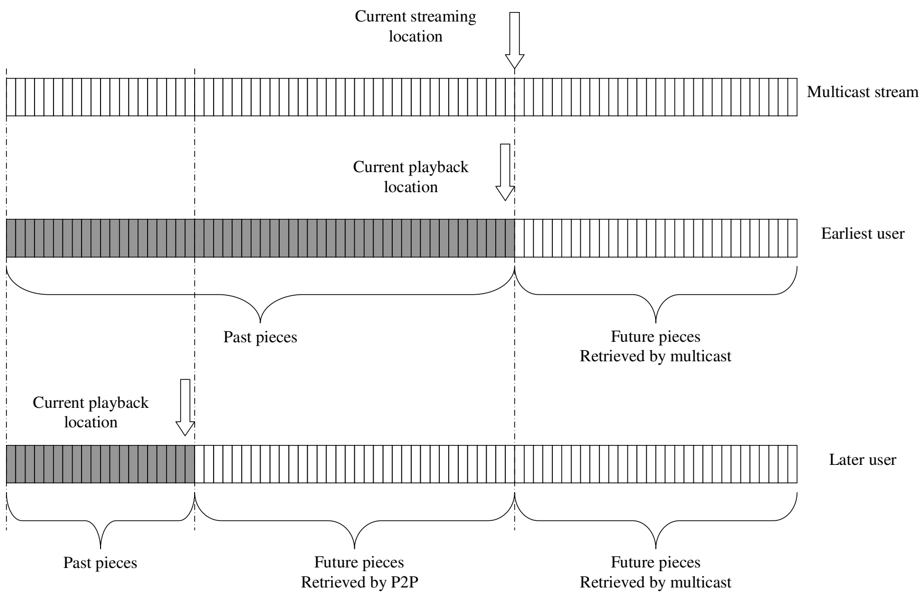

Figure 6: Illustration of Patching method.

Below are two real implementation of IPTV service using P2P distributions, which show the efficiency of the combination of IPTV and P2P. One of the first tested hybrids of P2P and IPTV called Zattoo which was tested in 2006 for the first time. Currently it serves user in 8 European countries with 60 TV channels [3].

Octoshape is another example of TV stream using the P2P that is belongs to a Danish provider. Octoshape P2P servers use P2P grid technology to minimize bandwidth usage of the end users [28].

5 User behavior study

The server capacity, bandwidth usage of the links, amount of data available in nodes as the seeds are designed considering the user behaviors. So we did some study to gain a rough idea of the end users behavior which leads us toward a more realistic result in our simulations.

The Table 1 summarizes the study results from researches done by several statistical institutes and research companies in Sweden [29] [30] [31]. The columns show different TV show options and the interest of each age group on them. The last column shows the average amount of time that each group spent on watching TV shows or video on demand per day. The rows in the Table represent the Swedish population divided by age and the occupation. We used this Table to define the characteristics of the user profiles.

Table 1: Statistics on Swedish population's interest about TV programs.

		VoD	TV movie	Series	TV sport	Children program	TV news	cultural program	Average hour/ day
child (< 15) (20%)									1 hour in the evening
Young	student with family(35%)								1.5 hour per day
15-34	student & work (10%)								2 movie per week
48%	Work - live alone (55%)								
Midle age	parents (33%)		weekends						1.5 hour per day
35-64	non-parents (15%)		2 movie/ month						
Old (> 65) (20%)									2 hours per day

Moreover, we did some research on the movies life time and popularity and their influence on the efficiency of our hybrid model. Although there are plenty of researches in this issue, there is no specific data available. The long tail theory is one of the famous hypotheses in this area that we focused our study on it to define some of our profile characteristics. Generally speaking the long tail believers say that there is a decrease on the demand for hits movies while the demand of niches(less popular) movies increase over time. Naturally when the product variety is large, the demand for each product tends to be smaller than when the product variety is small [32] [33].

The definition of hits and niches varies with time, as both the product variety and consumer base vary. There are two methods for defining movie popularity: absolute and relative. When movie popularity is measured in absolute terms, a specific number of movies are considered as the hits and the rest are niches (e.g. top 10 movies are hits). When using relative popularity, the growing number of products is in mind, and instead of exact numbers, a percentage of the available movies are used (e.g. the top 1% of available movies is considered as hits). Research shows that, there is only partial evidence to support the long tail effect by measuring the popularity in absolute terms [32].

The new movies are usually popular at their arrival to the market and their popularity decrease over time [34]. The period between the time when a movie enters the market till the time when there is no demand for it, called movie life cycle [35]. Anderson [36] mentioned that "over 95 percent of the movies (niches movies) generate the 80 percent of the revenue of a Video on demand services in their studies".

Most of the proposals about movie popularity follow Zips-like distributions [37] [38] [39]. Since it is not related to this report, we do not discuss about it more. Although we did not use the distribution in our project, we used the general idea about hits and niches movie and the popularity definition of long tail theory.

6 The implemented system

So far there is an approximate solution for all of our required characteristics. We can use hybrid method to have an efficient video on demand services with locally aware content fetching. Also we can offer catch-up IPTV capability using the patching method.

III Design and implementation

1 Overview

So far we present an overall view of the work has been done during this project. As you may know by now, this report discusses the two different video services: IPTV with catch-up capability and Video on demand as a combination of P2P and client-server based model. From now on, we call the former one catch-up IPTV and the later one hybrid system.

In this chapter we present the design structure and implementation of the project in more detail. At the beginning of this chapter there is a short introduction about the tool we used for the simulations, followed by a presentation of the network architecture and the main components of the network with their characteristics. The chapter ends with information about my contribution and the implementation part.

2 Introduction to the simulation tools

Since migration to new methods normally required a capital expenses at the beginning, it is not always easy to convince a network owner or a service provider to switch to new solutions. In the real world it is too expensive to experiment with every new solution without being sure of its performance efficiency, so there is a need to employ computer-based tools such as simulation to support your proposal with quantitative measurements. Using simulation, all the factors which can influence the efficiency or may cause a problem can be tested. Simulations can also expose interdependence problem between service events, and help tracks them individually. It can reveal conflicts which would not be visible during the theoretical phase of the project. So, we run simulations to support our solution both qualitatively and quantitatively.

Network Simulator 2 (NS2) was used as the simulation tools in this project. NS is one of the most popular choices of simulators used in research papers and since it is an open source simulator, it's constantly updated by its large user base. It implements network protocols such as TCP and UPD, traffic sources such as FTP, Telnet, Web, CBR and VBR, router queue management mechanisms such as Drop Tail, RED and CBQ, routing algorithms such as Dijkstra, and more. NS2 also implements the multicast and some of the MAC layer protocols for LAN simulations [40].

NS2 is based on two languages (Figure 7); an object oriented simulator, written in C++, and an OTcl interpreter, used to execute users command scripts.

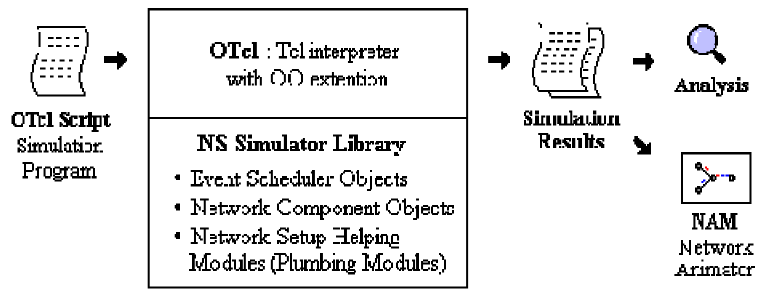

Figure 7: Simplified User's View of NS [16].

NS2 has a rich library of network and protocol objects [41]. The reason of implementing the C++ part of the simulator is to achieve more efficiency and faster execution time. Users use the OTcl script (that act as the user interface) to define the network topology, specify protocols and services which are planned to be simulated.

NS2 is a discrete event simulator, where the simulation time depends on the timing of the events which are maintained by a scheduler. The simulator has a list of events which are executed one by one in order. There is no need to worry about locking in the program, since it is a single thread simulator.

In NS2 you can easily make a new object either by writing a new object directly or by making a compound object from the object library in the C++ parts and connecting it to the OTcl script using the OTcl linkages. We used this ability to add some modules to the simulator before starting our simulations. More details about these additions are given in the rest of this chapter.

3 Network architecture

Since our services target the end users, we simulate the access network with four level of hierarchy. Figure 8 shows the physical topology of our network. A ring of routers is connected to L2/L3 switches, except one that is connected to the servers. At the next level, access switches are connected to L2/L3 switches, and end users are connected to the access switches at the end of the tree leaf. Network level multicast is active in all nodes.

Each user has a STB with the enough storage capacity to store several movies and act like seeds if needed. The application installed on the STBs supports P2P data transfer and internet protocols such as TCP to download from VoD server or peers, and multicast protocol to receive IPTV streams.

In the VoD series of simulations we have planned to show the effectiveness of the hybrid model. So we have several VoD servers which contain all the movies in the network and each one is connected to a router to handle requests from all nodes connected to this router. All the VoD servers have the same content, and the reason of having several servers is to support more nodes in our simulation: the number of nodes is limited by the VoD server upload bandwidth. Each end user initiates a TCP connection to the VoD server, so it can later download chunks through that connection.

There is also a tracker for the P2P application. The tracker contains the list of movies that each node already has, and it updates its database by sending content requests periodically. The IP addresses of the nodes are ordered in such a way that the tracker can distinguish the physical location of each node to keep the traffic local as much as possible. Each end user sends a peering request to the tracker via a TCP connection between its STB and the tracker, before starting to watch the movie. The tracker replies with a list of IP addresses of five peers that have enough chunks of the requested movie, ordered by their physical distance from the requesting node.

In the IPTV related simulations, we examine the catch-up IPTV service: we add an IPTV server to the router to broadcast multicast streams. In this part, the VoD server receives all the TV shows from the IPTV server, so it can provide the content afterwards as a catch-up service. The tracker has the information about all TV channels and their programs. So when a node sends a request to the tracker, the tracker may establish an UDP (User Datagram Protocol) connection between the STB and IPTV server, if the requested movie is being broadcasted as an IPTV channel.

