PhD-FSTC-2013-15
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defense held on 14/06/2013 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Bernd KLASEN
Born on 18 August 1979 in Gerolstein, Germany

SOCIALLY AUGMENTED CONTENT DISTRIBUTION IN HYBRID NETWORKS

Dissertation defense committee

Dr. Steffen Rothkugel (Dissertation Supervisor)
Associate Professor, University of Luxembourg, Luxembourg

Dr. Pascal Bouvry (Chairman)
Professor, University of Luxembourg, Luxembourg

Dr. Peter Sturm (Vice Chairman)
Professor, University of Trier, Germany

Dr. Adrian Andronache
Mobile Communication Security Solutions, DATEV eG, Nürnberg, Germany

Dr. Ingo Scholtes
Chair of Systems Design, ETH Zurich, Switzerland

Wahrlich es ist nicht das Wissen, sondern das Lernen, nicht das Besitzen sondern das Erwerben, nicht das Da-Seyn, sondern das Hinkommen, was den grössten Genuss gewährt.

Carl Friedrich Gauss

Ich habe keine besondere Begabung, sondern bin nur leidenschaftlich neugierig.

Albert Einstein

Acknowledgements

Having accomplished the research and studying this thesis involved and finally holding the complete document in my hands, I want to thank all those people who helped me to reach this point. My deep and heartfelt thankfulness goes to my family, most notably my wife Ruth and my son Pepe for their support, understanding, patience and especially for their love that gave me the required energy for my work. Further my parents, grandparents and all other family members for giving me support on my way from the very first day. I am also grateful for the support, inspiration and guidance of my supervisors Peter Sturm from the University of Trier and Steffen Rothkugel as well as of my adviser Frank Zimmer. Moreover my gratitude goes to my friends and colleagues Jean Botev, Stefan Elsen, Markus Esch and Ingo Scholtes for the inspiration and encouragement they gave to me and to Ingeborg Weitzel.
Further I thank the National Research Fund (FNR) of Luxembourg and my employer SES TechCom for their financial and professional assistance, the Research Group System Software and Distributed Systems of the University of Trier and the Faculty of Sciences, Technology and Communication at the University of Luxembourg. Concluding I want to put a special emphasis on the support and contribution from the University of Trier, that provided resources, working environment and great personnel even though it was not always mentioned in official documents and presentations. I am very grateful for this generous support.

Kurzfassung

Der weltweite Datenverkehr im Internet erfährt einen immensen Anstieg, dessen Ende nicht absehbar ist und möglicherweise auch niemals kommen wird. Ein Aspekt, der wesentlich zu dieser Entwicklung beiträgt, ist die wachsende Zahl der an dieses Netz angeschlossenen Nutzer und Geräte. Hinzu kommt ein beständig steigender Bandbreitenbedarf vieler Internet-Anwendungen. Insbesondere trifft dies auf online Video-Dienste zu. Die zugehörigen Videos wachsen nicht nur in ihrer Zahl, sondern insbesondere auch in der Qualität, was zu einem signifikanten Anstieg der Dateigrößen führt. Aber auch Softwaredownloads, -updates und -patches, sowie Online-Spiele, Cloud-Computing und -Speicher tragen maßgeblich zum Anstieg des Datentransfervolumens bei.
Ob es in Zukunft stets möglich sein wird die bestehende Infrastruktur ausreichend schnell auszubauen um dem steigenden Bedarf gerecht zu werden, ist unsicher. In jedem Fall würde dies ein kostenintensives Unterfangen werden und die Folgen des daraus resultierenden Anstiegs der Komplexität des Netzes schwer abzuschätzen sein. Überdies führt eine solche Maßnahme lediglich zu einer Verschiebung der Kapazitätsgrenzen, stellt jedoch stets nur eine temporäre Lösung dar, sofern die Skalierbarkeit nicht verbessert wird.
Für eine Vielzahl der über das Internet übertragenen Daten kann ein hohes Maß an Redundanz beobachtet werden. Beliebte Videos, die auf Portalen wie YouTube angeboten werden, können ohne Weiteres etliche Millionen Betrachtungen während der Dauer ihrer Existenz verzeichnen. Oftmals ereignen sich diese sogar innerhalb kurzer Zeitintervalle [63]. Aktualisierungen weit verbreiteter Betriebssysteme werden von hunderten von Millionen Clients bezogen [45]. Bedingt durch die zeitlichen Differenzen zwischen den einzelnen Anfragen und den Mangel an Alternativen zum Unicast Übertragungsmodus wächst der dadurch verursachte Datenverkehr linear mit der Zahl der Empfänger. Dies stellt

nicht nur für das Übertragungsnetz eine erhebliche Herausforderung dar, sondern darüber hinaus auch für die Server der Content Provider. Dieses Problem wird verschärft durch die tageszeitabhängigen Schwankungen der Nutzeraktivität, die je nach Art der Anwendung unterschiedlich sein kann, jedoch üblicherweise starke Spitzenlasten in den späten Abendstunden aufweist und am frühen Morgen vor Beginn der Arbeitszeiten ihre Talsohle erreicht.
Diese Dissertation argumentiert, dass eine Minderung der Höchstlasten im Internetdatenaufkommen sowie eine Reduzierung der Gesamtmenge an übertragenen Daten durch eine Verlagerung der redundanten Transfers in ein broadcast Netzwerk, insbesondere Satellitennetze, erreicht werden kann. Dabei werden unicast und broadcast Netzwerk zu einem sogenannten hybriden Netzwerk, welches eine dynamische Vermittlung der Datenpakete über jedes der beiden Netze ermöglicht. Basierend auf dieser Infrastruktur wird ein neues, auf Peer-to-Peer (P2P) Technologie aufbauendes sozial angereichertes Datenverteilungsmodell definiert, welches die Auswahl des geeigneten Übertragungsnetzes und die eigentliche Zustellung der Daten regelt. Darüber hinaus strebt es eine Bereitstellung der Dateien in Nutzer-Zwischenspeichern vor der Entstehung des konkreten Bedarfs an, wodurch eine weitere Angleichung der Schwankungen der Internet- und Serverauslastungen ermöglicht wird. Dies geschieht unter intensiver Ausnutzung sozialer Strukturen anhand von Berücksichtigung sozialer Bindungen, Ähnlichkeiten und wechselseitiger Beeinflussung zwischen Individuen. Es wird gezeigt, dass durch diesen Ansatz eine beträchtliche Reduzierung des Internet Datenaufkommens erreicht werden kann, während dafür nur eine verhältnismäßig geringe Satellitenkapazität benötigt wird. Gleichzeitig wird die von Nutzern erfahrene Servicequalität im Sinne verkürzter Übertragungszeiten verbessert.

Abstract

The global Internet traffic is experiencing a tremendous growth whose end can not be foreseen and potentially will never come. One aspect that contributes to this development is the increasing number of users and devices that are connected to this network. In addition, most Internet applications get more and more bandwidth demanding. This especially applies for online videos services. The corresponding videos do not only grow in quantity but particularly in quality, which leads to a significant increase in file sizes. But also software downloads; updates and patches as well as online games, cloud computing and storage account for a notable growth in Internet traffic.
Whether or not it will always be possible to upgrade the existing infrastructure fast enough to keep up with the increasing demand is uncertain. On any account this undertaking will be expensive and the consequences that arise from the thereby growing complexity of the network are hard to predict. Moreover, this will always just push the limit a little further, but as far as scalability is not improved, this will always remain a temporary solution.
For many of the files delivered through the Internet a large degree of redundancy can be observed. Popular videos provided on sites like YouTube may easily have several millions of views during their lifetime, often even within a short time period [63]. Updates of popular operating systems are downloaded by hundreds of millions of clients [45]. Due to timely discrepancies of the requests and the lack of alternatives to the unicast transmission mode, the induced traffic increases linearly with the number of downloads. This is not only a challenge for the network used for delivery but also for the servers of the corresponding content providers. The problem is aggravated by the time-of-day effects in user activity which varies for different classes of content but commonly has a high peak in the late evening and bottoms out in the early morning hours before the working

hours.
This thesis argues that the peaks in Internet traffic can be truncated and the aggregated amount of bytes transfered be significantly reduced by shifting this redundant traffic to a broadcast network, particularly satellites. Thereby the unicast and broadcast network are combined by means of a hybrid network that allows dynamic routing of data via either of them. Based on this infrastructure a new socially augmented content distribution model relying on peer-to-peer (P2P) technology is defined that manages the selection of the appropriate network and the actual delivery of data. Further it aims at delivering files to user caches prior to the actual demand and thus can further level out the fluctuations in daily Internet and server load. Thereby it makes intensively use of social network structures by means of capitalizing information about social ties, similarities and mutual influence among individuals. It will be shown that by this approach the Internet traffic can be significantly reduced while only requiring a comparatively small amount of satellite capacity. At the same time the user perceived quality of service is increased by means of a reduced delivery time.

Contents

CHAPTER 1

Introduction

1.1 Motivation

The rise of the Internet and WWW

More than 40 year have gone by since the first attempts of inter-networking by means of message exchange between few computers. Roughly 20 years later the birth of the World Wide Web (WWW) was celebrated. Just a few years later many companies and private households owned an Internet connection and thus access to the WWW. Since then both became more and more widely available and have been adopted by an enormous and ever growing number of users worldwide. In conjunction with a vast amount of web pages, online service and content providers a point has been reached that corresponds to a prognosis printed in an article of an issue of Popular Science in the year 1970 [106]. Therein Wernher von Braun cites Arthur C. Clarke with the following words: *"Imagine a console in your office, he says, combining the features of a Touch-Tone (pushbutton) telephone, a television set, a Xerox machine and a small electronic computer. Tuned in to a system of synchronous satellites, this console will bring the accumulated knowledge of the world to your fingertips."* Even though the satellites are rather the exception for online access, the Internet and the WWW have become an essential part of many people's life, not only in offices but also for leisure time in a similar extent.

On the indispensability of the Internet

These days it seems difficult to even imagine a life without Internet. The reason for this is that the increasingly ubiquitous access gave rise to a paradigm change in the way we access digital information. While several years ago people used to download files and store them locally for the sake of better access to it, today this has been inverted. People do not store online content anymore since it is often much simpler to just download it again. Beyond that more and more private content is shifted from local hard disks to online storage in order to have it accessible from everywhere and from any device. Even applications are more and more substituted by online services and browser applications. According to [97] there is a noticeable trend to move parts of our personal memory into the internet, since there seems to be little need for remembering facts that can be looked up at any time.

Traffic explosion, the consequence of success

This trend necessarily leads to an increasing amount of data transfered through the Internet. This is reflected in measurements and prognoses on future global Internet traffic such as shown in figure 1.1. Such a growth pushes the existing

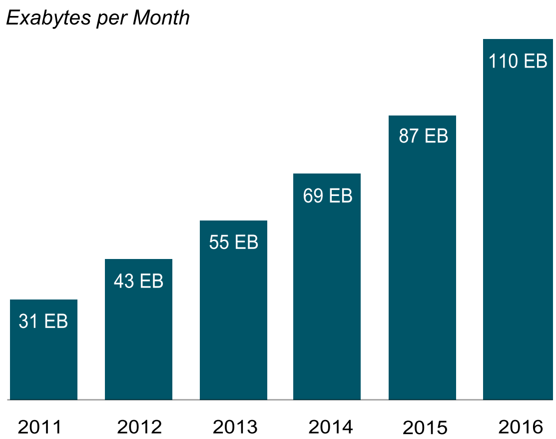

Figure 1.1: Internet traffic forecast provided by [27]

Vulnerability of complex networks

network infrastructure consistently to its limits of capacity and thus entails the demand for upgrades. However, adding nodes and links to a network increases its complexity and leads to increased risk of functional failures, as stated in [50] and [31]. In the past several notable failures of Internet services and infrastructure have been observed at recognized providers which might have been considered as highly reliable. One example is the Google failure 2009[1], the complete failure of Level 3's backbone in 2005[2] which had worldwide consequences or the breakdown of the popular video and voice communication service Skype in 2007[3]. While it was possible to solve all these problems within a relative short amount of time, they nevertheless caused immense deficits for businesses and costs for carriers as well as annoyance for private users.

[1]On 14.05.2009 a change in routing behavior made Google's services unavailable for approximately 14% of worldwide Google users.

[2]For more than two hours, the backbone infrastructure of the US IP-Carrier Level 3 was out of service on the 21.10.2005.

[3]The P2P based service of Skype failed for three days on 16.08.2007, which paralyzed the communication for many private and business users.

What can be derived from these events is—in the style of Ockhams Razor—that better than increasing the systems complexity by constantly upgrading is to use existing facilities more efficiently in order to achieve the goal of coping with future Internet traffic.

Implications on other delivery networks

The aforementioned change in the way we access content—that is ubiquitously, on demand and highly individualized—potentially put a threat on other technologies that are not designed for this new paradigm. A very prominent example are television broadcasts that use static program schedules. This discrepancy to actual trends in human behavior might lead to a falling acceptance in the future. Indications for this can be observed in the development of advertising reported by [57] and [96]. Furthermore the composition of Internet traffic depicted in figure 1.2 show a substantial growth in demand, particularly for

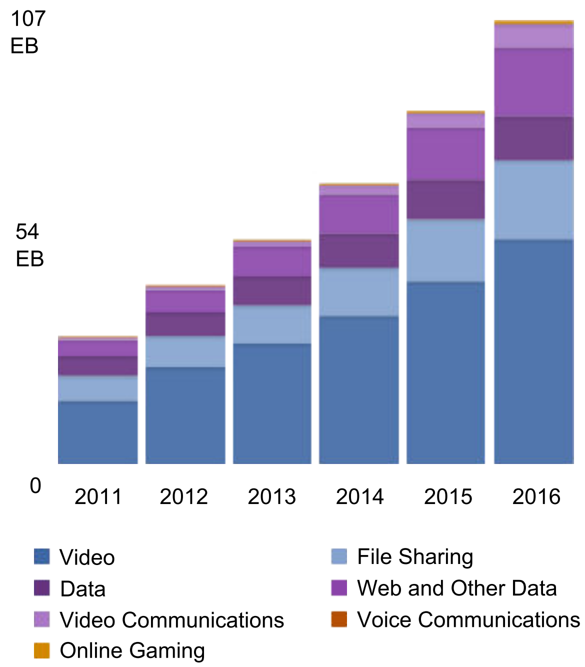

Figure 1.2: Prognosis of future Internet traffic classified by application type provided by [3]

online video. Should content providers or television channel operators decide to entirely switch their distribution channels towards the Internet, a considerable amount of capacity will become idle in broadcast networks. Despite this conjecture there is an aspect that reduces the efficiency of television broadcasts. The complete program schedule for a television channel is broadcasted, even if only very few people are watching. In a stricter sense, this kind of unrequested and ignored information transfer must be considered a waste of resources.

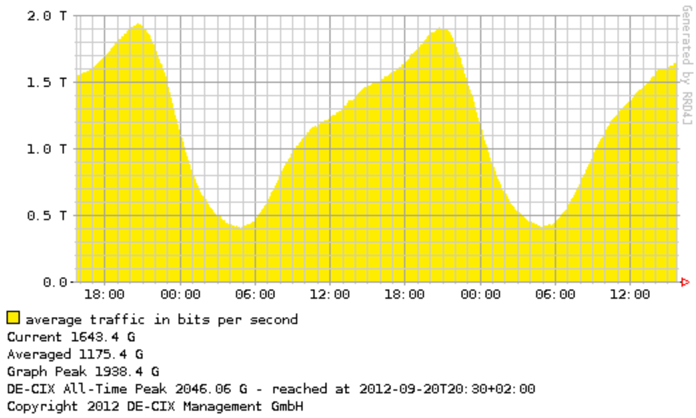

Figure 1.3: Internet traffic measured at an European IXP [34]

The tie between the problems of broadcast networks and the Internet

One could ask how this relates to the problems arising from growing traffic in the Internet. First, as figure 1.2 reveals, online video is one of the most growing types of application in terms of traffic volumes. Second, broadcast networks have fundamental advantages. The latter are most distinctive for satellite broadcasts. They provide a great coverage and a practically infinite scalability with regard to the number of recipients within a satellite's footprint. This makes it a very effective tool for distributing classical television content to millions of viewers. It is safe to assume that the audience rate will remain largely unaffected by a change of transmission medium in case there is no perceivable difference for the consumer. However, the moment when people start watching will differ,from a few milliseconds up to several days or even years. This skew depends on the general attractiveness and on the type of content. However, especially for highly desired and thus potentially popular videos it can be assumed that a significant number of views occur within a relatively short period of time after the provision. Since online services rely on unicast transmission, frequently requested content is a challenge for content providers and for the network itself since the load grows linearly with the number of viewers. This is further intensified by the non-uniformly distribution of user online activity over the course of a day. It exhibits high peaks alternating with periods of low activity as figure 1.3 shows.