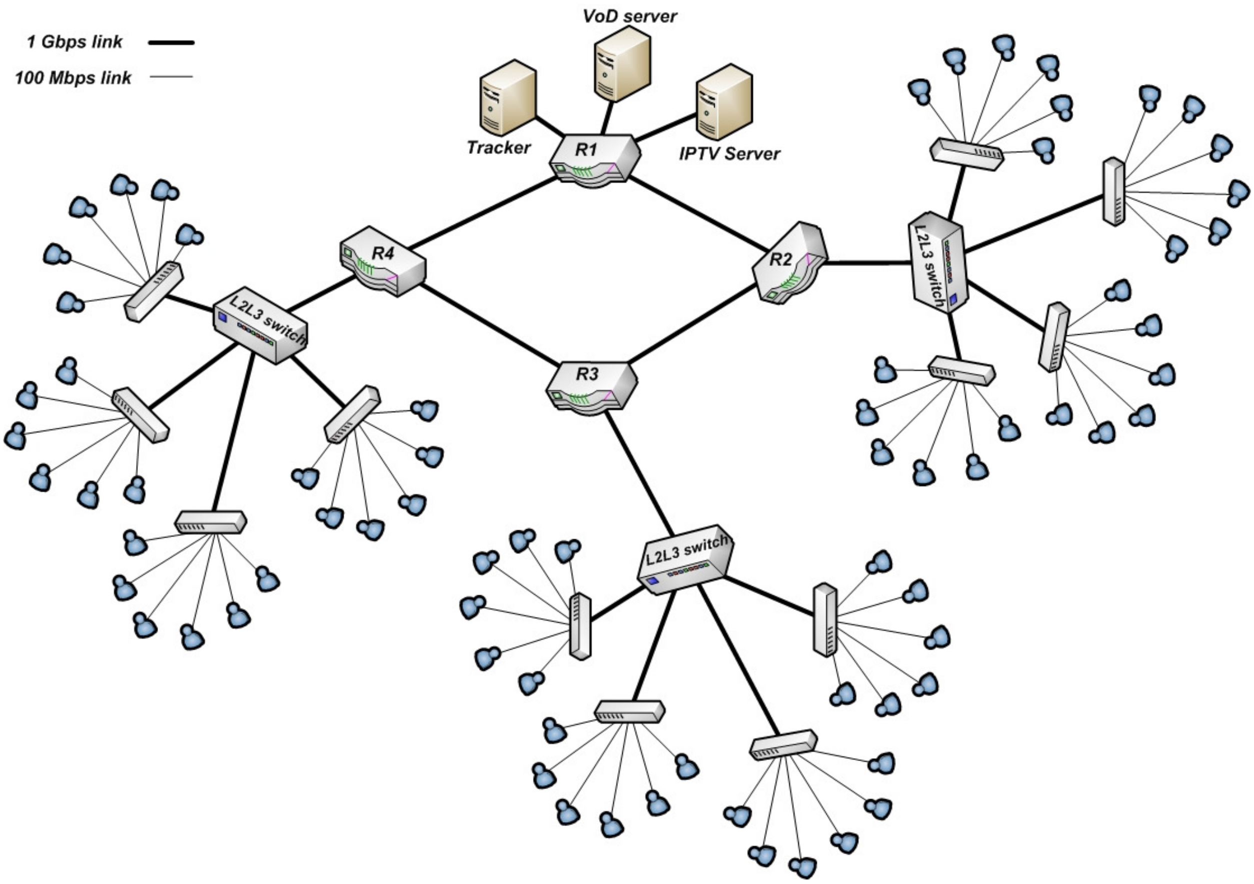

Figure 8: Network topology.

As it is clear from the Figure 8, all link bandwidths are 1 Gbps except the links between end users and their access switches which are limited to 100 Mbps.

4 Functional Implementation

a) Definition of concepts

- **Swarm**: a swarm is a group of peers trading chunks of the same file.
- **Area**: Area is formed from a number of end users connected to access switches belongs to same router and has the same subset of the IP addresses. For example in our case we have 3 area one for each router.
- **Profile**: We have different types of profiles in our scenarios. Each profile represents a special group of end users with the same behavior regarding the favorite movie or TV channel.
- **Seed**: Some nodes are not watching any movie during the simulation and they are just used as a source of download for other nodes. We assume that they watched the movie before and stored it on their STB to share with other peers later.

- **Movie**: A movie is a file containing chunks of data, which are either downloaded via a TCP connection with the VoD server or peers or received from an IPTV stream**.**
- **Sliding window**: The chunks near the playback position that have higher priority. The sliding window contains N chunks starting from the current play-back position. In our implementation, the sliding window size is equal to 25 chunks, and if the next empty chunk is inside the sliding window, the application downloads it from the VoD server (Figure 9).

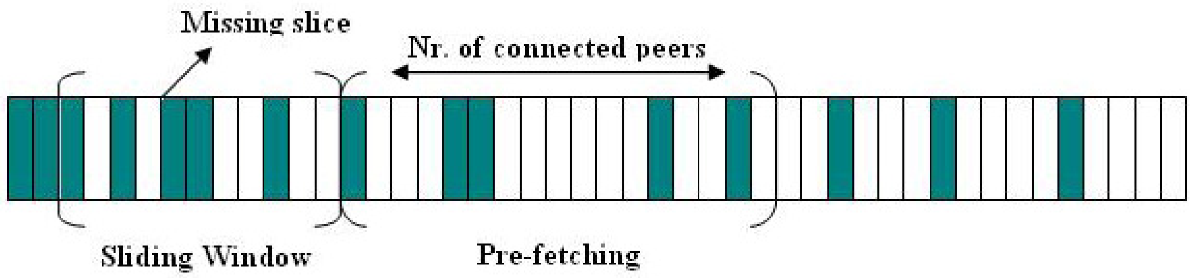

Figure 9: User STB modules for download [42].

- **IPTV nodes**: In the scenarios that have IPTV channels, some nodes arrived on time and started watching TV channel from the beginning. These nodes do not download any chunk from other sources, and receive the whole movie from a multicast stream. But they may be used as seeds for catch-up nodes.
- **Catch-up nodes**: If a user requests a movie that is broadcasted from an IPTV channel after the start time of the show, it is called a catch-up node. Depending on the scenario and services, the way that a catch-up node retrieves the show is different. More information in this regard can be found in the simulation part.

b) Implemented Hybrid Solution

This part of the report explains the model used during the simulations and my contribution to it. Two separate but similar routes were followed: one without IPTV streaming and another one with IPTV service. In both case we have a VoD server, a tracker and a P2P application on end users STBs. The following section describes how each part is implemented in our case, but before that the general structure of the implementation is presented.

As has already been said, the NS2 simulator uses a combination of the C++ and Tcl languages. The general topology of the network, the simulated physical parameters, number of the nodes and their applications are defined in the Tcl script written by me. A smaller version of the script had already been written before me, and I used it as the basis of my script, so I did not wrote everything from the scratch. The most important missing part in the version I received was the Multicast capability, which I added to the NS2. I also adapted the old scripts with this new feature for my simulations.

The changes to Tcl script lead to changes in the C++ code, since there is a direct linkage between these two modules of the simulator. Another sizable part of the work to prepare the simulations was to create profiles to load to the end-user nodes. The profiles contain the information about the scenario which is going to run, the profile name, the movies that each node has in its hard drive and the time and the title of movie that the end user is scheduled to watch. More information about the profiles is given in a later part of the report.

VoD server

The VoD server is our video server. It stores all the movies that can be watched by the end users. It also receives all IPTV streams send by the IPTV server. It is used as a source by the end users who can download their desired chunks via a TCP connection they have with this server. In our simulations, each node initiates one unidirectional TCP connection to the VoD server at the beginning of the simulation. There may either be one main VoD server connected to the main router, or several local VoD servers with exactly the same content, each connected to the one of the routers of each area.

Tracker

The tracker is the node responsible for peering between the nodes in the network. The tracker has a list of all the peers with their IP addresses. Also it has other necessary information about the end users such as their Table of content (ToC), the number of downloads and uploads of each node, the available IPTV channels and their time tables. Also during the simulation, the tracker updates the peers ToC periodically by sending ToC requests to each peer. Tracker uses this information to send back a list of maximum 5 IP addresses of peers to the client who has sent a peering request. Each peer initiates two unidirectional TCP connections to the tracker at the beginning of the simulations.

When it's time for a peer to start watching a movie, it sends a request to the tracker via the pre-established TCP connection. The tracker processes the request and acts upon it. If the requested movie is a TV shows, the tracker first checks if the movie is broadcasting at the time of request. If this is the case, it may initiate an UDP connection for connecting the peer to the appropriate IPTV stream. If the P2P capability is active in the network, the tracker searches for peers who have enough of the requested content: the tracker checks both the locality of the peers and their content, and it tries to chose the nearest peers regarding to physical distance (from same area) who still has enough upstream capacity.

In our implementation each node can seed up to 8 other peers and open a maximum of 5 connections for downloading chunks. To keep the traffic as local as possible, the tracker first tries to find peers connected to the same access switch as the requester. If it cannot find enough nodes in this stage, it searches one level higher in the nodes hierarchy. It continues this process to find the maximum number of seed for the peer.

Peer and STB download strategy

Each end user that is scheduled to watch a movie or is used as a seed is called a peer. All the peers know the IP address of the tracker and the VoD server from the beginning and are connected to them. Each end user receives a configuration file at the set up time of the simulation which defines its content on hard disk, the movie it is planned to watch and the start and stop times of watching. Figure 10 summarizes the download strategy of peers where P2P service is used in access network.

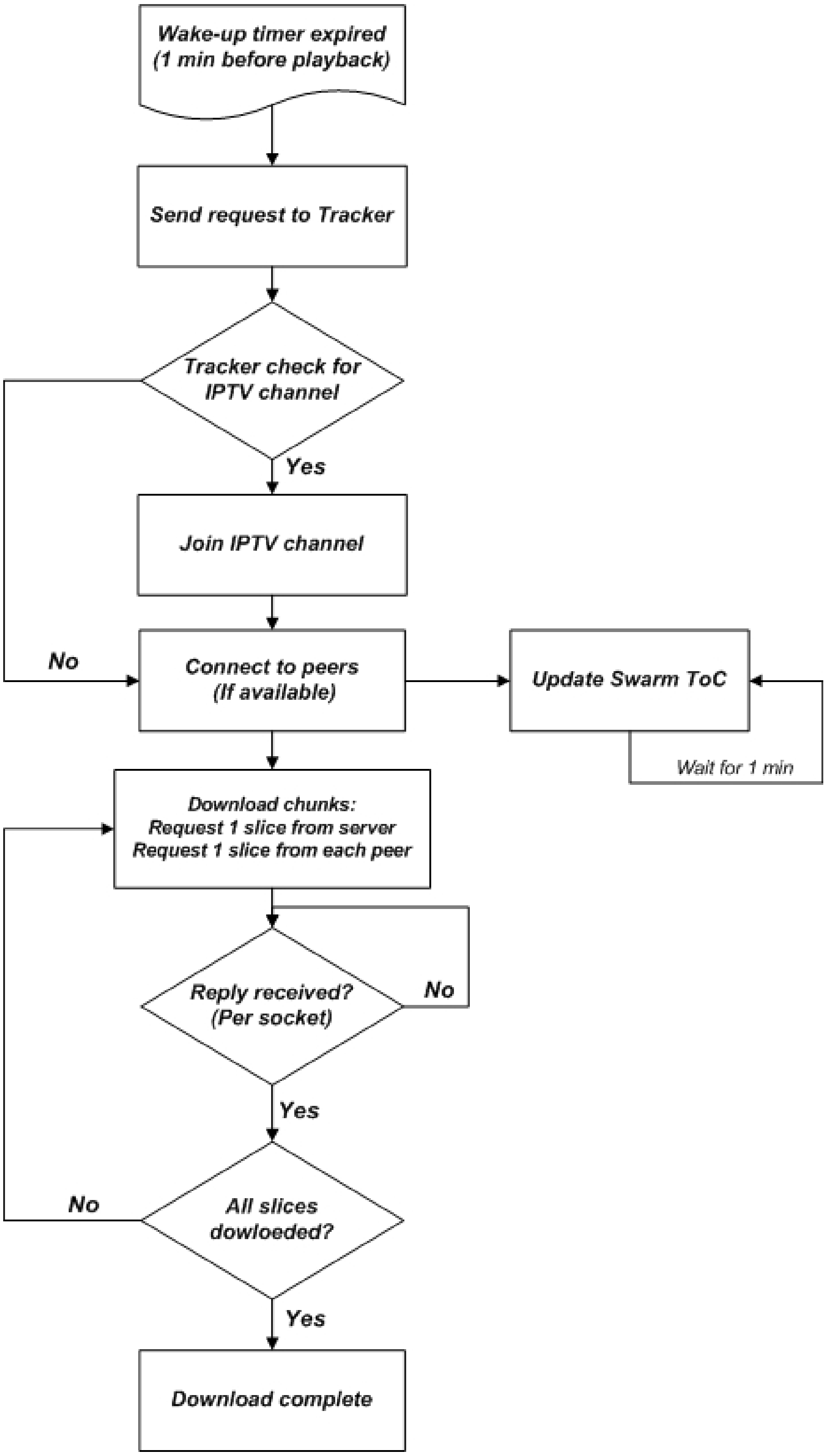

Figure 10: Downl.oad strategy of user STB.

One minute before the start time, the STB application sends a peering request to the tracker and waits for the reply. Depending on the reply from the tracker and the scenario type defined in configuration file, the following states may happen.

1. **IPTV:** The node is scheduled to start watching an IPTV show during the first minute of it. In this case, the node downloads the whole movie from the multicast stream. Since the tracker has its ToCs, the node can be used as a seed for other peers.
2. **Catch-up IPTV:** The node requested to watch an IPTV show, after the show has started. Depending on the simulation it may download the whole file from the VoD server or join the IPTV to get the latter part of the show while it downloads the missing chunk from the VoD server or by P2P.
3. **Pure VoD:** These nodes are planned to watch a movie that already exists in the VoD server. The content is retrieved completely from the VoD server.
4. **Hybrid scenario:** Regardless of presence of an IPTV stream, these nodes download content from the VoD server and from peers. As was mentioned before, there is a sliding window with a length of 25 chunks. Chunks inside the sliding window are retrieved directly from the VoD server, while other chunks are downloaded from peers. Each peer has two TCP connections with other nodes, one as client (to receive chunk or send request) and one as the server (to reply request or upload chunks). While download peer will send request via client socket to its peers and VoD server simultaneously and wait for reply from the same socket. After receiving the reply that port is free to request another piece of data. Also end users update their Table of Content every minute in order to know the latest contents of their peers.

IPTV server

This server is configured as the source of multicast streams and sends several TV channel to the receivers. A UDP connection is created between the VoD server and the IPTV server for each TV channel, to let the VoD server receive the entire stream. UDP streams with a bandwidth of 4 or 5 Mbps are sent, depending on the scenario. The server starts sending multicast streams from the beginning of the simulation, independently from the other part of network. Peers who want to watch a movie from this stream join the IPTV stream via the tracker. To have the possibility of using the data received from the IPTV stream as seeds for catch-up nodes, the movie content is divided into chunks of the same size as in the P2P applications. Each chunk contains a constant number of UDP packets. There was no multicast sender actually sending data implemented in the NS2, so I added a new class to the C++ code of the simulator, as well as the corresponding linkages codes to Tcl.

5 Logging and analysis tools

Running the simulation does not provide any quantitative result to us. One of my contributions to this project was rewriting logging at the application level and designing and implementing logging at the network level. At the application level, a logging class was added to NS2 in C++ to store information in log files. The information we gain through these log files is listed here:

1. The list of IP addresses and the content percentage of end users offered by tracker to requester node to use as seeds in P2P application.

2. Statistics of the number of chunks downloaded by each end user from different sources. For example in the scenario where we have IPTV, VoD and peers, we log the number of downloaded chunks from each source separately.
3. The playback quality, and the local cache status during playback
4. Errors related to the performance of the scenarios while running simulations, such as missing a chunk due to network congestions.

NS2 has a monitoring ability over the links in the packet and flow level. We add some OTcl script to use this possibility to log some useful information. We monitor all the access network links except the links between end users and access switches, to have the bandwidth usage and statistics on the number of dropped packet in our simulations. Two log file are produced for each link for two directions. The logging period can be defined by the user. Since we are planning to measure the bandwidth, we set the logging time period to one second. We have the number of dropped and total packet per second, as well as the number of dropped and total bytes per second, both per flow and in total.

Table 2 shows an example of one line of these monitoring files. As can be seen there are 13 columns on it. The first one shows the sampling time. The flow id defines the different flows, since we have several services here, or different source and destinations, we have several flows. The packet type is shown in the third field, which in our case has two values for TCP or UDP packet types. The NS2 internal address of sender and receiver, which is different from their IP addresses. The rest of the field present the accumulative number of bytes or packets arrived to the node from the beginning of the simulation, in total or for this special flow.

Table 2: Sample output line of monitoring file.

Time (sec)	Flow ID	Packet type	Sender	Receiver	Packet / flow	Bytes / flow	Total packet	Total bytes	Total dropped Packet	Total dropped bytes	packet dropped / flow	bytes dropped / flow
33	1	0	8	338	294	124088	338	1361184	0	0	0	0

After gathering the results of the simulations, some analysis tools are needed to give a more understandable view to us for the comparison. Using the graphs for demonstrating the results is one of the most common ways for analysis. So, we wrote a C++ program to extract the information we needed to a separate file which is used as the input data for drawing graphs. The graphs produced from the raw data were very noisy (due to the fine grain sampling of 1second), which made it hard to visualize the numbers. That is why we used the Gaussian filter to smooth our graphs.

6 Profiles

As was discussed earlier, using profiles allowed us to dictate each node behavior just as we need for our simulations, so we implement profiles fitting our goals. The factors we are considering for defining the profiles are as follows:

Decision on the profiles types: After studying some research documents that shows the statistical data on the Sweden population behavior regarding the internet and video services, we decided to have three different profile types: Student, Family, and Retiree. Having three profiles let us include varieties of behaviors and interests caused by aged difference in the society, as well as keep the simulation simple by dividing peoples to three groups.

The number of movie available for download: For the Video on demand series of the simulation, we need to define the number of available movies per profile, since it influences the efficiency of the P2P application. The long tail theory helped us to have a better estimation of the number of movies suitable for each scenario, though we did not follow the formulas. To see the effect of the niches and hits movies on the efficiency, we decided to run two scenarios, one with more variety of options, but less seeds. This scenario is to show the niches movies situation when there is less demand for lots of movies, which causes a lack of data in the nodes, since fewer users have watched them before. In the other scenario for more popular movie we have a lower number of movies with more demand on them.

The number of request per movie: We use the number of customers as an input factor for defining the number of movies. For example since we have more student profiles than retiree in our network, the number of student movies are more than the number of retiree movies.

The amount of available seeds for each movie: The amount of available data in the end users is an important factor in the efficiency of P2P services. So we have to come up with the percentage of nodes that have already watched the movies and can be used as the seeds for other nodes, considering the fact that nodes that are watching movies also can share it simultaneously so will be counted as the seeds.

The number of TV channels and their distributions between profiles: In the IPTV simulations, there are TV shows available for download instead of VoD movies. To decide on the number of channels and their content, we went through Sweden's TV channels SVT1, SVT2, TV3, TV4 . . . and investigate the time and variety of programs for each age group in the specific time slots. At the end we decided to have 6 TV channels, with different watching probabilities for each profile. For example the probability of watching the channel 1 by a student profile is 60 percent while this number decreases to 20 percent for the retiree profile. The numbers inverted for another channel that has a show that is more compatible of the retiree's interests.

Percentage of catch-up nodes: There are two kinds of end users in the case of IPTV scenarios: IPTV nodes and the catch-up nodes. The portion of each group compared to the total number of node is another factor that should be defined in the preparation phase. For each profile, different percentages of the total nodes were defined as the catch-up nodes and started watching TV shows at least 15 minutes later than the start of the show.

Number of areas and profile distribution per area: To simulate the access network, we need to decide on the population distribution. In our simulations we divided the access network to three different areas; a family area that mostly contains the family profiles, with some student and retiree nodes, a student area occupied by students as well as 20 percent of family node, and finally a downtown area which is a combination of the three profiles.

IV Simulation preparation and result Analysis

1 Overview

As should be clear now, we are investigating two different video services in our project: Video on demand services and IPTV services. In this chapter, simulations are presented in more details. Two series of simulations were run with three scenarios in each. The first scenario of each series is used as the reference data for comparisons and analysis of the results which can be found in the next section.

Although the IPTV scenarios includes some VoD results, the focus is on catch-up services, and the behavior of a VoD service is not easily decoupled from the effects of other services. This is why VoD gets its own separate study. The goal of the VoD series of simulations is to investigate the factors which can influence a P2P service. The IPTV series show the effect of a P2P application on the load of VoD server and network links, but we need some numbers to show under which condition it has more effect on the network.

The later part of this chapter presents the analysis of the results. We ran several simulations to support our proposals for video services over the internet and compare advantage or disadvantage of different strategies. We continue this chapter by introducing the result of IPTV related simulations. There are three simulations in this part to compare 3 different catch-up methods for IPTV services and find the best solution. Then the result of the 3 other scenarios are presented to see the effect of mixing P2P distribution with a pure VoD services.

2 IPTV services simulations

Generally our physical topology consists of four levels of aggregation. In this series of simulations, there is one VOD server with one gigabit of bandwidth. Using this bandwidth we calculate the number of VOD node that can be served and generally the total number of nodes in our simulation. We have four routers connected to each other as the ring. One of these routers is connected to our servers: the IPTV server, the VOD server and the tracker. The other routers are connected to layer2/layer3 switches.

Each router facing end users is connected to the three L2/L3 switches. At the next level, there are four access switches per L2/L3 switch. Finally, 8 end users are connected to each access switch. Table 3 represent the node distribution used in these series of the simulations.

Table 3: Physical distribution of nodes in access network.

Node type	Number per level	Total number
Router	3	
L2/L3 switches	3 per router	9
Access switches	4 per L2/L3 switches	36
End user	8 per access switches	288

We have three different users profile; Family profile, Student profile, Retiree profile. More details of how we defined the profiles and their characteristics can be found later in this chapter.