Aspects of video request for online content

Outlining the problem of content distribution

In summary on the one hand there is a unicast network that is used for up to million-fold redundant transmission of the same data, on the other hand exists a broadcast medium that is used inefficiently in terms of temporarily serving only very few people. The latter reaches its climax in bidirectional satellite Internet access, where the broadcast is used to send individual content to just one user. Thus this not only affects videos but the whole spectrum of online content.

The intend of this thesis is to define a hybrid network infrastructure combined with a content distribution model that addresses the problems highlighted above. Thereby a fusion of both delivery networks is applied that allows a dynamic routing via either of them. This allows popular content to be broadcasted while individual and less often requested content is sent via unicast. Furthermore a prediction based in-advance delivery scheme is used that stores content in local user caches before a request for these files is actively made. As a result it reduces Internet traffic by avoiding peaks caused by very popular files. More so it allows to partially shift traffic from the times of high user activity (see figure 1.3) to calmer periods. This approach is not limited to television or video transmission in general, but can be used for arbitrary types of content. The entire process would be transparent to the user who will not be aware of the transmission channel that is used, same as most users today are not concerned by the routes the individual packets of their downloads might take. This is facilitated by introducing a small box, containing a small computer that takes care of the data reception from both networks and thereby brings the accumulated knowledge of the world to the users fingertips. And thus, by the integration of satellites, will finally make Arthur C. Clarke's prophecy come entirely true.

Hybrid networks as a basis for a new content delivery model

Closing the circle

1.2 Related Work

In their paper "Accelerating YouTube with video correlation" [26] the authors analyze the video correlation on YouTube and develop a P2P file sharing system with a focus on live streaming of small files and short videos. Thereby they find that the graph of related videos exhibits a high average clustering coefficient. Based on this finding they create a P2P file sharing system that establishes overlay networks not for each file but for sets of related videos. The evaluation also provided in this paper shows that they achieve their primary objective of reducing the server load significantly more than traditional P2P streaming protocols while still maintaining fluent video playback. The traffic in the Internet backbone and between the peers has not been taken into account.

Accelerating YouTube with video correlation

A multicast approach for a client/server based content delivery scheme for video on-demand applications is presented in [6]. Thereby the authors use a partitioning of videos into smaller chunks and do a server side request aggregation on chunk level. The paper puts much effort into the analysis of an appropriate request aggregation and delivery scheduling strategy. Thereby they find that an earliest deadline first (EDF) approach delivers best results and allows fluent playback. Their EDF strategy relies on the assumption that a video request implies the scheduling of the delivery of all its chunks. According to the viewing position, each chunk has a deadline. The basic idea is to deliver all chunks as

The effectiveness of intelligent scheduling for multicast video-on-demand late as possible in order to be able to aggregate as many requests as possible. While being intended for multicast delivery, this strategy can also be applied to the piece selection strategies in P2P systems. In this work onlyuser-requested videos are considered. The prefetching process is not covered.

P2Prec: a P2P recommendation system for large-scale data sharing

The authors of [37] and [36] propose a P2P system for recommendations based on social ties, e.g. friendships. This decentralized recommendation approach suggests documents for P2P download by means of calculating similarities between the user and neighbors or second level neighbors in the social graph. Besides the recommendation their system further classifies the peers according to their usefulness for potential future file exchanges.

Offloading servers with collaborative video on demand

In [43] a P2P based video on demand content distribution model is proposed and analyzed. It is stated that the objective is a reduction of server load. The authors highlight the difficulties in the P2P distribution of VoD content compared to P2P live streaming. These are the timely shifted requests and the resulting distinct viewing positions. Since according to their statistical data (from year 2003) the number of concurrent downloaders/viewers is insufficient, their approach relies on the recruitment of helpers, peers that support the download of other users with their idle capacity. Furthermore the authors present a new piece selection strategy by the name of "biased random". Thereby the next piece for download is chosen with a probability inversely to its replication rate. They argue that common approaches like *earliest first* or *rarest first* leaded to a piece distribution that was adverse to the delivery of VoD content. It is stated that the former favored the earliest pieces too much, while the latter did the opposite and thus caused stuttering playback.

Design and deployment of a hybrid CDN-P2P system for live video streaming

LiveSky, an operational content distribution network that incorporates P2P technology is described and analyzed in [110]. The system uses a static CDN backbone organized as a tree. The edge-servers act as seeders for the P2P content delivery. The exchange of files and their pieces is only allowed between clients connected to the same edge-server. The analysis provided in this paper shows that the presence of the edge servers as reliable seeders is essential to prevent rebuffering.

Should internet service providers' fear peer-assisted content distribution?

The impacts of P2P based content distribution on Internet service providers external traffic is investigated in [59]. The paper focuses on the benefits of locality awareness and of caching on the premises of Internet service providers (ISP). The authors highlight that common P2P solutions request data from distant peers even in case it was locally available (in the same ISP network). They state implementing locality awareness resulted in a performance gain for the users in terms of 50% faster downloads for 24% of all peers. At the same time it could reduce ISP's inbound traffic by more than 50% compared to a client/server model, outbound traffic by more than 80% compared the standard BitTorrent protocol. The authors recommend an active involvement of ISP and emphasize the potential of further improvements if clients did remain as seeder for a longer time.

Leveraging social networks for increased BitTorrent robustness

In [41] the utilization of social networks for avoiding free-riding in BitTorrent is analyzed. Thereby BitTorrent is modified to implement its own social network. The proposed approach relies on giving absolute priority to friends. It is stated that these changes make BitTorrent more robust against free-riders. Further the authors find that social sharing alone is not sufficient for the sharing of rare/unpopular content. Furthermore very long download durations for 20 to 30% of all peers are reported when the exchange is strictly limited to friends due to the power law degree distribution. In consequence many clients have not enough peers to achieve reasonable download speeds.

Prefetching strategy in peer-assisted social video streaming

The authors of [107] investigate prefetching and caching strategies for the distribution of online videos by capitalizing social networks. Thereby they capitalize information of previous download behavior of a corresponding user as well as of his close social contacts. For the caching three distinct cache classes are defined. Class 1 holds videos that are currently watched, class 2 those who are subject of a prefetching job and class 3 for all remaining videos. The authors propose different cache replacement strategies for these classes. These are no replacement for class 1, first in first out (FiFo) for class 2 and least frequently used (LFU) for class 3. It is further stated that the size of the cache has a crucial impact on the prefetching quality.

TRIBLER: A social-based peer-to-peer system

An approach for increasing P2P performance by exploiting social ties among peers is presented in [87]. The social network is capitalized for improving content discovery, content recommendation and downloading. Tribler implements an exchange of preference lists among peers, referred to as BuddyCast. It periodically selects a peer to exchange preference lists with. The authors state that in order to improve P2P content distribution durable and unique user ids are mandatory as well as persistent information such as social relations, uptime statistics and similarities in taste to other users. By using this information they implement a collaborative downloading strategy that consults helpers in order to improve download performance. It is stated that this approach can speed up downloads by a factor up to 6 for clients with a download bandwidth of 8Mbit. Especially the importance of a sophisticated coordination between downloader and its helpers is emphasized.

2Fast : Collaborative Downloads in P2P Networks

The collaborative downloading in Tribler relies of the 2Fast protocol introduced in [42]. It implements groups of trusted peers (buddies) that denote their idle bandwidth to speed up downloads of (social) group members. The authors argue that due to the asymmetric Internet connections of most users and due to the tit-for-tat policy of BitTorrent, peers could not saturate their download capacity, but it was limited by the upload bandwidth. Thus in 2Fast they allow peers to ask their *buddies* for support, who will then start downloading the corresponding file from other peers and send missing pieces to the inquirer. However, half of the helpers upload capacity is used in order to receive file pieces from other peers since they also have to satisfy the tit-for-tat principle. Nevertheless the results presented in this paper show that this strategy can result in a significantly increased download speed.

1.3 Contribution

Novelty of this research

According to the developments in the extend of Internet usage and the resulting increases in web traffic this thesis argues that new strategies for content delivery must be applied in order to handle future demand. Otherwise the high traffic and especially the peak loads will overstress the infrastructure. This not only affects the Internet backbone but further its edges in terms of Internet access service providers as well as content providers and thus in consequence also the Internet users. Several studies address this problem for one or more of these involved domains. This thesis proposes a solution that will tackle the problem for all mentioned fields. It is stated that this can be achieved by just making better use of existing networks. Besides the Internet itself this further incorporates broadcast and social networks.
The examination of the required preconditions, the design and specification of a corresponding content delivery model and the implementation of an adequate protocol comprises several research domains that result in contributions in various areas that are described separately in the following.

1.3.1 Content Distribution Model for Hybrid Networks

Proposal of a new content distribution model

A content distribution model using a hybrid network as underlying physical infrastructure is proposed. This model incorporates peer-to-peer structures in order to achieve a first level of scalability with regards to the server load on the content provider premises. As a novel concept additionally satellite broadcasts are being utilized for the distribution of highly popular files respectively pieces of those. Other approaches found in literature either use satellite broadcasts as a standalone solution decoupled from Internet, or for filling the caches of dedicated super seeders which are part of a static backbone or transform the physical broadcast into a factual unicast. The model proposed in this thesis uses both network in parallel, allowing traffic to be routed via either of them. By means of this new approach the solution obtains the aptitude of dynamic adoption to changing conditions of user requests. The utilization of the satellite broadcast in a true one-to-many manner brings scalability and coverage. This further relieves the content providers servers but also takes traffic from the terrestrial transportation network.

In-advance distribution prior to demand

In order to allow better aggregation of transmission requests by means of increasing timely correlation of content transfers an in-advance delivery concept is proposed. It makes heavy use of findings in the field of social interactions and mutual influence among individuals organized in social networks.
The benefits of this content distribution model are a better scalability, reduction of peaks in server and Internet traffic, a mitigation of the restraints adherent to small bandwidth Internet connections, enhanced traffic locality and faster downloads.

1.3.2 Peaks and their predictability in online video portals

Detecting peaks in YouTube access patterns

For the efficiency of the broadcasts the number of concurrent recipients for a specific file is essential. While this can be increased by means of the above mentioned in-advance caching, having knowledge about real world access patterns of users to online content is important to estimate the prospect of success of the proposed content distribution model. Thus statistical data of online videos hosted by YouTube has been collected and analyzed in order to determine the peakiness of video requests. based on the observations this thesis gives an estimation on the predictability of future peak probability for certain classes of videos.

1.3.3 Protocol implementing the content distribution model

SatTorrent protocol definition and evaluation

A P2P protocol is presented that implements the proposed content distribution model. This is done in a two-step approach. In the first stage the hybrid network by means of an additional broadcast channel is considered in the design of the SatTorrent protocol. This allows a comparison to existing P2P protocols, in particular to BitTorrent, by evaluating simulation results of a single file distribution process. This protocol can be implemented in a real world application as it is and thus allows providers of a corresponding solution an easy market entry due to a reduced complexity.

Social SatTorrent protocol definition and evaluation

In a next step the protocol is extended to Social SatTorrent. The integration of social networks allows clients to optimize their own file acquisition process. One facet is the improved prefetching by means of better predictions of future demand based on collaborative filtering and consideration of mutual influence among socially close connected users. This constitutes an exploitation of the causes for extraordinary content popularity for mastering the handling of its consequences with respect to server and Internet load. The results show the appropriateness of socio-aware application design, particularly in the field of content distribution and P2P networks, for coping with the consequences of increasing online activity.

CHAPTER 2

Introduction to Network Technologies

2.1 Introduction

Narrowing down the notion of networks

In our every day life we are confronted with arbitrary types of networks. Typical examples are street networks, food networks and telecommunication networks. For decades the latter have usually been mainly associated with telephone networks until the Internet entered the stage. Nowadays most people will probably first think of the Internet when talking comes about networks. This is due to the great importance this network gathered during the few years of its existence.

Structure of this chapter

But what exactly is this Internet made of? We will have a glimpse into the structure and technological aspects in the following section, followed by an introduction to other networks that are of significant importance for the research presented in this dissertation. One of these are *Satellite Networks*, which are capable of sending information to millions of users respectively devices all around the world (section 2.3) in just one or very few transmissions[1]. This is followed by the presentation of social networks (section 2.4), content distribution networks (2.5) and peer-to-peer networks(2.6). All of them play an important role for the research presented in this document.

2.2 The Internet

The Internet and the WWW

It can be safely assumed that almost everybody who will read this thesis has a certain idea about what the Internet is. Nevertheless a short outline is provided

[1]For a global coverage (except the poles) three satellites are needed, as stated in [10].

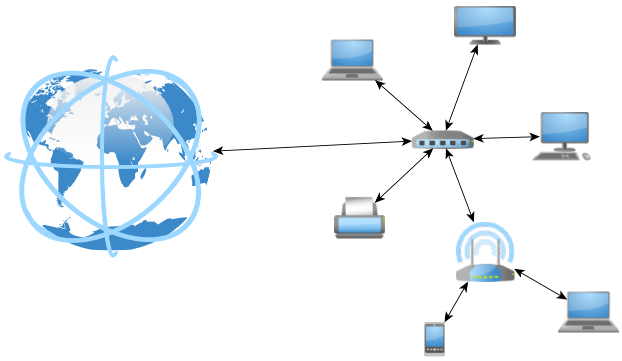

Figure 2.1: Exemplary home network

in this section, especially on those aspects that will be relevant in the course of reading this work. First, since they are often mixed up, it should be clearly distinguished between the *Internet* and the already mentioned *world wide web*. The former refers to the technical infrastructure, to a large set computers placed all over the globe which are interconnected by means of routers, gateways and of course the Internet protocol stack. The latter is an application layered on top of the Internet. In an abstract view it can be summarized as a collection of interlinked hypertext documents, hosted on servers which can be accessed by clients via the infrastructure provided by means of the Internet.

The rise of the Internet

The first Internet applications where limited to the exchange of very simple messages between a few computers located at scientific laboratories. The great breakthrough commenced with the advent of the aforementioned world wide web, better known under its abbreviation *WWW* or *web*. This allowed companies and later also private users to present themselves by means of web pages accessible for every Internet user around the world. While the first web pages exhibited a very simple structure and hardly contained more than text—some pleased their visitors by colored, potentially even blinking text—they where quickly enhanced by pictures, sounds and animations. Today every user can share messages, multimedia files like images, music or videos, links to other pages or documents even without having the slightest knowledge about the protocols, programming languages and technology that is involved.

The structure of the Internet

The term Internet is a short form for *interconnected networks*, which already describes its nature very well: It is made up of various smaller networks that are interconnected. The smallest entity that we consider here is a device equipped with at least one network adapter. This device may be part of a small private network within a household—consisting of just a few devices as shown

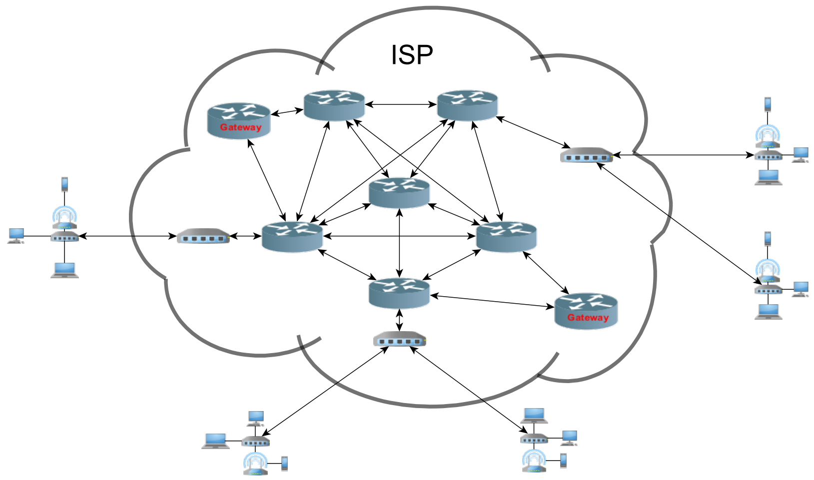

Figure 2.2: Coarse internal structure of an ISP

in figure 2.1—or of a bigger company network—composed of several hundreds of devices. Such a network potentially consists of smaller subnets which may use different topologies and technologies. The described structures are referred to as local area networks (LAN). A LAN or also a single device connects to an Internet service provider (ISP) in order to become part of the Internet. To allow communication between hosts located in different ISP networks the thousands of ISPs in the world must be interconnected. For this interconnection a complex structure evolved over the year that consists of different tiers of ISPs. Those who provide Internet access to end users are referred to as *access ISPs*. Their geographical expansion is limited—it usually stops at country borders as the latest—and thus they rely on higher tier ISPs that provide a connection to hosts located in remote access ISP networks. On the highest level we find the tier 1 ISPs which provide a globe spanning network that offers extremely high link capacities. If we make an analogy to public transport, a tier 1 ISP might correspond to an airline. And just as airports are only build at densely populated locations, the tier 1 ISPs do not expand their infrastructure in order to reach all potentially small ISPs in the world since there is little economical incentive. This gap is closed by the regional ISPs which can be seen as the counterpart to the railway companies in public transport if we assume that they are not selling tickets themselves but rather take over passengers from ticket shops and bus lines. The latter two would then correspond to the access ISPs.