To be more realistic, we decided to have three different areas in our simulations; a Family area, a Student area and a Downtown area, in the way that all the branches under each router correspond to one area. The profile distribution of each area, are listed in the Table 4.

Table 4: Profile distribution per area.

Area type	Family nodes	Student nodes	Retiree nodes
Family	60,00%	30,00%	10,00%
Student	20,00%	80,00%	0,00%
Downtown	50,00%	30,00%	20,00%

There is another type of node group which divides nodes to two groups: IPTV nodes and catch-up nodes. The percentage of the catch-up nodes is different depending on the user profile. For example since students normally have a less organized life, there are more catch-up nodes in this groups who will start their favorite shows later. In our case, 80% of student's nodes will use catch-up services. This number decreases to 20% for retiree profiles, since they have plenty of free time and can adjust to show times. We assume that families' behavior is between these two groups, so the proportion of catch-up family nodes is the half of total nodes.

Moreover we decided to have six IPTV channels, with one movie per channel, to have enough diversity in end users options in our simulations.

a) Simulation 1-1: IPTV service and catch-up with VoD

The differences between the scenarios are in the download strategy of end users and sources of download. In this scenario, which is used as the reference for the two following ones, nodes send the join request to the Tracker. The tracker first checks if there is any IPTV channel for the requested movie, then compares the request time to the show time. If the request arrived within the first minute of the TV show, the tracker connects the end user to the IPTV stream. Otherwise the movie is considered as a catch-up movie, and the node has to download the whole show from the VOD server, via their TCP connection.

This method is similar to the available catch-up services in the today's internet. Tracker is responsible for finding a suitable TV channel and connecting the node to the IPTV stream.

b) Simulation 1-2: IPTV service and catch-up with VoD and IPTV

In this scenario we are using the patching method. It means that if the catch-up nodes start watching a movie that is being broadcast, the tracker connects them to the IPTV stream. So they will store chunks received via multicast stream in their local cache and download the first parts directly from the VoD server. We think that this method will alleviate the VoD server load, without increasing the network load.

c) Simulation 1-3: IPTV service and catch-up with VoD, IPTV and P2P

Here we have the same conditions as in the previous scenario. The only difference is that the missed part of the IPTV stream is downloaded via the VoD server and the peers. As it is described before, the chunks inside the sliding window are downloaded from the VoD server and the rest are retrieved from other peers that the tracker chose for the node. This is our favorite and most complete solution in these series of simulations.

3 Video on Demand services simulations

The physical topology of the network is the same as in the previous simulations with minor changes in the number of nodes and the addition of local VoD servers. The number of nodes in each level is summarized in the Table 5. In this Table we do not write the router connected to the servers, since there is no end user connected to it. Also, there is one area type per router with the same percentage of profiles types and catch-up nodes as before. As was described in the previous section, the VoD server bandwidth limits the number of end users who can be served in the network. As a result, we decided to have one local VoD server per area to increase the number of nodes in our experiments. In this way we can study the benefits of P2P applications in the scalability of services.

Table 5: Physical distribution of nodes in access network.

Node type	Number per level	Total number
Router	3	
L2/L3 switches	4 per router	12
Access switches	4 per L2/L3 switches	48
End user	24 per access switches	1152

In this part, we still have the same types of profiles but with different ToC for each scenario.

a) Simulation 2-1: Video on Demand with pure client-server model

In this scenario which is used as the reference for two following scenarios, nodes are considered as VoD users. Each node downloads its content from the VoD server connected to its local router.

b) Simulation 2-2: Video on Demand with a hybrid model for hit movies

The download strategy and peering model are similar to the third simulation of the IPTV series, except that patching is not used here, since there is no multicast stream. One of the affecting factors in the efficiency of the P2P applications is the amount of data available in the peers. To study this influence, we run two similar simulations with different numbers of movies per profile types. In this scenario we have fewer movies with higher demands. This simulates the efficiency of P2P for popular and newly-released movies. There is a huge number of requests for these movies during the first several weeks of their release. Consequently there are enough seeds for them in the network, since lots of nodes watched them recently and stored them on their STB's hard disk.

c) Simulation 2-3: Video on Demand with hybrid model for niche movies

The efficiency of video on demand can be changed by some factors such as the availability of the requested contents. So in this last simulation we have more movies compare to previous scenario with fewer amounts of data pre-stored on the peers. In this scenario some movies are not popular, so no one has them in the hard disk. For this kind of movies, P2P does not help, and they have to be downloaded from the VoD server.

4 Efficiency assessment of IPTV and catch-up services by comparing the VoD server load

a) Simulation 1-1: IPTV service and catch-up with VoD

In this scenario there are two kinds of peers: the ones who start watching their favorite TV show on time and receive all the show via IPTV (multicast stream), and the ones who are late for the show and download the whole show from the VoD server. We configure the nodes in the way that catch-up nodes start watching the show at least 14 minutes after the start time. The graph below shows the bandwidth usage of the VoD server for this scenario. The red line shows the data arrival rate and the blue one represents packet losses on the link between the VoD server and its router.

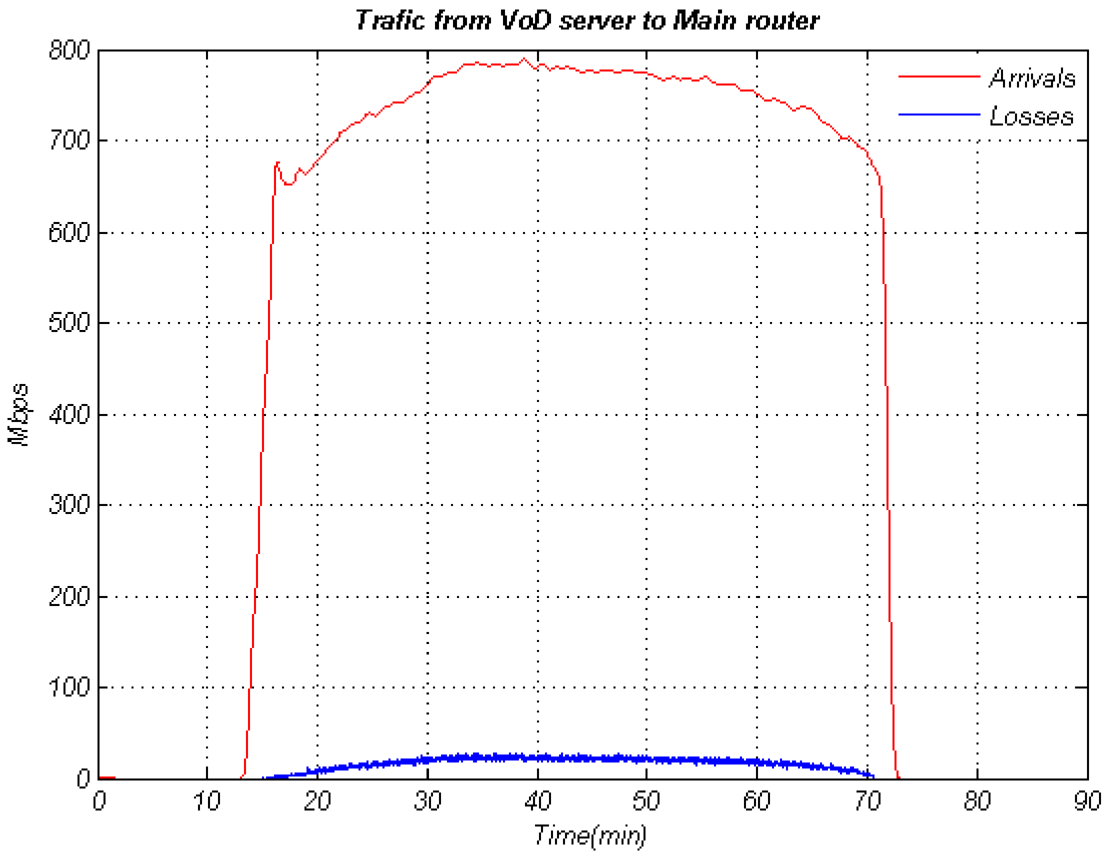

Figure 11: Traffic from the VoD server to the Main router, IPTV with VoD catch-up.

There is a small amount of packet loss on the link, and it follows the same trend as the link load. Since we use TCP connection for downloading from VoD this loss does not have any effect on the results and it is due to the characteristics of TCP protocol. The red curve shows the data amount in Mbps during the simulation. Each movie lasts between 50 to 60 minutes in our simulation so, nodes start download their show from the VoD server around 14 minutes after the beginning of the simulation. This causes a sharp increase in the bandwidth usage of the link. The number stays constant at around 800 Mbps for around 50 minutes of the simulation, and decreases to zero after all nodes have downloaded the show completely. To serve more customers, the bandwidth of the VoD server should be increased, which is not a cheap and efficient solution.

b) Simulation 1-2: IPTV service and catch-up with VoD and IPTV

The only difference between this scenario and the previous one is on the downloading strategy of the catch-up nodes. Here all nodes register to the IPTV stream when they start watching the TV show. But if the show has already started, the end user retrieves the missed parts of the show from the VoD server, while storing the later part of the show from the multicast stream. This solution decreases the work load of the VoD link, since nodes do not need to download the entire file from the server. Figure 12 shows the traffic on the link between the VoD server and its connected router, during this simulation.

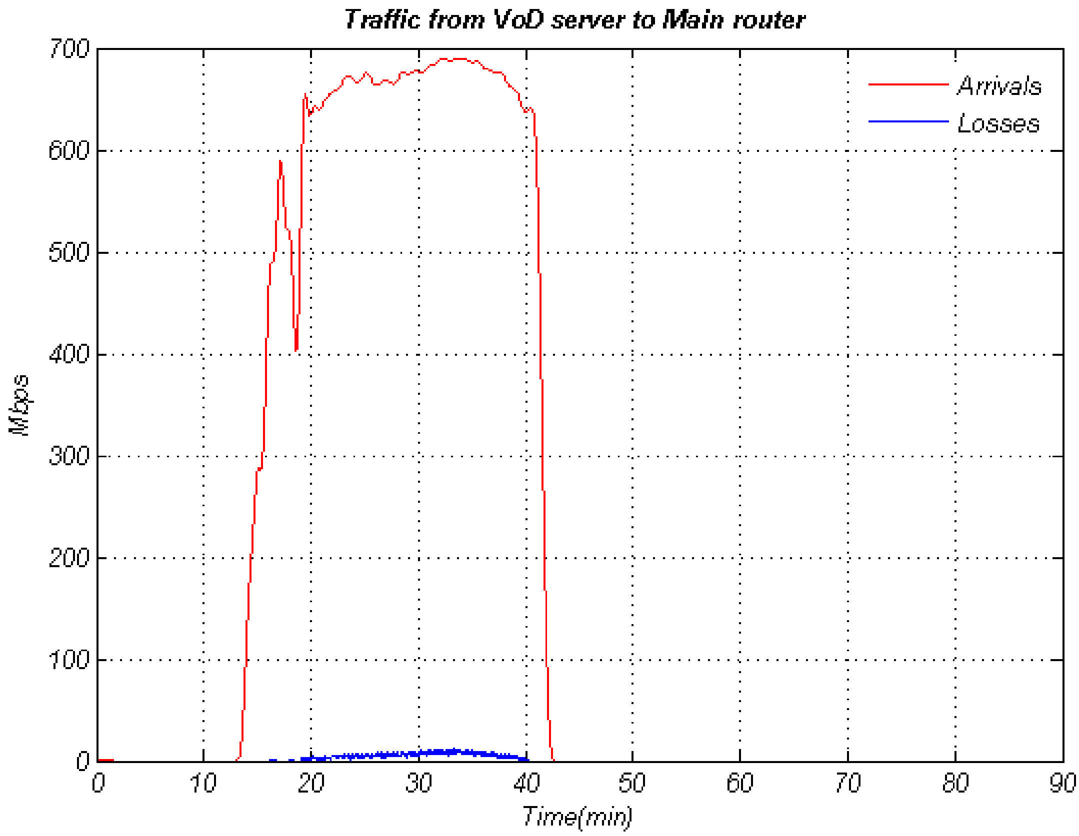

Figure 12: Traffic from the VoD server to the Main router, IPTV with VoD and IPTV catch-up.