Local area networks

The different tiers of Internet service providers

Each ISP network consists of numerous routers and links between them as illustrated in figure 2.2. Thereby there is a strict separation between *internal routers* that are responsible to route traffic within the providers network and

The inner structure of ISP networks

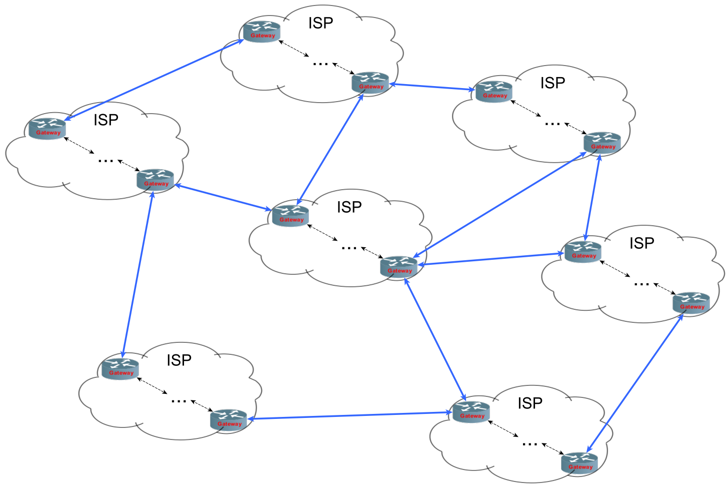

Figure 2.3: Exemplary interconnection of various ISP networks

the *gateway routers* that provide connectivity to other external networks (see figure 2.3). The places where the latter are located are referred to as *points of presence (PoP)*. In our transportation network example, these would be the airports, train stations and bus stops.

Interconnecting ISP networks

Sending traffic through a higher tier is usually charged. In order to reduce the costs, adjacent ISPs on the same level may directly connect. This approach is denominated as *peering* and the involved partners usually have agreements that makes the corresponding traffic free of charge. Internet exchange Points (IXP), often operated by a third party provider, offer the infrastructure to allow multiple ISPs to peer at a single location. This concentration of interconnections is on the one hand a simplification of routing and on the other hand potentially a reduction of path lengths since without IXPs each of the ISPs would probably only connect to a small fraction of the other providers.

Charges for inter-ISP traffic

An example for an interconnection of ISPs at different tiers is illustrated in figure 2.4. Thereby black links indicate a connection that is free of charge while on red links traffic usually will induce costs. Thus a message from the access ISP on the lower left in figure 2.4 to the one on the lower right might be sent via the tier 1 providers for a certain fee. Alternatively it might be passed from one access ISP to the next—probably via an IXP—until it reaches its destination for free. However, this comes at the cost of a higher delay and a higher risk of packet loss. Depending on the type of application, this might be tolerable or not. Further, an ISP might complain when it is repeatedly abused as a free transit network for other ISPs.

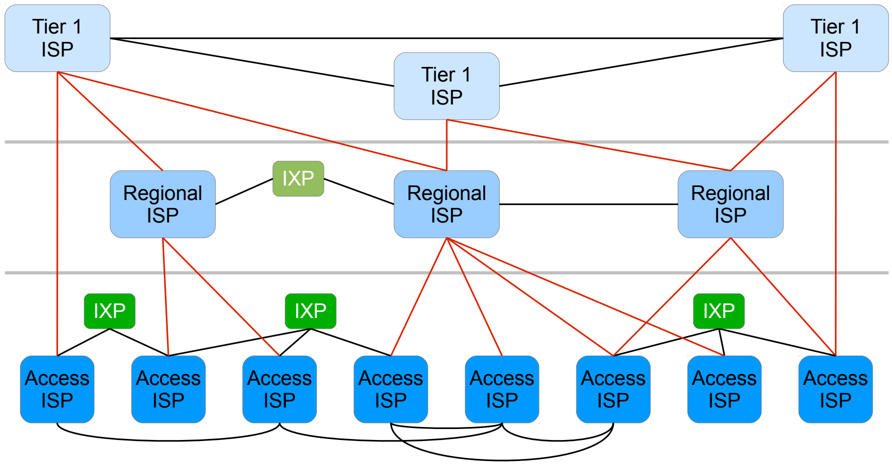

Figure 2.4: Connectivity of ISPs on different tiers

The described interconnecting of various types of networks and the involved host systems makes up todays Internet. Its link infrastructure—an approach of capturing its structural layout is illustrated in figure 2.5—is not a result of a clear design, but has rather evolved over the years to a highly complex but nevertheless robust and reliable system. The reliability is a result of the existence of alternative paths and the employed routing protocols. The latter can quickly detect link outages and thus use alternative routes in order to maintain the Internet's functionality.

Autonomous systems and their relation to ISPs

Internet routing is separated into two parts, routing within a so called *autonomous system (AS)* and routing between distinct AS's, with an AS being defined as *a network that is under one administrative control*. Correspondingly both are called *intra-AS routing* and *inter-AS routing* respectively. In order to enable addressing of autonomous systems they are being assigned an official worldwide unique *AS number*. Globally responsible for these numbers is the *Internet Assigned Numbers Authority (IANA)* which delegates number assignment to several regional authorities. In many cases, the network of one ISP is an AS, which might be the reason why both terms are often used synonymously. However, an ISP can as well be partitioned into several AS' and also a company network (which is not an ISP) can be a registered AS. Further not every access ISP is necessarily an AS that has its own official number.

Intra-AS routing

The fact that the Internet is partitioned in such smaller networks reduces the complexity of routing notably. Each AS is responsible for its internal routing. The internal host addresses and routers are well known which facilitates an efficient routing. The corresponding protocol for the intra-AS routing can be freely chosen. We recall the illustration given in figure 2.2 that also applies for

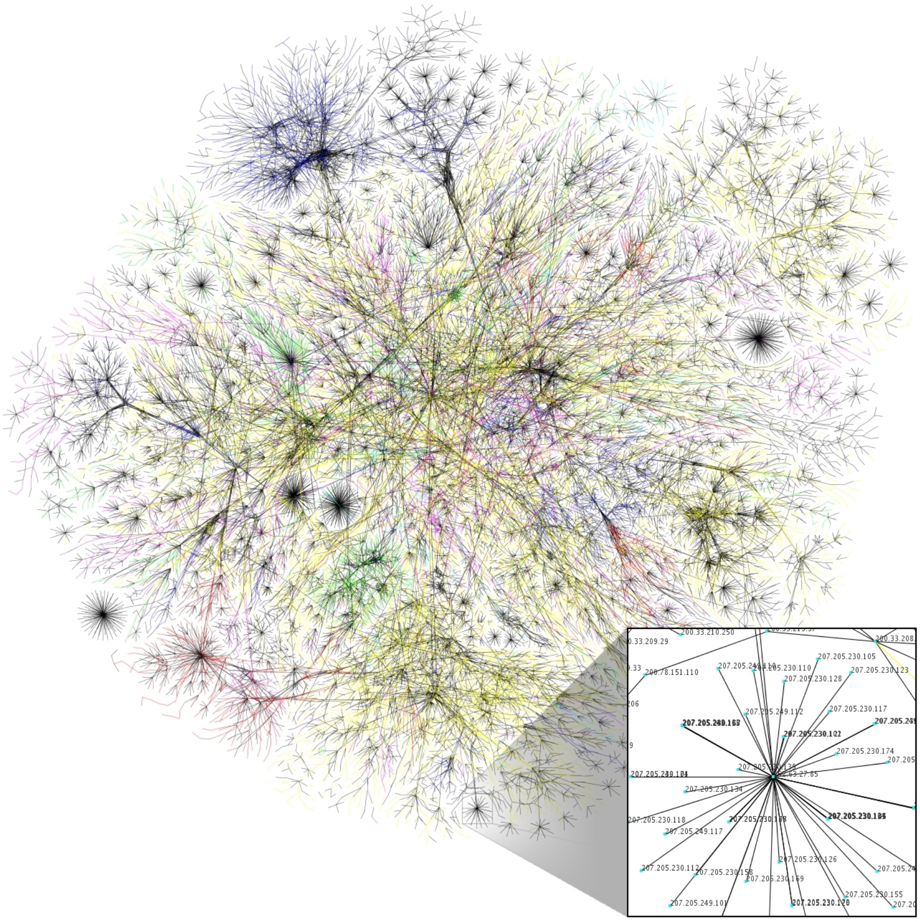

Figure 2.5: Partial map of the Internet based on the January 15, 2005 [21]

Inter-AS routing

autonomous systems. If a router receives a message that is addressed to one of the devices in the LANs directly connected to the AS, it forwards it to the next internal router on the path to the destination. If the destination is outside the AS the message is sent towards the appropriate gateway router. For the inter-AS routing the connected systems must use a common protocol. This must also enable the gateway routers to learn from adjacent AS' what IP address ranges can be routed to them. Thereby, whether the corresponding neighbor is the destination AS or just a next hop on the path is not of that gateway's concern. Both, inter-AS and intra-AS routing protocols, must be able to recognize failed links or routers and adapt to this situation in order to keep the Internet operational.

Further information about routing

We will not go deeper into routing and the protocols that are involved. The interested reader might find further information on routing in general in [56]; [80]; [83]; [76]; [89]; [13]; [48]; or [28], on the performance of Internet routing

Figure 2.6: Model of an artificial satellite in space (Source: SES `www.ses.com`)

in [71] or [86] and on computer networks in general (including routing) in e.g. [100] or [73].

2.3 Satellite Networks

What do we mean by satellite? A first broad definition.

Introducing the artificial satellites

In the sky above us—mostly invisible for the human eye—a plethora of satellites are flying. Most of us are aware of television (communication) or the navigation satellites for the widely used GPS respectively the soon being used Galileo [2]navigation systems. Of interest for this study are only the so called *artificial satellites*[10], defined as *man-made objects that are launched into an orbit*. The remaining are classified as *natural satellites*, e.g. the moon, the earth or any planet that is orbiting around its sun. The number of artificial earth satellites is about 6000, while approximately 5200 of those already reached their end of life but still are circulating earth in a so called *graveyard orbit*. The remaining satellites are operational and continuously monitored and controlled from earth stations. This becomes necessary since satellites recurrently need slight corrections on their flight in order to stay in the correct orbital position—a process referred to as *(orbital) station keeping*. This need arises since a satellite is exposed to several forces—such as atmospheric drag, lunar and solar gravitation—that cause a deviation from the desired position. Further many satellites deliver data for various purposes, e.g. scientific measurements or military observation, that must be continuously recorded and evaluated and thus require uninterrupted attention.

[2]The Galileo platform is announced to be operational by the end of 2014, as stated by [30].

2.3.1 Technical aspects

Flight altitude and its implcations

Satellites can fly at different heights respectively distances to earth, following an elliptical—in some cases circular—curve referred to as *earth orbit* or just *orbit*. Thereby they use the centrifugal force in order to countervail earth gravitation and thus to keep a constant height of flight. The smaller the altitude, the higher the influence of earth gravity. As an immediate consequence, a satellite's flying speed is inversely proportional to its distance to earth. The increased speed and reduced length of the orbit lead to a shorter orbital period, which is the time the satellite needs to once travel the complete orbit. The most used orbits—or ranges of orbits—are the following:

LEO Low Earth Orbit. Precisely it refers to altitudes up to 2000 kilometers. The corresponding orbital period is about 100 minutes. This orbit is mostly used for military and scientific satellites as well as for human spacecrafts.

MEO Medium Earth Orbit. Covers the range between LEO and GEO and thus allows altitudes from 2000 to 35,786 kilometers. However, most common are heights of 19,000 to 24,000 kilometers which correspond to orbital periods of approximately 12 hours. In this orbit we find primarily navigation satellites such as those used for GPS or Galileo.

GEO Geostationary Earth Orbit. Satellites flying on this orbit always stay fixed above a certain point on earth. This is only possible at a certain altitude of approximately 35,786 kilometers and perpendicular to the equator. Obviously the orbital period is exactly one (sidereal) day, respectively 23 hours, 56 minutes and 4.091 seconds, as described in [105].

Important aspects concerning a satellite's position

Despite the cost of bringing a satellite to its orbit, which increases with the height of this orbit, four aspects about flight heights are especially important. During their discussion the word *bandwidth* will be used. However, this word is used in the field of computer science as well as in satellite business with different meanings. In the latter if refers to a frequency range within a frequency band. In contrast computer scientists use this word in terms of a maximum bitrate that can be achieved on a specific link, measured in bits per second. This is referred to as *data rate* by the other group. In this thesis the definition used in computer science is applied.

Visibility depending on height and speed

Visibility How long a satellite is visible from a specific ground station depends on its height which determines its speed of flight. Geostationary satellites have a permanent visibility since they remain on a fixed position. For all lower orbits the satellites are moving regarded from a terrestrial reference system. Thus an antenna for reception must either be able to adjust on a potentially fast moving satellite and follow its path (*Tracking antennas*) or use alternative technologies, at the cost of signal strength.

Bandwidth The lower a satellite's orbit the higher is the flight speed. Consequently pointing an antenna exactly to a satellite becomes increasingly difficult with lower altitude. Thus for the end user market only antennas are used that do not need permanent adjustment, which makes them easier to build and thus affordable. However, as mentioned above, this significantly reduces the signal strength and thus the effective bandwidth that can be achieved.

Signal strenght is crucial for the effektive bandwidth

Delay The higher the distance to earth the greater is the lower bound for the delay, since the signal needs to cover this distance always twice. In case of an satellite in GEO this distance is 35,786 kilometers, the maximum signal propagation speed is 299,792,458 meters per second. Thus from sender to reception site it takes at least 238.74 milliseconds. This sums up with propagation delays of the satellite itself and send/receive hardware. Plus, in a bidirectional connection the signal must travel to the satellite and back twice, thus the aforementioned delay is doubled. This is absolutely negligible for television broadcasts, especially when all recipients have the same delay, even several seconds would not be recognized in most cases. However, it is very critical for voice applications like phone calls or also for most online games.

Latency is proportional to the distance

Coverage The higher the satellite's position above the earth, the greater the potential maximum area where its signals can be received, the so called *footprint*. Further there is the aspect of visibility that has been discussed above. A GEO satellite, due to its fixed position relative to the ground, always covers the same area. This makes it easier to determine which recipients are within its range at a certain moment, compared to satellites in lower orbits.

Coverage grows with flight height

Currently satellites use radio signals for communication. The frequency spectrum ranges from 230 MHz up to 75 GHz, while frequencies above 31 GHz are rather uncommon. Although higher frequencies are attractive in general, their usage is much more challenging [84]. In general the higher the frequency the shorter the wave and the reach, assuming the same energy is used for sending. Furthermore for higher frequencies of electromagnetic radiation it becomes increasingly difficult and at some point even impossible to penetrate walls or roofs. In consequence, most satellite signals require a direct line of sight between satellite and the ground antenna.

Used frequencies and their constraints

2.3.2 Applications of satellites

Prominent applications of satellites

The range of applications for satellites is broad and includes navigation, television, earth and space observation and many more. Discussing them all is beyond the scope of this work. In the remainder of this section we will take a closer look on two fields that are of interest for the research presented in this thesis, television broadcasts and Internet access via satellite, both belonging into the group of communication satellites.

Television Broadcasts

Position and functional layout of television satellites

The television broadcasts and by satellite that could be received directly by the end users—referred to as *direct to home (DTH)*—have been introduced in 1974. The corresponding satellites are positioned in a geostationary orbit and equipped with antennas for reception and for sending signals. A receive antenna is connected to a send antenna by means of a *transponder*, a unit that is able to amplify the signal and potentially convert the frequency. For analog television channels one transponder is needed per program. Due to compressed video encoding used for digital content, several channels can use one transponder. Technical details of transponders will not be further discussed at this point. What is important to know is that the common end user receiver hardware can only be tuned to one transponder. Since the majority of television satellite broadcasts use frequencies in the so called K_u band and a bandwidth—in terms of a frequency range—per transponder of 36 MHz, a net link capacity of approximately 36 to 40 Mbit per second is available per transponder[3]. Without changing the underlying technology, the bandwidth can only be increased by using several receivers in parallel. This is a common practice in order to allow consumers to watch an arbitrary channel while another one is being recorded.

Footprints and coverage

Television broadcasts are unidirectional. The signal is sent to the satellite from a ground uplink station and then broadcasted. The size and shape of the corresponding footprint is mostly determined by the satellites send antenna and can be as big as a whole continent. However, in densely populated areas they are often designed to be much smaller, not for technical but for legal and economical constraints. Nevertheless, there are usually millions of potential receiving users within a footprint. This one to many transmission represents a very efficient use of satellites. However, the fixed program schedules of television channels that allow this scheme do not fully match the viewing patterns of modern viewers. Many people are largely attracted by the highly individual on demand services offered online. This can be observed in viewing statistics of online videos but also in the development on the advertisement market where investments are shifting from television towards online services [96]. In order to be prepared for

Concerns about current television broadcast schemes

[3]Source: SES TechCom

potential future challenges, satellite providers should develop new strategies and delivery models that adopt to the new trends. Thereby the ideas and results provided in this thesis might be helpful.

Satellite Internet Access

History of Internet via Satellite

A notable attempt of Internet access via satellites started already in 1990. The company *Teledesic* started research and development of a system that was intended to use a constellation of 288 satellites in LEO. It was planned to allow Internet connections with 100 Mbit per user—back in that time a considerably high number—with an coverage of 95% of earth surface. However, the project has been discontinued in 2002 before the network was even partially operational [101].

State of the art

Today all companies that are offering satellite Internet access are using geostationary satellites. As already mentioned in the technical section the coverage of these satellites is great. Thus these systems allow Internet access at almost any place on earth where laying of cables would never have been an option, e.g. in sparsely populated area or on the sea. Further satellites provide a reliable and unrestricted Internet access in politically instable regions. It is distinguished between one-way and two-way satellite Internet access. While the former needs an additional terrestrial connection for the back channel, the latter uses the exclusively the satellite for both directions of data flow. Thereby they reach bandwidths of up to 18 Mbit per second for the downlink and 6 Mbit per second for the uplink. However, these are maximum values and since the communication relies on a shared medium and uses *time division multiple access (TDMA)*, these high data rates can only be achieved at short peaks and not over a longer period.

One- and two-way satellite Internet access

Physical limitations of two-way Internet access via satellite

As just mentioned two different modes can be used for satellite based Internet access, *two-way satellite communication* and *one-way broadcast*. For the remainder of this section two-way satellite Internet access will be considered. A corresponding satellite must be able to receive signals from many locations and then forward them to usually a singe point on earth where the data is passed into the terrestrial Internet. However, the uplink antennas installed at the end users have a comparably small diameter of down to 70 centimeters. This is, besides the already mentioned shared medium, another aspect that limits the maximum possible uplink capacity. When data is sent to an end user, it is routed to the service providers uplink station, sent to the satellite and then broadcasted. Since the corresponding data can be received at any point within the footprint, the data must be encrypted allowing decryption only to the desired receiver.

The application of satellites for a one-to-one communication transforms the broadcast to a unicast. Thus the potential of satellites is not fully utilized. Even

Issues of doing unicasts via broadcast medium

worse, in case that all customers of one satellite Internet access provider—just as an example assuming that these were 10,000—should request the same data at more or less the same time, this data would be transfered 10,000 fold. This is an suboptimal utilization of this medium. However, for general Internet usage due to reasons of privacy no aggregation of such transfers is possible.

2.4 Social Networks

Introduction to social networks

Usually we get in contact with various other individuals almost every day. Those of them we interfere with more frequently are being considered as being part of our social relations. The intensity of these connection may vary widely. Some people we see every day others only occasionally, some are perceived as highly important while others may be expendable. Further all of our contacts have their own social ties. All these connections together make up a complex network referred to as a *social network*. Such a network can be mapped to graph—in this case called the *social graph*—with the nodes being the individuals and the edges being social connections between them. Weighted edges can be used to express the intensity of the tie. Social scientists are exploring such structures and their relevance for decades. However, it has always been a challenging task to gather information about the social contacts and their intensity for a larger group of individuals.

New perspectives in social network research

This changed tremendously with the advent of online social networks (OSN). People can use these platforms to create their user profile and to establish connections to people they know and want to get respectively to stay in contact with. Thereby they give the OSN provider detailed information about their social network. However, users commonly only create a link to another user without rating it. Some OSNs extended their platforms in order to give users the possibility to classify a link e.g. as *close friend* or *professional contact*. Yet not all users provide this information, and if they do, it may not be complete, plus it is uncertain how precise such classifications may be. Thus additionally other approaches such as using the frequency of communication as an indication for the intensity of the connection are being used.

Social networks recently received great attention

Social networks are a natural fact and OSNs are the—probably only partial—virtual analogy to them which allow an easy acquisition and analysis. The reasons why social networks are so important are manifold and will be discussed in the following It will be highlighted how the knowledge about them can be used for improvements in general and for computer applications in particular. The numerous properties exhibited by social networks—as stated in [108]—that justify this great regard. First they exhibit a high clustering, which means we find

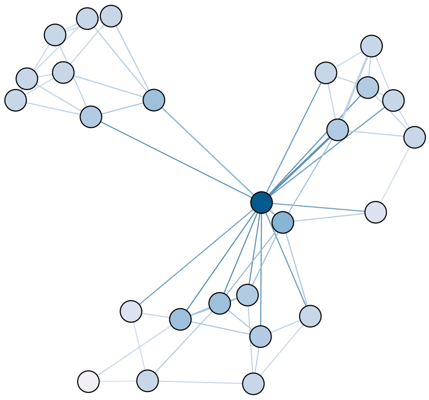

Figure 2.7: Small world example highlighting the clustering.

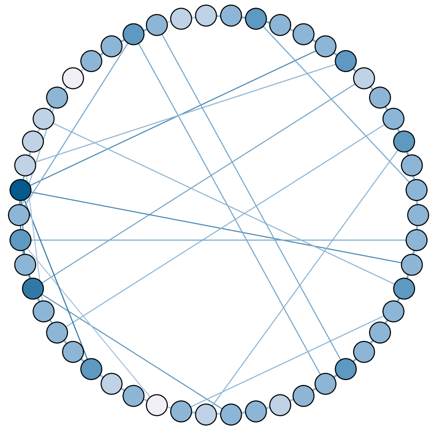

Figure 2.8: Circular layout of a Watts-Strogatz small world graph.

a groups of tightly connected nodes and sparse edges that interconnect these clusters. For example, the extraordinary high clustering in the Facebook social graph has been documented in [104]. Secondly, social networks have small average distances. This means for almost any two nodes that are picked from the network the length of the path between them is relatively short compared to the size of the network. The combination of both characteristics is being referred to as the *small world property* and the corresponding graphs as *small world graphs*. Figures 2.7 and 2.8 both show examples of the latter, the colors of the nodes reflect their degree. The different layouts illustrate the properties. The clustering can be immediately observed in figure 2.7. One might imagine finding a similar structure when considering the own friends, colleagues from work and people known from high school. Figure 2.8 points out the short average paths which result from the *shortcuts* that slice through the middle of the circle while most connections are between neighboring nodes on the circle. The small world property can be observed in many real world networks such as the Internet, the WWW, brains, transportation networks, power grids, human sexual contacts, professional collaboration and many others [79], [108].

Clustering in social networks

Besides this, social networks belong to the so called *scale-free* networks. These are characterized by a typical distribution of node degrees that follows a *power law* with an exponent between two and three. This implies that they have a few nodes with an extraordinary high degree—the so called *(social) hubs*—and a large number of nodes with a low degree. These hubs are mostly responsible for the connectivity between clusters. Remarkable is that these properties hold even for a random subset of a graph's nodes. Further this also applies if nodes are randomly added or removed, which is why they are called *scale*

Social network graphs are scale-free

Robustness of scale-free networks

free. As a consequence of this property those networks are extremely robust against random node failures since most likely the removed node will have a small degree and thus not integral for the connectivity of the graph. However, if hubs are selectively deleted the network will soon be separated into smaller, disconnected sub-nets. Thus they are vulnerable points that must be protected against targeted attacks as good as possible. Many of the aforementioned examples of small world networks are also scale free, such as the WWW, the Internet and power grids. All these networks are very reliable and we seldom get any notion about the occurring node failures, although serious system outages with cascading node failures happen occasionally.

Scale-freeness a result of evolution

All these networks have not been designed with scale-freeness and a power law degree distribution in mind, but this property has evolved. The fact that it can be found that often as a result of an evolutionary process might be considered as an indication that these networks have some remarkable advantages which make scale free small world networks so fascinating and valuable for many applications. Besides the already mentioned resilience, it has been observed that information can spread extremely fast within scale free in general and social networks in particular [85],[8], [35]. Further it can be observed that not only information is passed between nodes in a social network, but also personal opinions. And depending on the distance in the social graph but also on other factors such a physical proximity and intensity of the relation, there is a strong mutual influence among individuals in a social network [18], [47], [24], [23]. Knowledge about such details can help to improve predictions on the future development of trends and thus to improve content distribution [7].

Spread of information and mutual influence in social networks

2.5 Content Distribution Networks

Why content distribution networks are required

As already discussed in the introduction, the Internet experiences an enormously growth of users, content and also of per-user online activity. A first obvious problem is that this increases the general load due to a higher traffic. However, this trend is challenging also for online content providers whose content experiences a very high popularity. This applies to software distributors—for downloads of installation packages as well as for online updates—but especially to online video portals. While a slow throughput during downloading the updates for a game might just be annoying, it makes a video unwatchable due to stuttering playback or rebuffering in case it drops below the video bitrate and is thus not acceptable in this context. Examples for important online video portals are *MyVideo, Vimeo, Dailymotion, Youku, Todou, Metacafe* and *Youtube*, the latter being the most famous and the most popular video service. The videos their users upload and others can access are growing in size and quantity. Furthermore the number of viewers, and thus downloads, is growing. In addition the

Real time content is especially challenging existence of online social networks, where links are shared and recommendations are made, speeds up the request rates for popular videos and further increases the total number of viewers.

High request amounts require potent servers

In order to serve such high amounts of videos—in case of YouTube this exceeded four billion video requests per day[4]—an extremely high-performance data center is needed just to manage the requests. However, this is not the only problem to solve, the content delivery is also a very challenging task. The data of every video that is watched must be sent from the data center to the viewer who can reside at any place of the world. Consequently—at least when content of international interest is offered—a significant amount of data must be sent over very long distances, through several networks and even more routers. Besides the delay that usually grows with the distance, every hop on the delivery path increases the risk for packet loss and throughput bottlenecks. Both will result in stuttering video playback which must be avoided. What is more, since the route for each packet that is sent is not known in advance and usually will pass several autonomous systems, it is impossible even to estimate the minimal throughput, not to talk about guarantees. Further, sending all this data causes high costs for the content provider and also its ISP potentially might not even be able to handle such tremendous traffic at all.

Delay, lost packets and throughput must be optimized as far as possible

CDNs make video streaming work conveniently

Nevertheless it can be observed that especially the most popular providers of online videos and other often downloaded content are able to serve their viewers even at peak loads. This is achieved by using *content distribution networks (CDN)*, sometimes also referred to as *content delivery networks*. They are in charge of bringing data to end users for their customers—the content providers—at high speed and under any load conditions. In order to do this, two approaches exist, also described in [73], which are briefly introduced in the following.

Getting as close as possible to the recipients

Edge Caching The first relies on the installation of *edge servers* within several access ISP networks which are responsible for caching content near the end users, at the edges of the Internet. This allows caching of content close to all users connected to the corresponding ISP. However, edge servers are not present at every ISP in the world. For Internet users who are homed in those ISPs who do not have one installed, the best edge server in their proximity must be determined. It is worth noting that the involved measurements induce—depending on their strategy—some or significant extra Internet traffic. In [98] the authors investigated the quality of these measurements for Akamai, the biggest CDN using the edge caching approach who proclaims to have more than 100,000 of

On the performance of current CDNs

[4] Source:http://www.reuters.com/article/2012/01/23/us-google-youtube-idUSTRE80M0TS20120123

these servers worldwide located in 75 countries [1]. According to their results, Akamai really can identify notably faster paths and thus it seems to be worth the effort, obviously at least for Akamai and its customers. In contrast to the CDN technology described in the following paragraph, edge caching does not raise a demand for maintaining a network infrastructure with potentially long geographic distance links.

Providing a dedicated network allows guaranted throughput

Private High Speed Network The second approach implements few big data centers in proximity to the PoPs of important ISPs, mainly those of tier 1. There the CDN stores respectively caches its customers content. For interconnecting these data centers the CDN operators install their own high speed network. While this requires a notable effort, it gives the CDN operator full control over the network capacity and allow load monitoring. However, it can not bring the content as close to the end user as edge servers do. In consequence this CDN design can not reduce the traffic on the remaining path, that is between CDN and viewer.

Handing over requests to CNDs by DNS redirecting

Both CDN methodologies need to recognize user requests and serve them instead of the original server at the content provider. This can be achieved by letting the content provider's DNS server rewrite the DNS request in order to redirect it to the DNS server of the CDN, who can then answer with the IP of its own, best suited server for this request. We wont go further into detail of this technique here. A more detailed general description can be found in [73], a comprehensive explanation of the DNS redirecting of Akamai in particular is given in [98].

What current CDNs can provide

Current CDNs are able to reduce the server load of their customers, the content providers as well as to decrease their outbound Internet traffic by reducing redundant transfers of files. Furthermore they avoid the content being unavailable due to server overloading. In many cases this already means a great improvement for content providers and makes it reasonable for them to pay for content delivery rather than to upgrade their own infrastructure. Nevertheless, content provider and CDN can be represented by the same company. The most notable example for such a vertical integration is Google, who has its own highly potent CDN to deliver, besides much other data, the videos hosted at YouTube.

Limitations of current CDNs – Scalability and last mile bandwidth

However, while content distribution networks do a good job in delivering online data from servers to clients today, they do not solve the problem in general. No matter how powerful the installed hardware may be, the capacity of a CDN is limited, may it be the caching capacity of the edge servers, the storage on the data centers or the link capacity of the private network. In consequence current CDNs require upgrades when the number of users, files or the size of files exceeds the bounds of installed capacity. Further, they are neither able to reduce the traffic on the last mile connections to the end users nor can they overcome the bandwidth limitations on this final part of the delivery path. Thus there is still potential for optimizing content distribution networks.

2.6 P2P Networks

Limitations of the client/server principle

The traditional modus operandi for all communication between nodes in the Internet followed the client/server principle. On the one side there is a server waiting for requests, on the other side are a multitude of clients sending them. For many applications this is a practicable solution. However, this is not scalable, neither for growing numbers of requests, nor for increasing transfer volumes. The latter affects mostly the Internet connection of the server that will always have a limited bandwidth, no matter how big it might be. The number of requests, more precisely the requests per time unit, usually rather brings the server's hardware to its limits but might also cause performance degradation on the Internet transport or lower layers. These issues together with having a single point of failure (SPOF) are inherent problems of centralized approaches such as the client/server model. The alternative is a decentralized setup where every node acts as a client and a server at the same time. This approach is referred to as peer-to-peer. The involved nodes—the peers— donate their own resources—CPU time, storage, Internet link capacity—to the network and in consequence the available resources grow as the number of participants increases. One of the most common applications for P2P networks—and the one which is of interest for this thesis—is the exchange of files.

The peer-to-peer model, a distributed approach

Introduction to P2P file sharing

In P2P file sharing applications the files that are distributed are separated into smaller parts, most commonly referred to as *chunks*. For the ease of understanding, in the following we will discuss a scenario where just one file is exchanged. Initially the network has usually one server and an arbitrary number of clients and thus it seems similar to the traditional client/server approach. However, as clients finish downloading of chunks, they immediately become a server for these specific pieces themselves. Clients that are acting as a server in the way just described but do not have completed their download are called *leechers*. After the completion they are referred to as *seeders*. This strategy results in a quick increase of the available download sources for the involved clients and thus reduces the load on the one original seeder with regards to its own capability to serve requests as well as its Internet connection upload capacity. This load is now distributed over all the involved peers. The available resources increase as more peers join the network. Thus scalability can be pro-

Advantages of P2P file sharing over the client/server approach vided by P2P systems in case maintenance overhead and signaling traffic are kept within limits. Furthermore the file, or its chunks, are replicated several times within the network which makes an unreachable server or failing nodes tolerable, at least up to a certain extent. Important for the file availability is the replication rate of the pieces. If we assume that just one piece exclusively exists at only one node while for all others several hundred copies can be found, an outage of that specific node would make the file unavailable and all other pieces worthless.

Piece selection strategies

In consideration of this problem strategies have been developed that should provide an almost equal replication rate for all pieces of a file. Most commonly applied are random approaches and *rarest first*. For the latter, peers intend to download those pieces first which have the lowest replication rate. Obviously, in order to do this correctly they need information about the number of copies of each chunk within the swarm. However, such global knowledge is usually not available. Consequently a less accurate approach is used that is based on local information which—according to [74]—delivers good results.

Optimized piece selection for specific applications

Some applications might put further constraints on the download sequence of chunks, as for example P2P video streaming. Here clients need to obtain the data in chronological order to maintain a fluent playback. An approach that aims at achieving a balanced duplication while at the same time incorporating incremental chunk reception to avoid stuttering playback is described and analyzed in [112].