For example if a node starts watching a movie 15 minutes late, it needs to download these 15 minutes from the server and the rest of it is retrieved from IPTV service, which means ¾ decrease on the link load for this special node. Although there is not considerable decrease on the peak load, the idle time of the link is much more than the previous scenario. In the first scenario the link becomes idle after 70 minutes of the simulation start, while this number decrease to about 40 minute in this scenario.

c) Simulation 1-3: IPTV service and Catch-up with VoD, IPTV and P2P

In this scenario we are evaluating our hybrid model to see how efficient it is compared to the other two. As you can see from Figure 13, the graph has a totally different trend. There is a considerable decrease in the peak load of the VoD link to 160 Mbps that is 5 times less than the maximum bandwidth capacity of the link. It means that using P2P along with the VoD server for catch-up service, allows us to serve more customers without increasing the VoD server bandwidth, since each node uses the server bandwidth for a short period at the beginning of download. The discrete lines in the graph are due to the fact that nodes download the chunks inside sliding window from VoD server in the first seconds after they tend to watch TV show, and retrieve the rest of missed part from their peers. Similarly to the second scenario, nodes store the IPTV stream from the point they start watching the show.

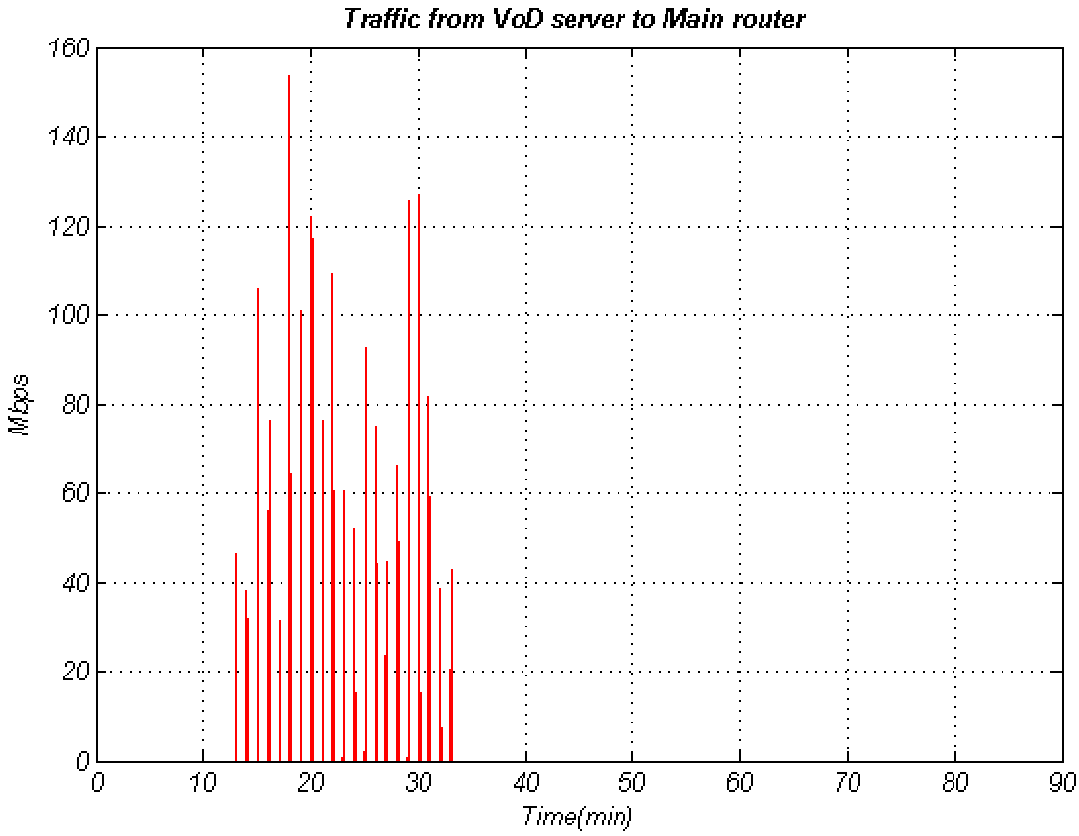

Figure 13: Traffic from the VoD server to the Main router, IPTV with VoD, IPTV and P2P catch-up.

In this part we just focused on the influences of the different catch-up methods on network resources. The next part shows the effect of P2P distribution on network efficiency in more detail.

5 Efficiency assessment of Video on Demand services via comparing the VoD server load

a) Simulation 2-1: Video on Demand with pure client-server model

In the first series of simulation, we have planned to see the effect of different catch-up TV methods on the access network load. In this part we are trying to purely assess the proposed hybrid method for VoD, without mixing it with other parameters such as IPTV. To increase the number of end users to have a more realistic simulation, we put one local VoD server for each area (see Figure 14).

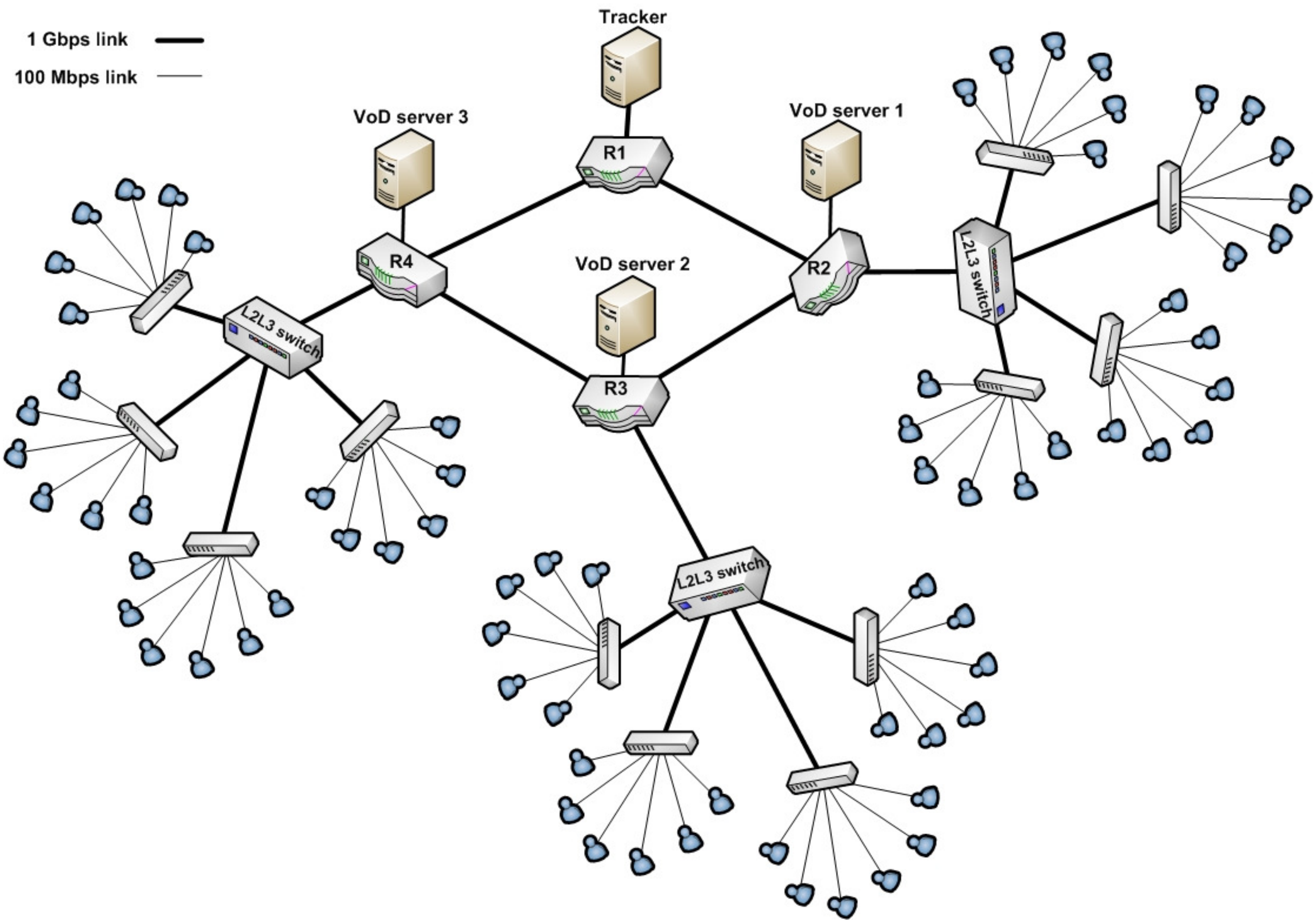

Figure 14: Network topology with one VoD server per Area.

In this scenario, end users start watching TV at different time from the beginning of the simulation. They download the whole file from the VoD server while watching it at the same time.

It can be seen from the Figure 15 that there is a sharp rise in the bandwidth usage of VoD server from the first minute of simulation and continues for around 70 minutes, at which point all the nodes have completed their download. The bandwidth usage is near to the maximum capacity of the link (1 Gbps), which shows that the server cannot serve more nodes than is used in this scenario.