Potential issues of P2P file sharing

However, besides these valuable aspects there are challenges concerning P2P technology that have to be managed. First to mention there is the *file availability* which has already been described in the context of chunk selection strategies. If only some bytes of a file become unavailable the sharing fails. Unfortunately the involved clients are most commonly private computers with little or no reliability and thus might often fail unexpectedly. Further, as results in [75] show, such hosts exhibit rather short online times and therefore are frequently unreachable.

Churn in P2P networks

Therefor P2P networks have to deal with a situation where clients join and leave incessantly, a procedure often referred to as *churn*. However, P2P systems perform surprisingly well even under these conditions. When availability must be guaranteed other measures must be taken such as making the original seeders always available or implementing a structured, highly reliable backbone. Such approaches are described and discussed in [17], [19] and [16].

Free-riding, the problem of selfish peers

Another challenge for P2P networks is a third group of participants—besides the seeders and leechers—which is called *free riders*. Such clients are downloading data from peers but do refuse to upload. This behavior is a real threat for a P2P swarms. It was shown by [91] that in 2002 in the Gnutella Network 25% of all peers were free riders. The consequences of and possible counter-measures against selfish peers have been discussed comprehensively, e.g. in [61], [88], [5], [54], [58] and [60]. It can be concluded that—although they can not be thoroughly prevented—free rides can be restrained and thus it is strongly recommended to consider them in the design of a P2P protocol.

The reputation of P2P file exchange

Another aspect that damaged the reputation and thus also reduced—but in no way stopped—the success of P2P technology is the fact that they are often associated with illegal file sharing. While indeed P2P has been used in this manner frequently, such behavior is neither intended nor inherently promoted by the technology in general. However, it can be considered as a proof for the advantages of P2P approaches with respect to server load and content distribution to arbitrary numbers of users.

ISPs' concerns about P2P technology

However, most Internet service providers do not agree on the generally positive assessment of P2P users. While there is an undeniable benefit for content providers and consumers, the aggregated Internet bandwidth consumption is increased. What is more, since P2P demands its downloaders to provide upload resources in turn, ISPs are confronted with a growing amount of outbound traffic and consequently with increasing expenses. A popular, often disclaimed measure to limit the cost growth is throttling P2P traffic. In turn, P2P protocols implemented counter measures in order to hide from packet inspectors. Unfortunately this established a gap between ISPs and P2P users that also obstructed promising approaches to reduce the inter ISP traffic. The most notable example is a peer localization that is supposed to foster exchange with peers in close proximity and thus potentially keeps P2P exchanges within the same ISP network. As the authors of [59] state, when ISPs and P2P users cooperate, the cost induced by P2P traffic can be significantly reduced.

Focussing on BitTorrent

A plethora of P2P protocols has been developed and implemented in widely used clients. Describing all of them in detail is out of the scope of this thesis. However, one of them, namely BitTorrent, is the basis for the SatTorrent protocol that was developed as part of this thesis. As a consequence, the knowledge of BitTorrent is of great importance for the understanding of this work and thus is explicitly discussed in the following section.

2.6.1 BitTorrent

History of the BitTorrent protocol

The BitTorrent protocol has been published in 2001 by its developer *Bram Cohen*. It implements an unstructured P2P network which is easy to maintain. A notable aspect that distinguishes BitTorrent from other P2P protocols is its single file swarm approach. While other protocols maintain one large swarm, BitTorrent establishes an individual one—denominated as Torrent in this context—for each file[5] that is exchanged. As a consequence a client can be connected to numerous autonomous BitTorrent networks. The procedure for a host of sharing a file via BitTorrent and for a client joining a network respectively to start downloading a specific file is described in the following.

BitTorrent proceeding: Sharing a file

In order to share a file via BitTorrent a metadata file must be created which is referred to as the *torrent file*. As it is case for several other P2P protocols, BitTorrent partitions files into smaller parts. While these are often referred to as *chunks* in other protocols, the naming convention for BitTorrent is *pieces*. In order to optimize the transfer of pieces, they are further partitioned into smaller *blocks* which a commonly used size of 16 KByte each. The torrent-file contains all information that is needed to join the file specific overlay network respectively to start the download. That comprises the number of pieces the file consists of as well as the concatenated 20 byte hash values of all pieces. The latter makes every torrent unique and robust against undesired file modifications. Thus it is an unambiguous identifier for a specific version of a file.

BitTorrent proceeding: Multi and single file torrents

Further a torrent can be created in two different modes, the single file mode and the multi file mode. In case of the latter, the torrent contains a recommended name for a directory to save the files and a dictionary that provides the length, checksum (which is optional) and path relative to the specified directory for every file. In the single file mode this dictionary is not needed and replaced by three fields containing a recommended file name, a checksum (optional again) and the length of the file in bytes. In the remainder of this thesis only the single file mode is utilized. The complete structure of its torrent file is shown in table 2.1.

The BitTorrent tracker

The first entry *announce* contains the IP address of the so called tracker. A tracker can provide information about and keeps track of all downloads for one or more files shared via BitTorrent. It serves peers with an initial list of possible exchange partners and thus solves the bootstrap problem. In the official protocol only one tracker is allowed for a torrent, but extensions have been developed that allow multiple trackers [2] in order to avoid a single point of failure.

BitTorrent proceeding: Obtaining a torrent file

Clients intending to download a file must first obtain the corresponding torrent. This can happen via direct exchange between its creator and the potential downloader. However, this is often not a practicable method when the file is supposed to the shared with vast number of individuals that do not belong to a predefined group. Thus the common practice is to offer the torrent on a public web server where interested people can access it, often as part of a larger directory of various torrents. Well known examples for such a web servers are

[5]Protocol extensions exist that allow sharing numerous files with one torrent. However, this can be rather considered as an archive containing these files.

Field name	Description
announce	the URL of the corresponding tracker
info:	
piece length	byte count per piece
pieces	concatenation of all piece's hash values
name	recommended name
length	file length (in bytes)

Table 2.1: Torrent file structure.

Name	Description
info_hash	20 byte hash value of the info part in the torrent file
peer_id	a string providing an ID for the peer
ip	the peer's IP address
port	the port number the peer is listening on
uploaded	bytes uploaded by the peer
downloaded	bytes downloaded by the peer
left	bytes still missing
event	peer status (STARTED, COMPLETED, STOPPED, EMPTY)

Table 2.2: Structure of a GetRequest message

the ones from *The Pirate Bay*[6], *SuprNova*[7] or *IsoHunt*[8]. Unfortunately these servers are also heavily used for sharing illegal content that violates copyrights.

BitTorrent proceeding: The first tracker contact

After having downloaded the torrent, the client extracts the necessary information including the URL of the corresponding tracker. Now a *GetRequest*—sometimes also referred to as an *announce request*—is sent to this tracker. The structure of such a request message is illustrated in table 2.2. The *info* field identifies the file that is about to be downloaded. While for this purpose the content of the info field from the torrent could theoretically be directly used here, this would cause enormous message sizes. Since GetRequests are frequently sent, as we will see below, this must be avoided. Thus a 20 byte hash of this data is created and used for the GetRequest. Further it contains an ID, the IP address and the listening port of the client that is sending the request.

[6] http://thepiratebay.se/
[7] http://www.suprnova.org/
[8] http://isohunt.com/

Name		Description
interval		time interval (seconds) in which peers should send announce messages to the tracker
tracker id		a string providing an ID for the tracker (to be used in future announce messages)
complete		number of available seeders
incomplete		number of leechers
peers		dictionary with peer information
	peer id	the id of a peer (see table 2.2)
	ip	the IP address of the peer
	port	the port the peer is listening on
warning message		describes potential noncritical problems (optional)
failure reason		describes critical problems (only set if applicable)

Table 2.3: Structure of a tracker response message

The ID is randomly created for each download, thus one client can simultaneously have multiple IDs if it is downloading several files at once. Further the message provides information about how many bytes the corresponding peer already has downloaded, uploaded and how many are still missing. Due to possible transfer errors, the latter can not simply be calculated from the prior two. The last entry *event* indicates the purpose of this message. In the case of a new download—that we are regarding now—this would be a STARTED event. Just as the COMPLETED and STOPPED events, such a message is sent only once for each download. Obviously either a COMPLETED or a STOPPED event—which indicates that the peer aborted the corresponding download—is sent. Field code EMPTY is used for announce messages that are sent in regular intervals throughout the download phase in order to keep the tracker informed about the peers progress. Further they serve as a keep alive signal and peers not announcing within the expected intervals are considered as having failed and not being available any more. In consequence such a peer would be removed from the list of available peers that a tracker has for each file it is responsible for.

BitTorrent proceeding: The tracker response to a GetRequest

When a tracker receives a STARTED GetRequest it sends a *tracker response* message (table 2.3) to the requesting client. It provides the interval in which the tracker expects announce messages from the peer, the tracker's ID, the number peers that posses the complete file—the *seeders*— and of those who have just parts of the file—the *leechers*. Further a dictionary that contains information about peers that are in BitTorrent swarm for this file. The number of peers

Name	Description
bitfield	complete list of available and unavailable pieces
cancel	cancel a piece request
choke	prohibit download
have	inform that a certain piece has been completed
interested	announce demand for a piece
keep-alive	just avoid disconnect due to inactivity
not interested	no demand for the available pieces
piece	subset of a piece as payload
request	request for a specific piece
unchoke	permit download

Table 2.4: Peer wire protocol message types

in that list is limited by the tracker. It is argued that peers can not establish too many concurrent connections anyway—the version three of the official client stops connecting to new peers when it has more than 30 open connections—and thus do not need more than a certain amount. Commonly this does not exceed 50 entries, while often much smaller values between 20 and 30 are used. This limit is a trade off between the demand of the peers for connection partners and the tolerable size of tracker response messages.
In case of serious problems making the usual response impossible a failure reason is included in the response, while other fields remain unset (e.g. the peers dictionary). In case of noncritical problems that still allowed the request to be answered correctly, a warning message can be optionally included.

BitTorrent proceeding: Connecting to other peers in the overlay network

In case no errors occurred the peer receives the tracker's response and extracts the entries of the peers dictionary. Having this information it starts connecting to the provided peers. Each connection between two peers is commenced with a *handshake message* that introduces them to each other. In order to do so, they exchange information about the client and the protocol version they use as well as their ID and the file that is about to be exchanged. All further communication between peers is performed via *peer wire messages* (PWM). The header of a PWM contains the message ID that indicates the type and thus the purpose of this message. The official protocol version distinguishes ten different types which are listed in table 2.4. Immediately after the handshake, a message of type *bitfield* is exchanged that mutually informs the peers which pieces of the file the vis-a-vis already possesses. Peers store this bitfield for all connected nodes and thus maintain a knowledge base of the download status of all these peers. It is worth noting that this is usually only a proper subset of the

BitTorrent proceeding: Inter-peer communication complete swarm for this file, hence peers do not have a global knowledge about the Torrent. Based on the local information in the bitfield data a peer can determine whether the exchange partner can provide pieces that are still missing and send a PWM of type *interested* if applicable. An interested client is allowed to request—and eventually download—a piece after it has been *unchoked*. This requires each client to store for every open connection:

- its own interest it announced to the peer (am_interested)
- whether it is choked or unchoked by the peer (peer_choking)
- the interest status of the peer (peer_interested)
- whether it has choked or unchoked the peer (am_choking).

BitTorrent proceeding: Choking and unchoking of peers

After the handshake this status is initially set to *choked* and *not interested* for all peers. Within defined intervals the BitTorrent clients execute an choking algorithm in order to allow peers downloading pieces. The design of this algorithm is of significant importance for the protocol performance. A major aspect it must consider is avoiding too many downloaders at the same time. The motivation is that—due to the often relatively small upload bandwidth provided by consumer Internet connections—a large number of concurrent uploads would make each transfer very slow. This is amplified by underlying transport layer protocol TCP, whose flow and congestion control mechanisms experience a performance degradation under too many simultaneous transfers. Further the choking algorithm must ensure fairness which is maintained by BitTorrent's tit-for-tat policy that rewards peers who contribute a lot and penalizes selfish peers that do not share pieces to others. A widely used algorithm is described in the following.

A commonly used choking algorithm

Every ten seconds four peers with the highest upload rate are selected for being unchoked from those having the status *peer_interested*. In case they differ from those from the previous selection, the prior peers must be choked by sending the corresponding PWM. Since the newly unchoked peers are interested, they will start downloading immediately. Further there is the *optimistic unchoking* which is performed every 30 seconds. In this procedure a peer is selected for unchoking regardless of its upload rate. The probability of being selected is increased for newly connected peers, for all other peers it is equally distributed. The purpose of this modus operandi is to quickly provide new peers with complete pieces in order to allow an immediate start of piece exchange with other peers.

BitTorrent proceeding: The course of a file download

An unchoked client will—assuming that it is interested—immediately send a *request* PWM to the corresponding peer. This message not only specifies the piece that the client intends to receive but also the block within this piece. For a better performance usually several blocks are requested at once. The peer will respond with *piece* PWMs that contain the actual data of the requested blocks. After the client has completed the download of a specific piece it announces this event—by means of a *have* PWM—to all connected peers who will in turn update the appropriate bitfield. Thereby all peers keep the bitfields of all connected peers updated. However, this means that a lot of additional messages have to be sent which induce a notable amount of traffic. Thus a measure has been introduced in order to reduce the number of these have messages. This approach—which is not part of the official protocol—is referred to as *have message suppression* and prevents sending these messages to peers who already own the corresponding piece themselves. Obviously, these peers will not request that specific piece anymore and consequently do not that urgently need to be informed. However, this makes the identification of the rarest pieces inaccurate when the corresponding piece selection strategy is applied.

BitTorrent proceeding: Finishing a download

When a client has downloaded all pieces the download is complete. Afterwards it may leave the overly network or remain as a seeder. Seeders are valuable resources for a P2P swarm. However, BitTorrent offers little incentives for clients to seed after the download since the rewarding for uploads—represented by the tit-for-tat policy—is applied on a per-file basis only. Nevertheless, BitTorrent became and probably will remain one of the most popular and robust P2P protocols in existence.

CHAPTER 3

The Importance of Computer Simulations

The significance of simulations

In many areas simulations are of great importance, including product development, meteorology, finance market and especially for many research fields. Of special interest, in particular for this thesis, are computer simulations which offer a variety of advantages compared to real world testing and investigation of physical models. For the sake of simplicity the words simulation and computer simulation are from now on used synonymously. In the simplest case the benefit of simulations may just be a monetary gain since they can drastically reduce the costs for experiments. Even more important is the potential reduction or complete avoidance of risks, may it be for the involved actors or for the broader environment. In many cases a simulation is used as a cheap and safe alternative for real world testing, measurement and observation. However, there are also scenarios where it represents the only possibility to perform studies. This is the case if the corresponding real world processes are running too fast or too slow and thus observation and experiments are impossible. For example the analysis of forest vegetation over several generations of trees under different conditions; such as precipitation rate, average sun hours per day; would take hundreds of years for each simulation run in a real world test.

Benefits of simulative analysis

Simulation of notional systems

In the same way it is impossible to analyze systems that are not existing yet or can not be measured for other reasons. An example for the latter which is immediately related to this thesis, is the comparative testing of different parameters and protocols for software clients in a globally distributed system with thousands of participants. First it is doubtful whether a researcher would get into the position to measure data on such a high number of clients that are in private possession. Second, even if that was possible, coordinating the distribution and successful installation of the repeatedly changed client software respectively of the new parameters with such a high number of involved end devices is an error prone task. What makes it finally impossible is the fact that consistent measurements in such an environment can not be obtained, not least since the conditions are subject to continuous changes.

Porposes of simulations

In the application described above the simulation is used for the substantiation of a theoretical model. Other purposes they are used on include optimization (e.g. of processes) and prognosis of future developments.

The two steps of simulations

Regardless of the specific objective that is followed by a simulative analysis, it always involves two steps. The first step is the creation of a corresponding model that reflects the system and the second step the simulation execution. While both steps are important, the former is much more error prone. Thus the careful creation of an appropriate model is the key to any credible and reliable results. Two different approaches for modeling can be distinguished, the analytic and the descriptive models. The former rely on mathematical equations that specify the systems behavior which becomes increasingly difficult as the complexity of the system increases. The descriptive models implement a virtual copy of the real system that allows detailed analysis and measurement. Both approaches rely on an abstraction of the real processes. On the one hand describing every little detail in the model is not possible, on the other hand the more fine grained the model, the higher the computing power needed for the simulation. Thus a model must be as simple as possible and as complex as needed in order to achieve a trade-off between exactness and resource requirements for the simulation. However, from every abstraction arises a discrepancy between reality and simulation. This must be carefully considered in order to preserve the credibility of the corresponding results.