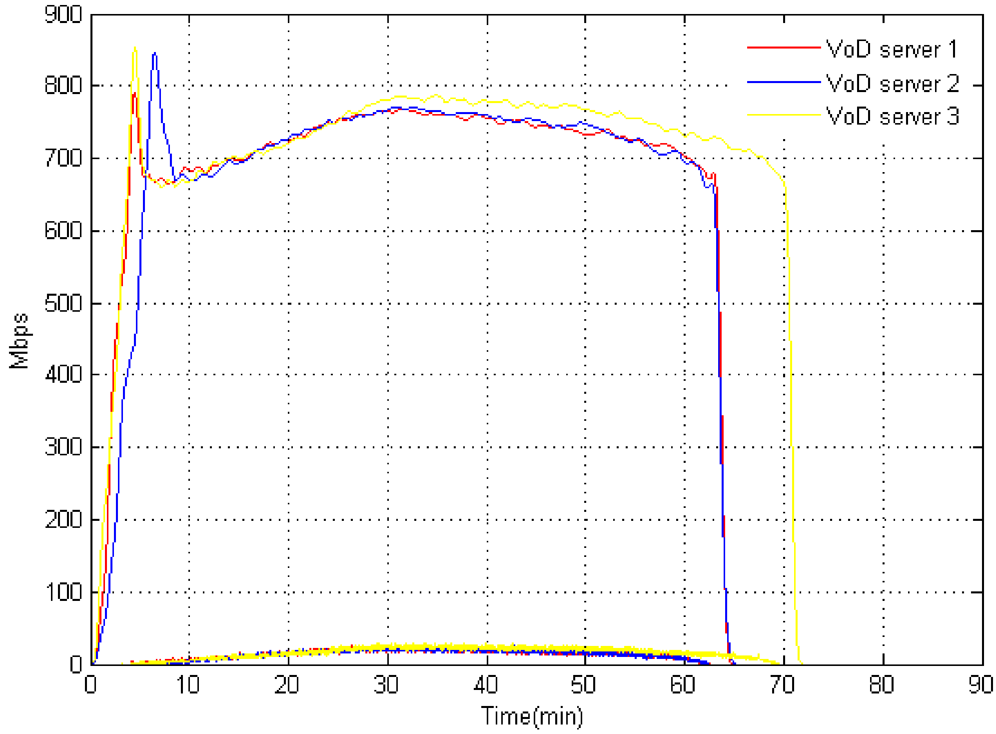

Figure 15: Traffic from the VoD servers to their routers, pure VoD scenario.

b) Simulation 2-2: Video on Demand with hybrid model for hit movies

The amount of available sources for download is one of the important factors in efficiency of P2P applications. So we set up two scenarios with different numbers of movies and available seeds for each movie. In this scenario we have few movies per profile with many seeds for each movie. We have a maximum of 5 movies per profile type, and each end user has at least one movie to share with its neighbors, meaning there are enough seeds for download in the network.

The traffic on the link between the local VoD servers to their connected routers are depicted on the Figure 16. First of all a considerable decrease in the server load is observed compared to the previous scenario due to using the P2P application, both in the download time and peak load. There is a peak bandwidth after several minutes from the beginning of the simulation. After that, the graph starts falling down, since nodes download the rest of file from their peers instead of the VoD server.

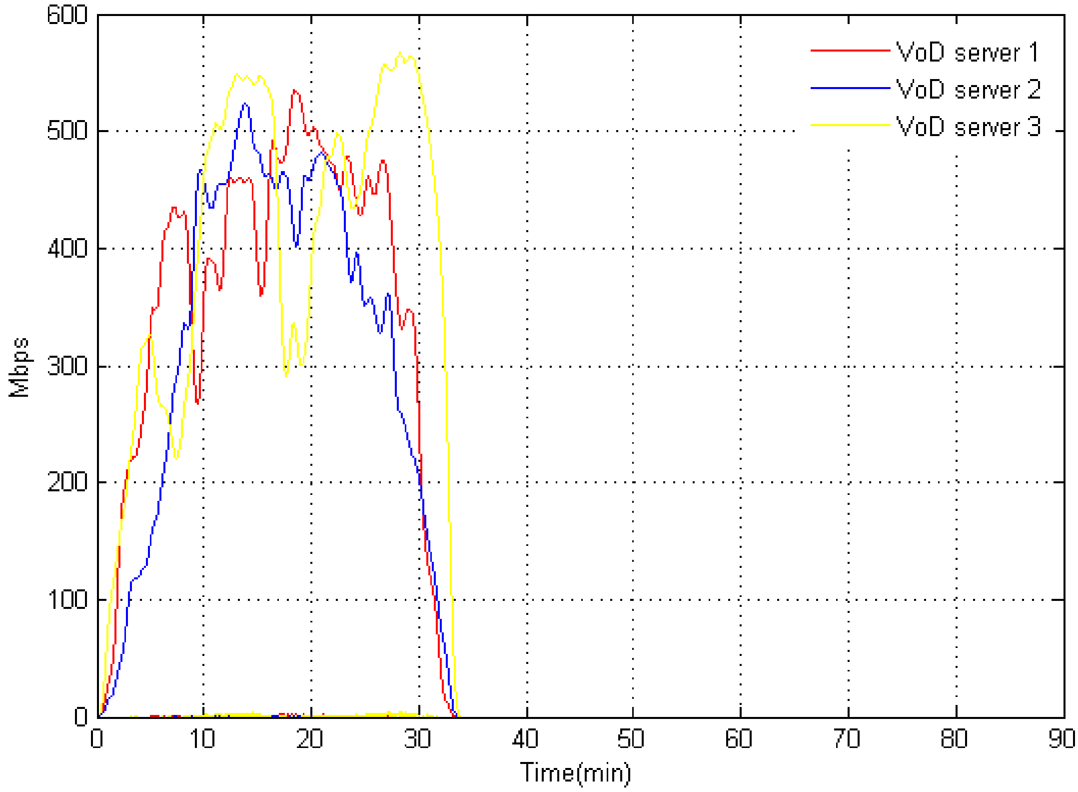

Figure 16: Traffic from the VoD servers to their routers, Hybrid model for hit movies.

c) Simulation 2-3: Video on Demand with hybrid model for niche movies

In this scenario we have more movies per profile type with fewer seeds per movie: this represents a non-ideal P2P network. In this simulation there are not enough seeds available for download, which decrease the efficiency of P2P and puts more pressure on the VoD server. There are 26 student movies, 19 family movies and 12 retiree movies in this scenario. Each node has one movie stored in its cache and nodes are able to share the movie they are downloading at the same time. The server peak load is higher than in the previous scenario, since there is not enough peers for the nodes who want to watch the show at the beginnings and more nodes have to download from their VoD server (Figure 17). Then there is an increase in the number of available seeds by time, since nodes that starts watching shows can be used as peers for new users. So there is a decrease on the server load from minutes 10 of the simulation. Generally the average load of server in this scenario is about than 100 Mbps more than the previous scenario, which shows the effect of available seeds in P2P networks. The more data available for download the more efficiency in application.

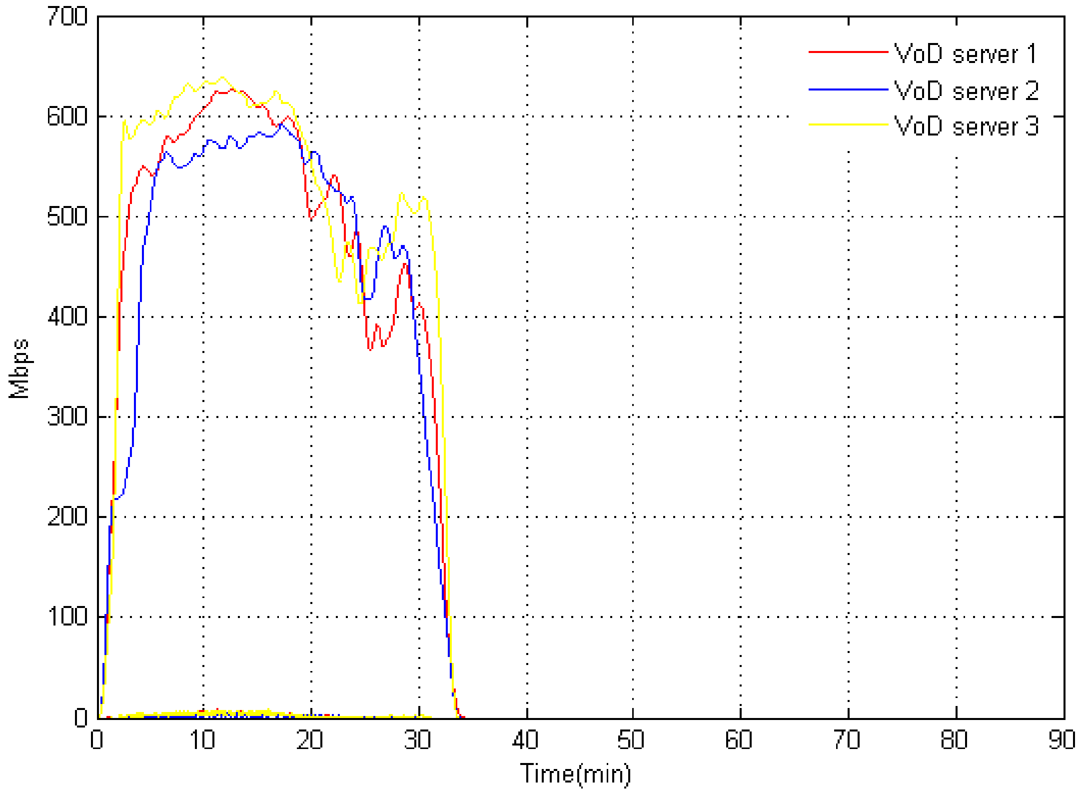

Figure 17: Traffic from the VoD servers to their router, Hybrid model for niche movies.

d) Network resources usage

In this part of the report, we compare the uplink and downlink bandwidth usage in 3 different scenarios. The Figure 18, Figure 19 and Figure 20 show the links between one of the routers and its connected L2L3 switches. In all graphs, red curves show the bandwidth usage in the uplink direction and the blue ones show the downlinks bandwidth. At first glance, the graphs show that traffic load of the network in the absence of P2P distribution is more predictable and remain nearly constant during the download period.

The maximum load in the downlink direction in pure VoD scenario is near to 160 Mbps, while uplink capacity is totally wasted with just 10 Mbps of traffic. The second figure which represents the hybrid model with niches movie, shows an increase in both the downlink and uplink directions, while the busy time of the links decrease to the half of first scenario. Since there are more sources for download (peers and VoD server) in this scenario, a node can download more data simultaneously, which leads to the more load on the link in smaller period of time. In the pure VoD model, peer will download at the playback speed (except slices of first sliding window) but in hybrid model end user uses its available bandwidth to download as much as possible from its peer.

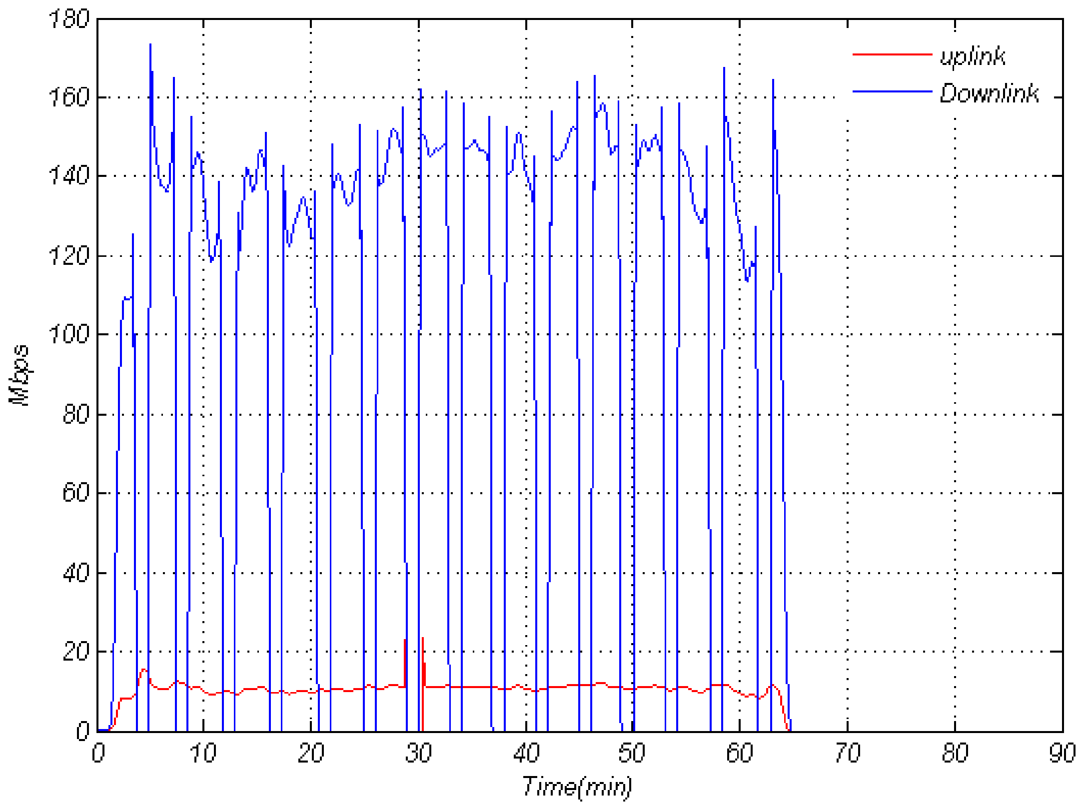

Figure 18: Bandwidth usage on the link between router1 and L2L3 switche1, Pure VoD model.

Also the P2P distribution uses the uplink capacity to feed the peers in other areas, which causes an increase in the uplink load to around 100 Mbps, 10 times more than pure VoD model. Although the downlink bandwidth usage of the VoD server decreases in second scenario compare to pure VoD model, the downlink usage of other links is increases duo to P2P application with limited seeds.

The graph 20 has the smaller network usage in downlink direction, since P2P sharing is more efficient in this scenario (hybrid model for hits movies) with more seeds. Data is exchanged locally between peers. The uplink usage in the link is around 70 Mbps, which is between the two previous scenarios. The uplink usage is more than pure VoD model because of using the P2P distribution and it is less than the niche movies model, due to the locality awareness algorithm: there are more seeds in the vicinity, so nodes mostly find their peers inside their own area, and the traffic rarely needs to travel between routers.

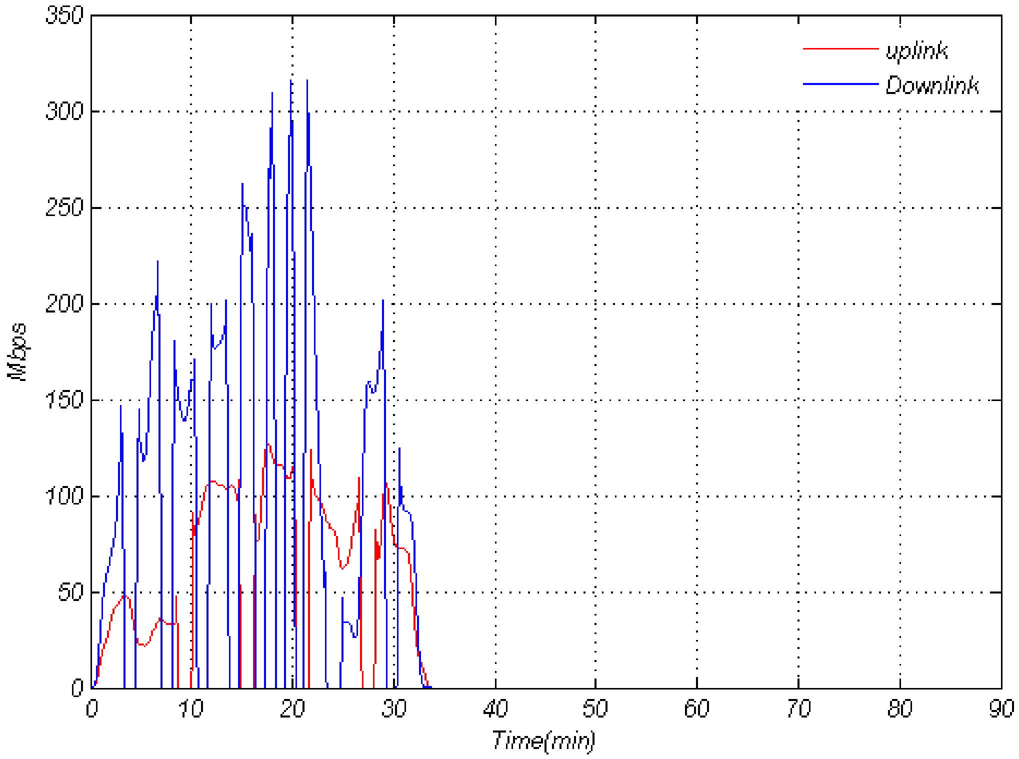

Figure 19: Bandwidth usage on the link between router1 and L2L3 switche1, Hybrid model, niche movies.

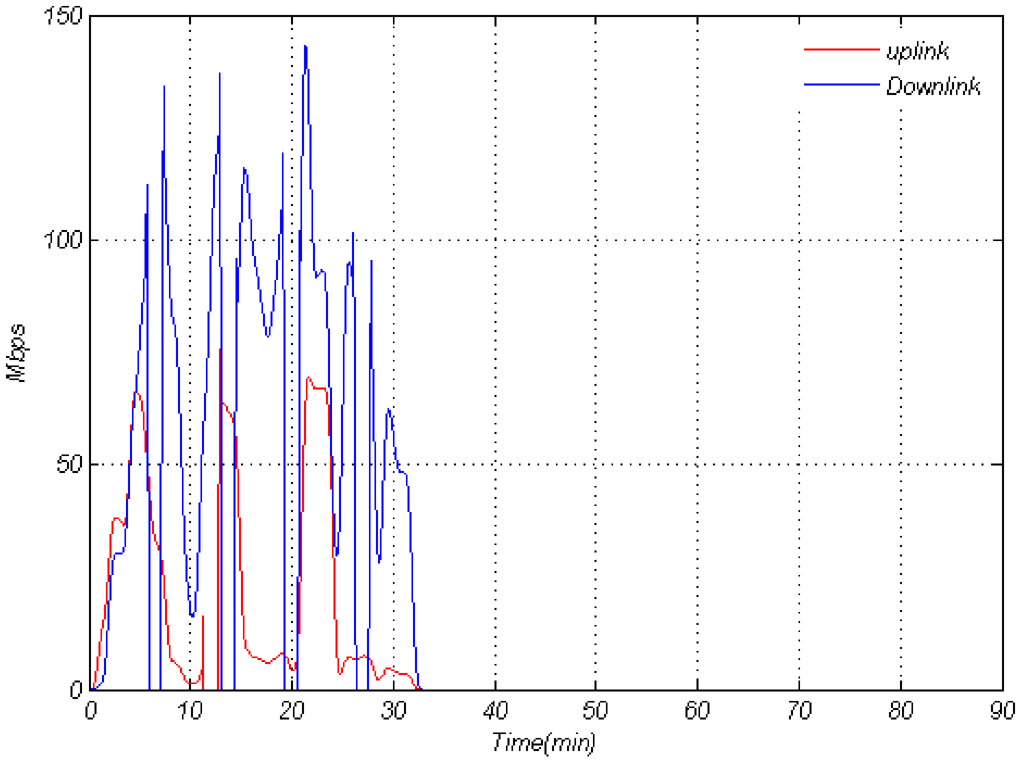

Figure 20: Bandwidth usage on the link between router1 and L2L3 switche1, Hybrid model, hit movies

The lower traffic load in both direction is belong to the hybrid model with hit movies, that shows this model is most efficient model for online video distribution services between our scenarios.

e) Peer to Peer locality awareness

The locality awareness is one of the important factors in the efficiency of P2P traffic. In our project we implemented a simple technology to keep traffic local as much as possible to increase the efficiency of the services. The access network is divided to four zones regarding the distance from requesting node. Peers are in region zero of a given end user when they are connected to the same access switches. If they are under same L2L3 switch but in a different access switch, they are in Region 1. Peers belonging to the same router but not the same L2L3 switch has a distance of 2 (region 2) with end user. The longest distance is with peers connected to a different router than the initiator node, with a distance of 3.

Looking at the end user distribution in the access network, we estimate the probability of peering with a node in another region, if all nodes are considered equal. The calculation shows that the peering probability increases with the distance, as can be seen in Table 6. It means that without having a method for locality awareness, traffic will scatter through the access network as much as possible.

Table 6: Peering statistics without locality awareness scheme.

Region	Region 0	Region 1	Region 2	Region 3	Total
Calculations	23/1151	(3*24)/1151	(3*4*24)/1152	(2*4*4*24)/1152	1
Percentage	2	6	25	67	100

Two other Table shows the peering statistics from the simulations regarding the locality. Table 7 shows a huge different between calculation and the simulation results due to using the locality awareness algorithm. There is no rising trend by increasing the distance. Most of the peers are belong to region 2, meaning nodes belonging to the same L2L3 switch. This saves a considerable amount of routers link capacity and network resources, but still 18 percent of peers are chosen from the region 3 and belong to another router.

Table 7: Peering statistics related to hybrid model with niche movies.

Region	Region 0	Region 1	Region 2	Region 3	Total
Percentage	18	36	28	18	100

In Table 8, which shows the peering locality of the simulation with hit movies, locality is even better, since P2P is more efficient in this scenario. Here the P2P traffic stays inside each area, and there is no P2P traffic between nodes of two routers and more than 80% of peers are within the same L2L3 switches. This leads to a hypothesis that adding a direct link between access switches of same L2L2 switches may improve the network efficiency even more, since traffic can travel directly between to access switches instead of passing two links and L2L3 switches.

Table 8: Peering statistics related to hybrid model with hit movies.