Analytic vs. descriptive simulation models

Limitations of simulative approaches

It must be admitted that simulations underlie certain limitations. One important aspect, the limitation of resources, has already been mentioned. It includes CPU power and memory as well as time. Further results are always only valid for the applied set of input parameters. Beyond that for reliable results all relevant influencing factors must be considered during the model creation process. Any aspect that is not included— may it be because they were forgotten, considered to be irrelevant or even not known at that time—might lead to completely different results in reality and in the simulation. On the other hand, the higher the number of parameters, the more challenging is the task of adjusting them correctly. Further this increases the complexity of the model and thus also the risk of errors. By carefully choosing the required parameters and by using reasonable abstractions, these potential problems can be managed.

In spite of these challenges that adhere to all forms of simulations, they remain an important tool for developers and scientists which is capable of significantly speeding up observation, evaluation and optimization of new approaches as well as of existing systems.

Concluding the survey on simulations

3.1 Models for Complex Networks

Simulating and modeling of social network structures are prevalent requirements for many research fields, especially for interdisciplinary research that often takes social networks into consideration. Also for the evaluation of the models and protocols proposed in this thesis simulations that incorporate social network structures are required. Thus in the following three of the most important models for graphs that reflect the same properties as social networks are introduced.

Models for social network graphs

3.1.1 Barabasi-Albert Model

This model produces small world graphs, thus such exhibiting short average paths. The key strategy of the Barabasi-Albert model (BA) [15] is preferential attachment. Starting from an initial graph consisting of a very small number of nodes ($m_0 \geq 2$), new nodes are added subsequently with a specified number of edges (m). The probability for connection to an existing node N is proportional to the degree of that target node. This leads to scale-free graphs that exhibit the typical power law degree distribution with few super hubs that have an extraordinary high number of connections and a high number of nodes with only very few links. The clustering in Barabasi-Albert graphs has been reported to decrease with growing numbers of nodes in [40]. Despite that the BA model is one of the best known and most widely used models for scale free networks.

Creation and properties of Barabasi-Albert graphs

3.1.2 Toivonen Model

The Toivonen model has been presented in [103]. It has been designed to deliver graphs that exhibit a high degree of similarity to those that can be observed in social networks. Correspondingly the TO model is supposed to deliver graph containing communities with very dense internal connections. Further short average paths, high average clustering coefficient and a node degree distribution that exhibits a steep tail. In addition the resulting network should be assortative. In order to achieve this, the TO model starts with an initial connected graph of size N_0 where new nodes are added by first connecting to $m_r \geq 1$ randomly chosen vertices. In a next step the new node connects to averagely $m_s \geq 0$ neighbors of each those nodes it connected to in the first step.

Creation and properties of Toivonen graphs

According to the results in [103], this model achieves its goal of delivering a realistic social network structure that exhibits small world properties, a power law degree distribution, short average paths and a high average clustering. Thereby, it prohibits the emergence of nodes with extremely high degrees—super hubs—that can be found in other models, e.g. in BA graphs. These highly connected hubs are usually institutions rather than individuals. Excluding these institutional sources might be interesting for certain P2P scenarios.

3.1.3 Watts-Strogatz Model

Creation and properties of Watts-Strogatz graphs

Another model for the generation of small world graphs was developed by Duncan J. Watts and Steven H. Strogatz and is described in [108]. It relies on rewiring of edges in a regular ring lattice with n vertices and k edges per vertex. Each edge in the graph is considered for rewiring with a probability p, also often named β in accordance to D.J. Watts naming used in [109]. The rewiring maintains the source node while randomly choosing a new target node from the set of all nodes. Thereby self-loops as well as parallel edges are avoided. Setting $p = 0$ leaves the regular ring lattice unchanged while $p = 1$ results in a random graph. According to the results published in [108] p should not be smaller than 0.3 and not greater than 0.5 in order to achieve reasonable small world properties. In addition to the short average paths these graphs further exhibit a high clustering.

CHAPTER 4

A Model for Content Distribution in Hybrid Networks

Highlighting the demand for a better model

The challenges the Internet is opposed to—being continuously pushed towards its capacity bounds—have been motivated already in the introduction of this work. Further the importance of efficient content distribution has been outlined in chapter 2 in the context of CDNs. In summary it can be stated that the Internet is confronted with steadily increasing number of hosts and traffic volumes. Not only does this trigger demand for infrastructural upgrades, the closer the network operates at its limit the more do delays increase and the higher the risk of packet loss. While current CDNs are able to significantly improve content delivery, they are just deferring the point where the network will become unable to handle the load and at which they might even collapse in the worst case. As this indicates a new content distribution model is vitally important in order to cope with the future Internet traffic. Even considering the current state and the near future, having a way to transfer online data to the users more efficiently—with respect to the Internet transfer volumes—might reduce costs, distribution time as well as power consumption and might avoid the necessity for infrastructural upgrades. There might be multiple ways to reach this objective, such as improving current CDNs while additionally concentrating capacity upgrades on the last miles, the edges of the Internet or to improve IP multicasts. However, in this thesis another approach is favored that uses a novel content distribution model which relies on a hybrid network infrastructure, a combination of terrestrial Internet and satellites. This architecture is described in detail in section 4.1. Having this network structure defined is the provision for the content distribution model that incorporates social network structures, user preferences and P2P technologies in order to reach the defined target. This

Problems of existing solutions

model is introduced in section 4.2.

4.1 Hybrid Networks

What is a hybrid network?

During the introduction we already had a glimpse into the understanding of hybrid networks that is being used in this work. This might be different from some also common associations with that term. Often it is used in the field of mobile telephone networks when the possibility for roaming between the phone network (e.g. GSM) and another wireless network (mostly wireless LAN) is meant.
Another very popular interpretation is the combination of an ad hoc network with an access point in the field of wireless networks [77]. The term can further be found in the context of networking. There it refers to the combination of two or more distinct topologies that are combined, e.g. several start networks interconnected by means of a ring, which is correctly called a multiplex network. According to this definition the Internet itself is a hybrid network, since it consists of a plethora of heterogeneous networks as it has been highlighted in chapter 2.2.

Examples for existing interpretations of hybrid networks

A plethora of further interpretations can be found in the WWW. Two further examples are given in the following, for which the network topology is not decisive. The first provided by [90] *"A hybrid network refers to any computer network that contains two or more different communications standards. In this case, the hybrid network uses both Ethernet (802.3) and Wi-Fi (802.11 a/b/g) standards. A hybrid network relies on special hybrid routers, hubs and switches to connect both wired and wireless computers and other network-enabled devices."* focuses on the combination of protocols.

Hybrid as combination of guided and unguided media

In [78] it can be read *"A hybrid network is a local area network (LAN) containing a mix of both wired and wireless client devices. In home networks, wired computers and other devices generally connect with Ethernet cables, while wireless devices normally use WiFi technology."* which uses the transfer medium as the distinctive aspect. It can be considered as describing a subset of the first.
A completely different approach is taken in the following definition published at [44] that puts its focus on the quality rather that than on the technological aspects. *"In networking terminology, a hybrid network–also called a hybrid network topology–combines the best features of two or more different networks". According to 'Information Technology Control and Audit,' hybrid topologies are reliable and versatile. They provide a large number of connections and data transmission paths to users."*

A definition of hybrid networks for this work

While none of the definitions is contradictory to an intuitive understanding of the term hybrid network, their multiplicity is confusing and thus handicaps discussions about this topic. However, for the scope of this thesis we need a clear definition in order to reach a common understanding of hybrid networks. This will enable comprehension of applications that build on top of this very special kind of networks that will be needed during the remainder of this work. Thus, for the scope of this thesis we use the following definition that corresponds to the description of hybrid networks given in [64]:

Definition 1 *A computer network is a **Hybrid Network** if it satisfies the the following rules:*

1. *It is a **combination of at least two networks**.*
2. *At least one of the involved networks must use **unicast** as its primary addressing scheme.*
3. *At least one of the involved networks uses a primary addressing scheme different from unicast, such as **broadcast**, **multicast** or **geocast**.*
4. *For every data packet that is send from a source to a target the routing decision can be freely taken for any of the involved networks without preconditions.*

*Thereby the **primary addressing scheme** of a network is that which is best suited for the corresponding Infrastructure and can be used in an efficient way.*

A direct consequence of definition 1 is that all nodes must be equipped with the hardware required to connect to at least two of the involved networks that satisfy the hybrid network conditions.

Distingtion between the Internet and a hybrid network

At this point a problem arises that has been mentioned before: How does the Internet differentiate from a hybrid network? Strictly speaking whenever we combine the Internet with any other network that uses the Internet protocol stack, it would just become part of the Internet according to its definition of being a network of networks. Let us imagine mobile Internet users who are connected via a GSM network. For sending data to a host, the latter technically uses a broadcast within the GSM cell the host is located, even though only the one destination device will process the data. From a wider perspective, considering the whole GSM operator network, it is rather a geocast. But from a global perspective considering the whole path, from an arbitrary Internet host to this mobile node, this is just a one-to-one communication, in other words: a unicast.

Consequently it is constituted that the Internet's primary addressing scheme is unicast. Although multicast and broadcast are supported by the IP, they are of little practical relevance. A global Internet broadcast is hardly ever reasonable and for multicast the overhead of group management is too expensive in the majority of cases, especially when the group memberships change frequently.

Constituting satellites for utilized broadcast network

For the remainder of this work we will consider only one specific type of a hybrid network that is composed of the Internet and a satellite network. From now on the term hybrid network will be used interchangeably for this specific type. Since satellites use broadcast as primary addressing scheme, all packets that are routed to the satellite uplink will be received by all nodes connected to the hybrid network. A more fine grained addressing is under the responsibility of the application layer protocol. Further we assume the utilization of geostationary satellites, radio frequencies in the K_u band and a bandwidth of 36 MHz. This allows the use of television satellites and has the benefit that equipment for reception of these signals is available at low cost and further is already installed at many households. This keeps entry barriers for implementing hybrid networks based on this technology low.

The home gateway as mediator between hybrid network and user devices

Connectivity to the hybrid network is achieved by means of a *home gateway* (HGW). This device was first described in [69] and further specified in [64]. It possesses external connectivity to the Internet as well as to satellite broadcasts. On the internal side it provides LAN connectivity, for the end user market this will usually be an Ethernet switch as well as a WiFi access point. This allows connection of the devices we find in typical households. The delivery channel, which is illustrated in figure 4.1, is thereby transparent to the devices connected to the HGW. A distinction, if required, can only be made on application layer. On the senders side no transparency is given. The decision whether or not a packet shall be broadcasted must be explicitly stated when it is sent. In general there are two possible models for realizing this procedure of hybrid network delivery.

ISP model A packet has a flag that indicates whether a broadcast is possible or not. In case it is it is further assumed that a broadcast is also desired. Every router the packet passes can now decide either to send it towards the satellite uplink station or to pass it on the next routers on the path to the recipients. This would give the satellite network operator the role of an ISP that operates on multiple levels. However, this approach requires updates for all routers since they must support the new protocol and further they must be able to handle groups of recipients. This is a significant effort and comes with the same problems that prevented the IP multicast from being widely used.

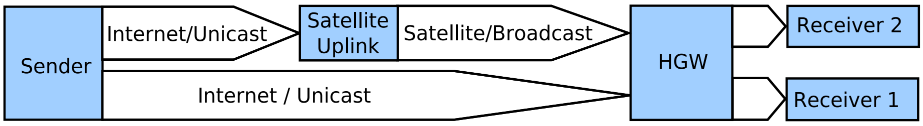

Figure 4.1: Data Channels in Hybrid Networks

CDN model In this model the satellite operator has the role of a CDN provider. Thus an agreement between sender and recipient must be made before the hybrid network can be used. When this precondition is met, the sender encapsulates a broadcast packet in a TCP frame and sets the destination address of the latter explicitly to that of the next satellite uplink station. This station will then extract the header information from broadcast packet and then puts the original TCP message in the broadcast queue accordingly. Preserving the TCP header is important for transparency on receivers side. In case congestion occurs the uplink rejects the packet and the sender will have to send it again via Internet in unicast mode. In order to allow most efficient usage of the broadcast network, a broadcast packet carries information on the number of recipients. This allows the uplink station to give a packet with 100,000 intended recipients the priority before one with only 1,000. Further a delivery deadline is provided by a broadcast packet that enables better scheduling of broadcasts. As a third header information a broadcast packet contains area codes that coarsely indicate in which regions the recipients are located, which allows the selection of the correct satellites respectively transponders.

While in the remote future the ISP model might become interesting, for the moment the CDN model is clearly the preferred one since it does not require any changes to the Internet infrastructure. Further it has only little overhead for the communication and thus can be realized on a short-term perspective.

Internet and Satellites, combination of opposed strengths

Why has this specific combination of Internet and satellites been chosen? The decision for the Internet as the unicast network is obvious since the defined target of this research is making online content delivery more efficient, combined with a reduction of Internet traffic. However, for the broadcast network also other solutions would have been possible such as cable TV, cellular or terrestrial radio networks. However, there are a couple of advantages that make satellites the preferred choice among all broadcast networks. Concisely it can be said that satellites and the Internet possess nearly orthogonal specialties. The Internet provides low latencies and a fast back channel. While these are points where satellites deliver a relatively poor performance, they come with notable strengths which can be derived from the introduction to satellite networks in chapter 2.3. In summary and with respect to the hybrid network approach these are the following:

Scalability Even though many other networks support broadcasts as primary addressing scheme, none of them can that easily adopt new user as satellites do. All those networks that rely on guided media as a physical medium demand for a cable being drawn from an (probably) existing hub to the new user. While unguided media such as WiFi, 3G or LTE obviously do not need any cables, they are limited in the number of users that can be concurrently connected to one access point. Thus also here a demand for additional infrastructure is raised as the number of user increases. Further, these wireless networks have a much more limited geographical reach than satellites. For the satellites it can be stated that within their area of reach they scale without limits with the number of users.

Coverage As already highlighted in chapter 2, satellites provide a great coverage. For the utilized GEO satellites, only three satellites are needed in order to achieve a global coverage. For guided media as well as for other unguided media broadcast networks—which have a much more limited reach—the cost for extending them towards global coverage makes this approach economically unreasonable.

Bandwidth Compared to many guided media networks, satellites respectively satellite transponders, do not provide a notable high bandwidth. Thus this aspect mainly applies for the comparison to other wireless technologies. For them the signal strength significantly depends on the distance to the next access point and thus the transmission rate degrades as distance grows. For satellite broadcasts the geographical location of the recipient does not notably effect the transmission rate as long as being within the footprint. There is no doubt that radio signals coming from an earth orbit underlie the same physical laws as terrestrial ones. However, the distance for GEO satellites is large in any case. Thus the relative effect on the distance to the signal source is negligible.

Simplifying the notion of a global satellite broadcas

As highlighted in section 2.3, one satellite can only cover a part of the worlds surface but for a complete coverage at least three satellites are required. For the sake of reduced complexity we will assume a full global coverage when the phrase *one broadcast* is being used, knowing that this refers to the event of three simultaneous broadcasts. In fact the results presented in [94], [22] and also in [12] show that content popularity tends to be rather geographically localized and thus in the majority of cases one broadcast using a single satellite is supposed to be sufficient.

4.2 The content distribution model

Expectations of this new model

In the preceding section we learned about hybrid networks. As announced at the beginning of this chapter we will now use this infrastructure to develop a model for scalable content distribution. Before we start let us quickly recall the expected benefits of that model.

1. Reduced Internet traffic, especially on Internet backbone.

2. Scalability with respect to the number of users and files.

3. Better or equal download duration.

Strategy: broadcasting popular files

The approach for reaching these objectives is based on the principle of broadcasting popular content that is frequently requested and unicasting the remaining, seldom requested content. Therefore it utilizes a hybrid network as underlying infrastructure. Since the broadcast allows every user connected to the hybrid network to receive and store the corresponding data, we assume a file level encryption that allows the free distribution of all content without violating copyrights. Every node in the network may receive, store and share all data without any restrictions. However, access to the content in terms of consuming it (e.g. watching a video, reading an ebook or running a software) is only granted after being authorized by the content rights owner. This encryption is not subject of this thesis and will not be further discussed.
Even though it is based on a hybrid network infrastructure, the content distribution model also supports mixed environments containing clients that have no connection to a broadcast network but only to the Internet or vice versa using a two-way satellite connection. Even though such nodes miss some of the beneficial aspects of the model, they still take profit from it as we will see during the reading of this chapter.

An abstract view on the content delivery chain

Before we go into the details of the model, let us inspect the process of content distribution in general. On one side there is a source for online content, the so called *content provider* (CP), on the other side a node requesting that content, the *consumer* or *user*. We find this situation illustrated in figure 4.2.

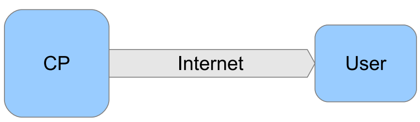

Figure 4.2: Usual delivery path for online content.