Region	Region 0	Region 1	Region 2	Region 3	Total
Percentage	34	52	14	0	100

Figure 21 shows an access network topology which is similar to mesh instead of tree. We believe adding the link shows with dashed line, should improve the resource usage of access network with a efficient P2P design, but there was not enough time to assess it during this project.

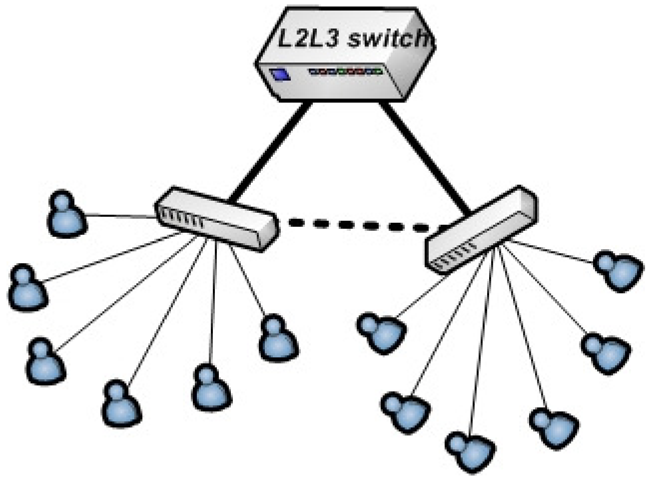

Figure 21: Part of an access network with mesh topology.

V Conclusion

1 Achievements

This report presents available client-server and P2P solution for real time video services with their weaknesses. At the preparation phase, a peering strategy and other tools (IPTV server and multicast streaming) have developed for simulating access network with the required capabilities to test available hypothesises. Also User profiles were defined based on some statistics on user behaviour and Sweden population. Physical topology of network was defined using deployed access networks in real world.

A hybrid method was proposed and tested during the project. Simulation shows that using proposed method cause a considerable decrease on the VoD server link which leads to more efficiency and scalability. Also, report shows that P2P techniques use the wasted uplink bandwidth to increase network efficiency. Moreover, the effect of locality awareness technology has been assessed during the project. It can be seen from the result that locality awareness techniques will decrease the inter area traffics and help to control the P2P traffic to keep it local.

Finally, the report proposed and simulated a solution for time shifting service for TV programs using a P2P-based video content distribution compatible with IPTV and VoD services. This makes it possible for users to watch their favourite TV shows whenever at any time within broadcast time, while save network resources such as the VoD server bandwidth. The catch-up solution is much more efficient than available methods.

2 Future work

The download strategy used in this project can be improved by using proposals such as TOAST model. Also capabilities such as VCR functionality, admission control for Quality of Services can be added and tested to the current solution. Handling flash crowd by replicating content before head or other solution is one important factor in service quality and efficiency that is not considered in this report. Peer disconnection and different seed time, also should be considered for the continuation of this work. Also lots of work can be done around movie popularity distribution and user profiles to have more realistic results.

In this report Tree topology is used for access network, which can be expanded toward a mesh topology in the future researches to increase the efficiency of resources usage and provide more locality to the P2P traffic.

VI References

[1] L. Bodenheimer, E. Pech and P. Pfeffer. "IPTV: Technology and Development Predictions", *Fiber and Integrated Optics*, 1096-4681, Volume 25, Issue 5, pp 325-346, 2006.

[2] Y. Tu, J. Sun and M. Hefeeda. "An Analytical study of Peer-to-Peer Media streaming systems", *ACM Transactions on Multimedia Computing, Communications and applications,* Volume 1, Issue 4, 2005.

[3] T. Look, S.C. Preiss and M. Pelka. "P2P-Next: Deliverable number 2.1.1, Requirement Report, Version: 2.0", 2008, <http://www.p2p-next.org/download.php?id=B40CCC39147226C8B72287C34BBFFFC9 > , (2010.08.01).

[4] K. A. Skevik, V. Goebel and T. Plagemann. "Analysis of BitTorrent and its use for the Design of a P2P based Streaming Protocol for a Hybrid CDN", Department of Informatics, University of Oslo, 2004.

[5] O. Findahl. "Svenskarna och Internet", World Internet Institute, Digital version 1.1, 2009.

[6] X. Yang and G. de Veciana. "Service Capacity of Peer to Peer Networks“, Department of Electrical and Computer Engineering, The University of Texas at Austin, 2004.

[7] J.A. Pouwelse, P. Garbacki, D.H.J. Epema and H.J. Sips. "The BitTorrent P2P file sharing system: Measurements and Analysis", Department of Computer Science, Delft University of Technology, 2005.

[8] S. Tewari and L. Kleinrock. "Analytical Model for BitTorrent-based Live Video Streaming", *in Proceedings of IEEE NIME Workshop*, 2007.

[9] H. Xie, A. Krishnamurthy, A. Silberschatz and Y.R. Yang. "P4P: Explicit Communications for Cooperative Control Between P2P and Network Providers", P4PWG Whitepaper, 2007.

[10] K. P. Gummadi, R. J. Dunn,S. Saroiu, S. D. Gribble, H. M. Levy and J. Zahorjan. "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload", *In SOSP'03*, 2003.

[11] Y. Liu, X. Liu, L. Xiao, L. Ni, and X. Zhang. "Location Aware Topology Matching in P2P Systems", *in Proceedings of IEEE INFOCOM*, 2004.

[12] K. Sripanidkulchai, B. Maggs and H. Zhang. "Efficient content location using interest-based locality in peer-to-peer systems", *In INFOCOM'03*, 2003.

[13] H. Deshpande, M. Bawa and H. Garcia-Molina. "Streaming live media over a peer-to-peer network", Technical Report, CS-Stanford 2002.

[14] D. Tran, K. Hua and T. Do. "A Peer-to-Peer Architecture for Media Streaming", *IEEE JSAC*, 2003.

[15] B. Cheng, X. Liu, Z. Zhang and H. Jin. "A Measurement Study of a Peer-to-Peer Video-on-Demand System", China, 2007.

[16] E. Damiani, S. DeCapitanidi Vimercati, S. Paraboschi, P. Samarati and Fabio Violante. "A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks" Italy, 2002.

[17] M. Castro, P. Druschel, A.M. Kermarrec, A. Nandi, A. Rowstron and A. Singh. "SplitStream: High-bandwidth multicast in cooperative environments", *in SOSP 2003: Proceedings of ACM*, pp 298-313, 2003.

[18] S. Banerjee, S. Lee, B. Bhattacharjee and A. Srinivasan. "Resilient Multicast using overlays", *IEEE/ACM Transactions on Networking (TON)*, Volume 14, Issue 2, pp 237-248, 2006.

[19] V. Janardhan and H. Schulzrinne. "Peer assisted VoD for set-top box based IP network", *in Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Kyoto, Japan, 2007.

[20] Y. Huang, Y. F. Chen, R. Jana, H. Jiang, M. Rabinovich, A. Reibman, B. Wei, and Z. Xiao. "Capacity Analysis of MediaGrid: a P2P IPTV Platform for Fiber to the Node (FTTN) Networks", *IEEE Journal on Selected Areas in Communications, Special Issue on Peer-to-Peer Communications and Applications*, Volume 25, Issue 1, 2007.

[21] Y. R. Choe, D. L. Schuff, J. M. Dyaberi and V. S. Pai. "Improving VoD server efficiency with BitTorrent", *in Proceeding of the ACM 2007*, pp 117-126, 2007.

[22] C. Dana, D. Li, D. Harrison and C. N. Chuah. "BASS: BitTorrent-assisted streaming system for vodeo-on-demand", *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2005.

[23] Wikipedia for Voddler, < http://en.wikipedia.org/wiki/Voddler> , (2010.07.28).

[24] R. Perlman. "Models for IP Multicast", *in Proceedings of IEEE International Conference on Networks*, 2004.

[25] K. Hua, Y. Cai and S. Sheu. "Patching: a multicast technique for true video on demand services", *in Proceedings of ACM Multimedia*, 1998.

[26] L. Gao, D. Towsley and J. Kurose. "Efficient schemes for broadcasting popular videos", *in Proceedings of International Workshop on network and operating system support for digital audio and video*, 1998.

[27] F. Thouin and M. Coates. "Video-on-Demand Networks: Design Approaches and Future Challenges", *IEEE network*, Volume 21, Issue 2, pp 42–48, 2006.

[28] T. Look, S. C. Preiss, M. Pelka. "P2P-Next, Deliverable number 2.1.1, Requirement Report", Version 1.0, 2008.

[29] A. Rudstrom and M. Sjolinder. "Capturing TV user behavior in fictional character descriptions", technical report No T2008:11, SICS Co., Sweden, 2008.

[30] J. Grudin and J. Pruitt. "Personas, participatory design, and product development: An infrastructure for engagement", *in Proceedings of the Participatory Design Conference 2002*, pp 144-161, 2002.

[31] S. Kujala and M. Kauppinen. "Identifying and selecting users for user-centered design", *in Proceedings of Nordichi'04*, Tampere, Finland, 2004.

[32] T. F. Tan and S. Netessine. " Is Tom Cruise Threatened? Using Netflix Prize Data to Examine the Long Tail of Electronic Commerce", Wharton Business School, University of Pennsylvania, Philadelphia, 2009.

[33] M. Chesire, A. Wolman, G. Voelker and H. Levy. "Measurement and analysis of a streaming workload", *In Proceedings of USITS*, 2001.

[34] C. Loeser. "Demand Forecasting in P2Pand GRID Systems", *In Proceedings of the10th International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA-2004)*, Orlando, Florida, 2004.

[35] C. Griwodz, M. Bär and L. C. Wolf. "Long-term Movie Popularity Models in Video-on-Demand Systems or The Life of an on-Demand Movie", *ACM 1997*, Seattle, USA, 1997.

[36] W. Page and E. Garland. "The long tail of P2P", *Economic Insight*, Issue 14, 2009.

[37] L. A. Adamic and B. A. Huberman. "Zipf's law and the Internet", *Glottometrics 3*, pp 143-150, 2002.

[38] Y. J. Kim, T. U. Choi, K. O. Jung, Y. K. Kang, S. H. Park and Ki-Dong Chung. "Clustered Multimedia NOD : Popularity-Based Article Prefetching and Placement", Department of Computer Science, Pusan National University, Korea, 1999.

[39] H. Yu, D. Zheng, B. Y. Zhao and W. Zheng. "Understanding User Behavior in Large-Scale Video-on-Demand Systems", *in Proceeding of EuroSys 2006*, Leuven, Belgium, 2006.

[40] J. Chung and M. Claypool. "NS by Example", <http://nile.wpi.edu/NS/>, (2010.07.25)

[41] K. Fall and K. Varadhan. "NS2 manual", VINT Project, 2000, <http://www.isi.edu/nsnam/ns/ns-documentation.html>, (2010.07.25).

[42] J. Yi. "The delivery of video on demand using peer to peer technology", master thesis at KTH University, Sweden, 2009.