As we learned from chapter 2 there are usually numerous hops on the path between these two nodes that cause delays and bear the risk of potential package loss. For the presentation of the proposed content delivery model the abstract model in figure 4.2 is used for the sake of a reduced complexity, keeping in mind the potential problems of this delivery channel.

The delivery chain for hybrid networks

In case of a hybrid network an alternative path is added to this delivery chain that allows the CP either to send the data to the requesting users directly via Internet or to pass it to the satellite uplink station (SUS) for having it broadcasted. No matter which delivery channel is being used, the data is first received by the HGW of the requesting user, that will pass it on to the corresponding user device the content request originated from. This leads to the delivery chain visualized in figure 4.3.

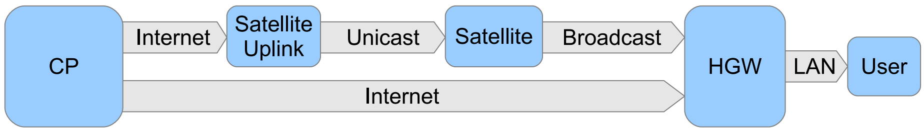

Figure 4.3: Delivery chain for online content in hybrid networks.

So far this is what one expects when a hybrid network infrastructure is being used. On top of that, the content distribution model adds functions at three points of this delivery chain. These are the CP, the SUS and the HGW. It is worth noting that in principle instead of the HGW the corresponding functionality could be implemented on any other device, may it be a dedicated hardware or an end user device such as a PC, laptop of mobile phone. Offering a corresponding software package that can be installed on various devices might be an interesting solution especially for user without access to a hybrid network, who do not necessarily require a HGW. However, for the scope of this work it is assumed that all users have a HGW installed that also provides the necessary functionality even if no hybrid network access is provided. Besides this saves us from having to differentiate at each point the HGW is mentioned, having this device in place as mediator between the home network on the one side and the Internet and/or broadcast network on the other is the preferential choice since it can provide all devices in the home network with the benefits of the proposed content distribution model. Further, the HGW is considered as being an always-on device—similar to the common case for most of todays Internet Modems and DSL routers—and thus can operate uninterruptedly.

The HGW implements all client side functionality

Now let us examine the features that the different nodes implement. The CP has to monitor the active downloads respectively requests for all its files and must decide accordingly on what data should be broadcasted instead of being unicasted. The SUS has to check how much satellite capacity is still available and corresponding to the result instruct the CPs to reduce their broadcast requests or to increase them. The biggest part of functionality is implemented

Summarizing the functions of the main actors in the delivery chain at the HGW. Besides receiving data and passing it to the devices in the corresponding home network, it also actively requests files according to the users preferences. The intention behind this **prefetching** is to download files even before the user's demand arises and thus have them already available in the local cache when they are needed. The prediction of future demand that is required in this context is probably the most challenging part of the whole content distribution model. In order to facilitate this, in addition to the user's preferences also data provided by his social network neighborhood in being capitalized. The details of these aspects will be discussed in section 4.2.3.

Demand prediction and prefetching

In order to be capable to do the described caching, obviously the HGW must be equipped with a sufficiently large storage. Besides prefetched files, also those that have already been requested and consumed by a user remain in the cache. Which files are cached depends on the cache replacement strategy and on the rating for future demand probability. Both aspects are described in detail in section 4.2.3.

As an immediate result of this caching policy there will be a lot of files being cached at the very edges of the network. It is more or less self-evident to make use of this situation and to serve future requests from other users in physical proximity from these caches. Since this concerns data that has already been delivered once before from the CP to the HGW's cache this process is referred to as **re-distribution** of content.

Delivery based on P2P technology

For this re-distribution among HGWs as well as for the delivery from CP to the HGWs a P2P technique is being used. More precisely a modification of the popular BitTorrent protocol—described in detail in chapter 2—has been selected, primarily because of its utilization of a tracker. As we know from the previous chapter, every peer first contacts the tracker when a new download is started. Further all peers periodically send announce messages that keep the tracker informed about their download progress. Thus the tracker has profound knowledge about the number of concurrent downloads of each file as well as the individual progress. This is precisely that information the CP needs in order to decide whether to broadcast specific data or not. We will come back to the connection between tracker and CP as well as to the details of broadcast decisions in section 4.2.1.

On the eligibility of a P2P approach

Besides the mentioned benefits that come with the trackers, a P2P approach is further the natural choice for the re-distribution of cached content. The involved nodes have usually a rather low Internet connection bandwidth and also a low reliability, both properties that harmonize very well with P2P. In the style of the BitTorrent protocol each file is partitioned in several smaller **pieces**, in other P2P protocols often referred to as *chunks*. According to that scheme from

now on the granularity on which data is either broadcasted or unicasted is set to one piece. Correspondingly the content provider decides for a piece whether it should be broadcasted or not, and the HGW either receives a piece via broadcast or via unicast network.
After this coarse introduction to the proposed content distribution model still several questions remain open. Probably the most noteworthy ones are the following:

Q1: How does the CP know which nodes can receive broadcasts in a mixed environment?

Q2: How can nodes determine which potential P2P exchange partners are nearby and which are far away?

Q3: How can one broadcast be efficiently used to substitute unicast transfers when the content requests are individual and thus issued at arbitrary, most likely different times?

Q4: How can the future demand be predicted in order to facilitate the prefetching?

Confining the model to registered users

The key, even though not the answer, to these questions is the utilization of user profiles and restricting the whole service to registered users. In order to realize this, each user has a profile that contains the required information. In principle also multiple profiles per user can be used, allowing to keep e.g. private use separated from professional use. These profiles are stored and maintained on the HGW and thus remain under the full control and disposal of the corresponding users. However, storing and using personal user information always raises security and privacy implications that have to be addressed. For the scope of this work and the corresponding research we assume that the users can freely decide whether they want to use the proposed service and what data they want to share and which is strictly private.

Constitution of user profiles

The user profiles contain data about personal preferences—what games one plays, what software is installed and what kind of movies; books or music one likes—whether or not the HGW is able to receive satellite broadcasts and the physical location of the HGW. For the latter a very coarse granularity can be used. For instance in the implementation of this content distribution model that will be discussed in chapters 7 and 8 just a number that identifies the autonomous system the HGW belongs to is being used for this purpose. This is in most cases sufficient to determine whether two users are located within the same ISP network, and thus can be used to keep inter ISP traffic as low as possible by giving preference for communication and data exchange to nodes within the same AS. Whether or not the profile also contains information about the user's social network or if rather data from accounts in existing OSNs is being used has no impact on the model in general. Thus it is just assumed that the social graphs are accessible for the HGW for all users it administrates.

Naming conventions: sat-enabled, sat-peer and SatTorrent

Before describing the functional details of CP, SUS and HGW two more naming conventions are introduced. From now on we will refer to the capability of a node respectively of an HGW to receive satellite broadcasts as being ***sat-enabled*** and to the corresponding nodes as ***sat-peers***. Further the modification of the BitTorrent protocol that will be presented in the subsequent chapters is named ***SatTorrent*** due to its special purpose of integrating support for satellite broadcasts into BitTorrent. For the moment it is sufficient to know that SatTorrent provides all features of BitTorrent, the remainder will be introduced in the chapters 7 and 8.

4.2.1 Content Provider Logic

New tasks for content providers

Obviously the most important task of a CP is to host its content and provide access to it. However, the proposed content delivery model adds further requirements that are essential for the functioning of the model. As indicated in the brief introduction above, for the general provision and distribution of files a BitTorrent like P2P approach is utilized. Correspondingly the CP's content provision is realized as a seeder that uses the same client software that is being used for all other peers. In contrast to them, this original file source—referred to as origin-seeder—will have a much higher reliability and Internet connection bandwidth in order to serve several concurrent requests up to a certain number via unicast. However, as results presented in [26] and [43] show, the bandwidth requirements for the content provider as well as its overall outgoing traffic is significantly reduced—by more than 65%—in consequence of the P2P technology. Since the proposed SatTorrent protocol is based on BitTorrent, it further relies on a tracker, as already indicated before.

Relation between tracker and content provider

The tracker might be located on an arbitrary node in the Internet and even be provided by another authority. However, there are good reasons to have the tracker maintained by the content provider and in physical proximity to the hosted files as will become obvious during the reading of this section. For now we will assume that tracker and content provider are one entity and that the tracker can access the files directly. Otherwise for an optimal performance it would be required to have a minimal throughput on the Internet path between content provider and tracker that conforms to the bandwidth of the satellite transmission. The torrent files can be hosted at arbitrary locations, however the corresponding servers should have a high reliability. In principle the access statistics of torrent files could be used as an early warning system for future downloaders, yet the precision is questionable since the timely correlation between torrent and file download is uncertain plus the torrent files might further be exchanged directly between users, potentially via OSNs.

The two-tier network approach

Summing up the specifications from the preceding paragraph we end up with a two-tier network architecture consisting of a highly reliable, always available backbone—composed of the origin-seeders, the trackers and the torrent file servers—and the consumers who download the files. The latter are assumed to commonly have small upload and download capacities as well as a low reliability. The benefits of such a network layout, most notable the reliable accessibility of persistent data, have been discussed in [19].

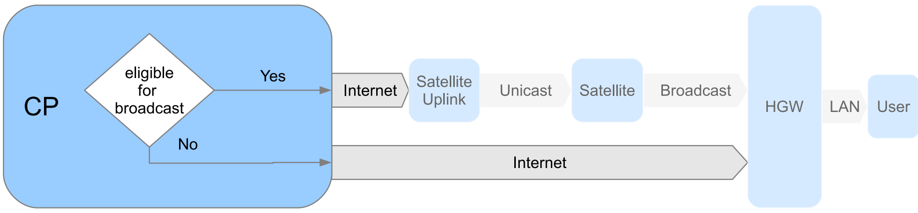

Figure 4.4: The content provider's broadcast decision.

The concept of broadcastability

As figure 4.4 indicates the CP has to decide which content respectively which pieces shall be broadcasted. First there is the fundamental question whether the nature of the content allows a broadcast. For example everything that exhibits a high degree of individuality—e.g. the content of a dynamic website—is probably not suitable for a broadcast. Further the delay induced by the long distance to the satellite and back might not be tolerable for certain types of content. We refer to the general applicability of content for being broadcasted as **broadcastability**. While for certain cases the decision against a broadcast is obvious, for the opposite direction this is much more complicated. Regarding currently available technology, a broadcast comes with much higher costs than one Internet unicast of the same amount of data. Obviously a certain number of concurrent recipients is needed for an economically reasonable broadcast. This value, that we refer to as the **broadcast threshold** (BT), depends on many factors—e.g. costs for satellite construction, satellite launch and satellite maintenance, station keeping but also for energy and many more—and may vary a lot. As a result it can best be determined by the satellite operator respectively the SUS. From a solely economical point of view one might argue that the SUS, as a service provider for the CP, simply charges his customer corresponding to the traffic and thus the CP would decide on the cost comparison. However, when the objective is the most efficient usage of the satellite, letting the SUS determine the BT is the preferential solution.

The broadcast threshold restricts access to the satellite uplink

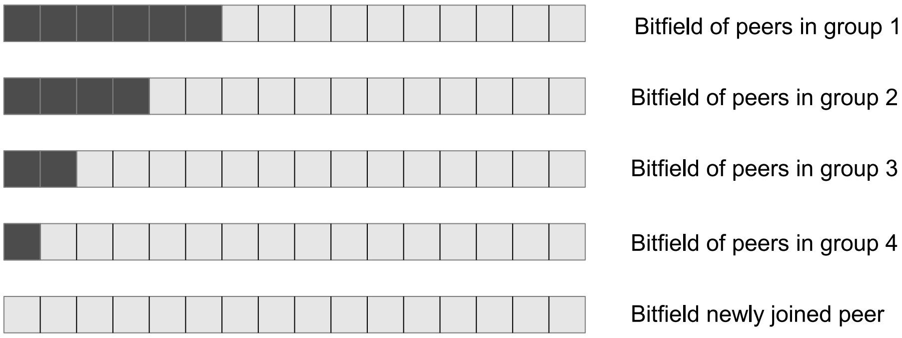

Figure 4.5: Exemplary client bitfields before a broadcast.

Dealing with delays in broadcasts

According to this BT is assumed to be a given parameter for the CP, who in turn has to determine which pieces have a number of concurrent requests that exceeds BT. While this is possible, waiting for piece requests to arrive, then making a broadcast decision and triggering the broadcast if applicable adds a significant delay for each piece. However, each piece belongs to a certain file, and it can be assumed that a user who requests a piece will continue to download the complete file. Thus when the sum of piece requests for one file from distinct users is greater than BT a broadcast of all pieces of the corresponding file might be considered. However, pieces are often requested out of order, and several users potentially already possess several of these pieces. Let us consider the download of a movie and the CP being a video on-demand provider. In a realistic setting each user would start watching—and thus downloading—the file at different moments in time, one after the other. Thus when the number of downloaders exceeds BT, all those that started watching before will have partially finished the download already. Accordingly only those pieces that have not been received by any user satisfy the requirements for a broadcast. Luckily, the tracker—which, as we recall, is an essential part of SatTorrent—has knowledge about the number of concurrent downloads by definition. Beyond that it also has detailed information about the download progress, by means of already downloaded pieces, for each user. This makes the tracker the perfect authority to decide about and to trigger broadcasts. If now, as it was recommended before, the tracker is hosted by the CP and has direct access to the files, it can immediately send the applicable pieces to the SUS for broadcast.

Determining global piece demand of sat-enabled downloaders

Exemplary download progress situation

An exemplary situation for a video on-demand movie download is illustrated in figure 4.5. A specific characteristic of such video streaming via P2P is the in-order piece download since pieces next to the current viewing position must be available first. Let us assume four groups of peers with a homogeneous download progress within the groups. We find these groups in figure 4.5 and the downloaded pieces represented by dark squares. Let the sum of all peers of

Download progress after the broadcast

these groups be $BT - 1$, thus the new joining peer will cause BT to be reached. Now the tracker determines those pieces that are still missing for D peers, with $D \geq BT$ and triggers their broadcast. This leads to the situation illustrated in figure 4.6 where all peers possess the broadcasted pieces, shown as green squares. In the meantime all peers continued to download pieces via P2P exchange, as a result they thus received the pieces marked as red squares. It can be observed that group 1 has obtained one piece twofold, once via broadcast and once via P2P. It is true that the tracker might have considered this circumstance and excluded this piece from the broadcast. However, the tracker has no guarantee that all peers will receive that piece via unicast. Therefore this is a design decision towards increased reliability which can be taken differently for each implementation of the model.

Service side estimation of future demand

Besides such dynamic piece broadcasts that are based on monitoring the current download situation there are other situations where the CP knows in advance how many users will download a file. This might be in case users made a subscription for a certain type of content, pre-ordered a certain file or even if a high popularity can be expected in advance. The latter might be the case for videos of certain sports events, the news or releases of patches; updates or addons of popular software. Further the results presented in [99] indicate that predictions on the future popularity of certain types of online content are possible after two to three days after the corresponding file has been made publicly available. This shows that a service-side prediction of future file demand is possible. However, this only applies for certain categories of content, in the case of [99] it was shown for videos shared via YouTube. Further these approaches only consider the pure number of expected requests and no geographical distinctions. This and the fact that the user clients may have access to much more information that allow better predictions without raising security and privacy issues, a client side prediction has been preferred for this content distribution

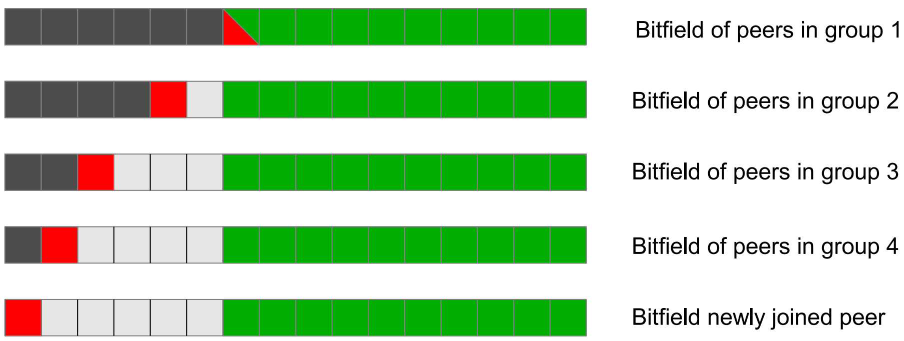

Figure 4.6: Exemplary client bitfields after a broadcast.

model. The details of this approach are discussed in 4.2.3.
It is worth noting that in general for the broadcast decision only sat-enabled downloaders should be considered since only these will directly benefit. However, it might sometimes be desirable to make a broadcast even if the number of sat-peers does not exceed the BT but at the same time the total number of downloaders is extremely large. In such a situation the broadcast would immediately increase the number of available seeders and thus speed up the distribution as well as reduce the load on the SP. Further, this can potentially reduce the Internet backbone load due to the location awareness implemented by SatTorrent.

Peers in consideration for estimating broadcast efficiency

Whether there are any files in existence that exhibit a sufficiently large number of concurrent downloads that exceeds the BT is to be clarified. However, the results of [33], [14] and [9] suggest that this should be the case. This assumption will be further substantiated by real world measurements presented in chapter 5. The total number of requests for a certain file offered online is often very easy to determine. For example in case of YouTube the absolute number of views for a certain video is immediately shown on the website. However, far more important that this value is the distribution of requests over time. In case they should be are equally distributed, it will be highly unlikely that a broadcast ever was economically reasonable. Fortunately there is evidence that for much online content there are often peaks of requests where numerous downloads occur in a relatively short period of time [39], [38]. In order to show more evidence for this temporal correlation of content requests the access patterns for randomly chosen and also for consciously arranged sets of videos have been observed and analyzed. This study and its results are presented in chapter 5.

Correlation of file downloads

4.2.2 Satellite Uplink Station Logic

The SUS has a precise knowledge about the number of satellites and the corresponding capacity that is has at its disposal for the hybrid network respectively for the content distribution. It can be expected that the demand for this capacity follows the typical daytime curve that has been discussed during the introduction of this work. Thus potentially there will be peaks of high load and more quiet time, especially at night. However, this curve reflects mainly the times when users have time to make downloads and not necessarily when they actually need it. This circumstance is exploited by means of scheduling the broadcasts according to their urgency and shifting low priority requests into nighttime. However, the SUS itself has no means of estimating the urgency of a specific piece. In section 4.1 we introduced the header information provided for each broadcast request on a packet level. For the content distribution model we will consider a more abstract form on a per piece level. Accordingly we assume the presence of the following information for every piece that is sent to the SUS:

The role of the satellite upling station

Number of recipients and the broadcast deadline. In case the latter is not set, the corresponding piece has highest priority and must be sent immediately. All other pieces can be sent at any instant of time before the deadline.

Handling congestion at the SUS

In case for a certain time interval more pieces are in the broadcast queue than satellite capacity is available the SUS must take measures to address this problem. First it must announce this incidence to all the piece sending CPs and instruct them to increase their BT values (see figure 4.7). While this might

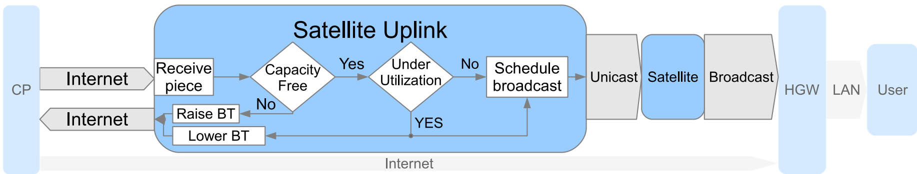

Figure 4.7: The satellite uplink station's decision chain.

avoid future problems, the actual issue must still be resolved. Some of the corresponding packages might just be deferred to the next time interval. However, depending on the content the corresponding delay might not be acceptable. Further, what if in the consecutively time slots also congestion occurs? The only way out would then be to drop the pieces which forces the intended receivers to request them again after a timeout occurred. However, considering the delay that is inherent to satellite broadcast[1] the packets might still be received in time when the SUS informs the CP about the congestion immediately and instructs it to use the unicast network instead. Doubtlessly, it would save time if the SUS addressed the pieces via unicast to the corresponding recipients directly. But knowing their addresses in advance would require the CP to send them for all recipients with every piece that is meant to be broadcasted. This would be a tremendous overhead, not to mention the privacy issues raised by this practice.

Feedback channel between SUS and content provider

Non-critical is the situation when the number of pieces, respectively their total size, does not saturate the available satellite capacity since it does not affect the successful piece delivery. However it is desirable in terms of efficiency to strive for a full capacity utilization. Thus in such cases the SUS instructs the CPs to lower their BT in order to increase the number of pieces selected for broadcast.

Whether an implementation of this model takes the risk of delayed piece delivery or rather triggers unicast compensation messages is primary a design decision. Both approaches will work and in case the HGW and the CP perform well in their tasks of predicting future demand respectively setting deadlines, such situations should occur seldom.

[1] At least when geostationary satellites are used, as it is the case here.

4.2.3 Home Gateway Logic

Reliability and availability of HGWs

As described section 4.2.1 a two-tier approach is utilized in order to increase the overall system reliability and availability. Since a node's probability to fail also largely depends on the complexity of its software, the most logic including the user profile management is implemented in the HGW where minor outages are admissible or even part of normal operation. For example some users might prefer to switch off their HGW during the night—even though this is not recommended—or short times of unreachable HGWs might occur as consequence of the commonly used dynamic IP address assignment. Furthermore, besides reducing software complexity in the reliable network tier, this approach gives full control about their private data to the users. When a user deletes his profile or turns off the HGW, no access to it will be possible anymore.

Functional requirements for the home gateway

Before entering into the details of the HGW's functionality, we briefly summarize the requirements it must meet in order to be able to deliver this performance. These include an Internet and/or a satellite connection as well as a sufficiently large storage that represents the persistent cache. Further—for the home network connectivity—it must provide an Ethernet and/or WiFi connection, ideally in form of a multi-port switch respectively a WiFi access point. Further we assume the presence of a hardware that is capable of executing the required client software and an up-time of 24 hours per day. The HGW provides its functionality to one or more users and requires their profiles to exist and to be stored locally.

Integration of social network structures

Further several functions of the client software rely on knowledge about the immediate social neighborhood of the connected users. How this information is being provided does not affect the functioning of the model's implementation. It is possible to access one or more existing OSNs whenever the corresponding data is needed, it can be requested from there once and then being stored locally or a new social network structure can be implemented by the HGW. The latter might be the preferred solution since it strictly distinguishes between people users want to communicate with and those they want to collaborate with in the scope of the proposed model, which probably requires a higher level of trust between the social contacts. Furthermore it might be useful for the system performance to limit the social contacts to those who are really close, since for those ties the mutual influence and thus the similarity in personal preferences is considerably higher than for loose ties as stated in [18] and [24]. However, in principle it has no influence on the functioning of the client software where the social network information is stored. Thus for the scope of this work we only assume that the HGW has unrestricted access to this information whenever this is needed.

Starting a new download

After having identified these preconditions analyzing the functionality provided by the HGW can commence. We start its illustration with a user connected to the HGW making a request for a specific online content C. Thereby the HGW serves at a proxy for this request and first checks its local cache whether C is already present, as illustrated in figure 4.8. In case it is, the re-

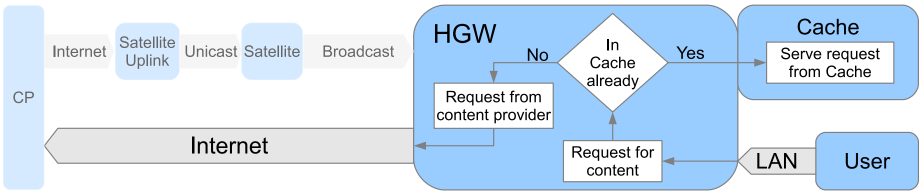

Figure 4.8: HWG handling of user requests.

Joining the P2P overlay network

quest is served from cache, otherwise it is forwarded to the content provider that corresponds to C. In the following the HGW will join the P2P swarm for C and download it, forwarding all received data to the user who made the request. Depending on the popularity of C the pieces will arrive via unicast or broadcast network, respectively a mixture of both (see section 4.2.1, broadcast threshold). The reception of pieces during the P2P file exchange is depicted in figure 4.9.

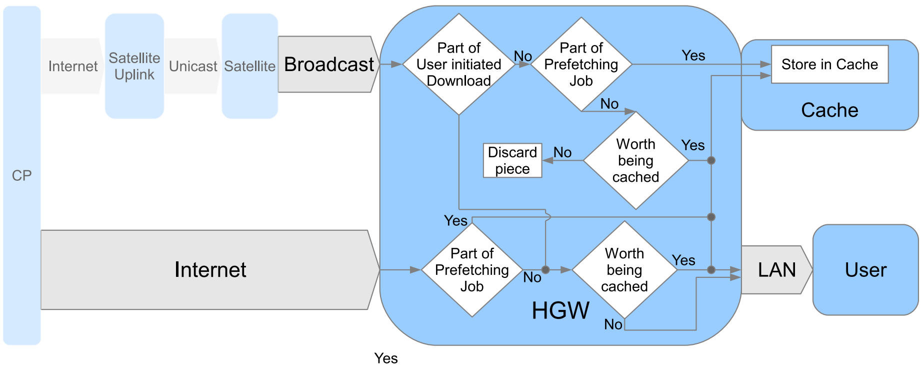

Figure 4.9: Piece reception at the HGW.

Handling pieces received via unicast network

Pieces that arrive via unicast network necessarily belong to a download that has been started by the HGW respectively by a user connected to it. In consequence the unicast pieces either stored in the local cache or forwarded to the user device the request originated from. In the latter case a copy of the piece might further be kept in the HGW's cache. The reasons for this behavior will be discussed later in this section.

Handling pieces received via broadcast network

The situation is different for pieces received via broadcast network. These might have been requested either by a connected user, by the HGW by means of a prefetching job or not at all. But even in the latter case it might be interesting to store the piece in the local cache. The decision whether or not this is the case is based on a metric that reflects the probability that a specific content will be needed in the future. This is decided on a per file basis rather than for every piece. This metric takes into account the preferences of all users registered at the corresponding HGW as well as their individual social network neighborhood. The reason for considering the latter is twofold. First, according to the observable mutual influence among social contacts as described in [18] and [55], a file that was accessed by a close friend might also be requested by the corresponding user in the near future. Second, when a file does not match the own preferences but those of a friend, it might be reasonable to cache it in order to support this friends potential future download of that file. This procedure is being referred to as **social caching** which together with the **personal caching**—which is responsible for the demand of local users—makes up the complete caching policy. Since storage capacity is a limited resource, priority is given to the personal demand of local users. As long as free space is available, everything that—according to the metric—reaches a certain minimum probability for future demand is cached, no matter if for personal or social use. When the capacity limit is reached, in an event of personal caching attempt first files from the social cache are removed. In case all cached files are for personal demand, those with the lowest probability are replaced. It is obvious that new social caching attempts can only replace files that have not been cached for personal demand. Depending on the total amount of available storage, the number of local users, their variety of interests and the number of files provided by content providers the portion of cache that is used for social caching might be low or even non existent. However, due to the similarity in taste observed for tight social ties still a significant number of files that were cached for personal demand should also be applicable for supporting social contacts.

Caching data for own demand and for friends

The traffic neutral approach of passive prefetching

In the course of this section prefetching has been addressed several times. The objective behind this technique is to have files stored in the local cache already when the connected users actively decide to download them. Prefetching has two dimensions. There is the **passive prefetching** where the HGW monitors incoming broadcasts and checks whether the received data belongs to a file that might be of interest in the future, that was already described above in the context of caching. As mentioned, an implementation of this content distribution model must provide a metric that allows rating of files for expected future relevance. In case the prefetching engine decides to cache a file it simply stores it in the local cache if there is space left. Otherwise cache replacement strategies must be applied in order to free cache space for the new data. However, this approach relies on broadcasts that have been triggered in response to other users requests. This might in many cases not be sufficient to have a maximum of files cached that might become subject of future user demand. This especially applies when the local users are interested in files that exhibit a very low popularity and thus are never broadcasted. In order to fill this gap and to increase the cache hits the HWG uses **active prefetching** that, as the name betrays, actively requests content that has been ascertained as being subject of future demand with a high probability. However this must not hinder any other, user initiated download and thus only occurs during idle times since it probably will allocate bandwidth on the unicast network.

Active prefetching demands for reliable demand prediction

While passive prefetching does not cause any additional network traffic, the active approach does with a varying amount, depending on whether the corresponding content is being broadcasted or not. Nevertheless, it is important to limit this traffic. Here the reliability of predictions on future demand play an important role, since when the prediction is correct, the user would have downloaded the corresponding file anyway, just at a later moment in time. In other words whenever we have a **cache hit**, and thus a **prediction hit**, for an actively cached file the traffic was not wasted. On the other side, files that are removed from cached in result of the cache replacement strategy without having even been accessed indicate a waste of resources. Obviously a good prediction engine is needed and only those files with a high expected reliability of predictions should be considered for active caching. It has been shown in [11] that future popularity of online media can be predicted with high reliability by means of harnessing social networks.

Prediction and caching is subject to client implementation

The implementation of this prediction is not subject of the content distribution model itself and thus not further discussed here. It may vary from one HGW to another and potentially even allow user customization. An example of a prediction engine will be discussed later in chapter 8 when an implementation of this content distribution model is presented. The same applies for details on the cache replacement strategies. For the moment it is assumed that both exist and deliver decent results.
In the following chapter on of the most popular online video portal provider, namely YouTube, is analyzed for determining the existence and predictability of exceptionally popular files and in particular for peakss in the request for online videos.

CHAPTER 5

Access patterns of online content

Challenges arising from intensified use of online services

The fact that many users frequently switch between a plethora of devices—tablets, PCs, laptop, smartphones—that all have an almost permanent access to the Internet completely changed the way how most people manage their data. It is easier and often also faster to store data online and download it on each device than using direct transfer options. Especially when files are to be shared with other people the utilization of online storage considerably simplifies the process. As a result the providers of online storage or specific service platforms experience a vast number of user requests as well as up- and downloads. As indicated in the preceding chapter whether or not a hybrid network respectively the proposed content delivery model is eligible to improve the distribution of files largely depends on two aspects. The number of users that have an interest in an individual file—referred to as the popularity of a file—and the timely correlation of the corresponding requests—referred to as the **peakiness** of access patterns. Both together determine the **broadcastabiltiy** of a specific file. In this chapter we will analyze access patterns for online content of a very popular content provider. Youtube[1], which is currently the worldwide best known and most used online video platform, experiences more than two billion requests for its videos every day [111]. Each minute 24 hours of new videos are uploaded [46] by professionals, companies or private users. The latter produce what is referred to as **user generated content** (UGC). This is the key aspect of the Web 2.0 and introduced a paradigm change in the Internet usage. According to [72], most files on YouTube are user generated content, but they seldom appear in the category of the most popular videos, which is dominated by professional

Peaks in access patterns and affect the broadcastability

[1] `www.youtube.com`

Distribution of video popularity limits the scope

content. Nevertheless, the most subscribed channels on YouTube are user channels. Due to the ZIPF–like distribution of user generated content popularity [4], a relatively small fraction of these videos causes a significant portion of traffic. While is it obvious that this small fraction of—according to [25] and [32]—about 10% is predestined for being broadcasted, identifying the corresponding videos early enough is difficult. This study will further narrow down the scope of videos that are potentially part of these very popular clips.

The importance of YouTube and its popularity as well as the existence of a sophisticated API were the most important criteria to choose this content provider for an analysis of file–request correlation. The aim of this study is to gain more knowledge about the peakiness of requests and thus to determine the broadcastability of existing material.

5.1 Data Collection

Aspects of the YouTube API

The decision on how to collect data is essential for the quality of the results. Technically the YouTube API allows requests for detailed information of a certain video, identified by te video ID. This information is updated by the provider in intervals between 30 and 120 minutes[2]. Further various feeds are provided that contain the IDs of videos that meet certain criteria, such as having a high popularity or been uploaded recently. The complete list of the available feeds is shown in table 5.1. All of them can be further filtered according to specified categories, such as News, Comedy, Entertainment, Nonprofits & Activism, Pets & Animals, Sports and many more. The feeds are updated in different intervals, varying from 30 minutes up to 24 hours. Further YouTube permits the creation of so called **channels**. These allow users to aggregate an arbitrary number of videos in a common group. For example all videos uploaded by a certain user can be found in his user channel. Further one might create further channels such as *vacation* or *research*.

Measurement procedure

In order to gain insight into the evolution of online video requests and their tomporal correlation, which is expected to enable identification of certain groups of videos that are particularly eligible for being broadcasted, numerous measurements have been performed. During each of them one or more of these feeds have been periodically checked and the contained videos have been added to the observation list in case they were not already embodied. Thus videos appearing in several feeds were not added twice. The number of videos per feed has been limited to one hundred. Besides their IDs also the uploading user, the corresponding location, the category, upload date and the duration of the video

[2]Source: `www.youtube.com`

Category	Description
Most popular (today)	Videos with the most views today.
Most popular (week)	Videos with most views during the actual week.
Most recent	Most recently added videos.
Most responded	The videos with the most responses.
Most shared	Videos most frequently shared on Facebook and Twitter.
Top rated	Most highly rated YouTube videos.
Top favorites	Videos most frequently flagged as favorite videos.
Most discussed	Videos that have received the most comments.
Recently featured	Videos recently featured on the YouTube home page or featured videos tab.
Trending videos	Videos from YouTube trends.

Table 5.1: Standard video feeds provided by YouTube.

have been recorded. All this is static data that does not change.

Periodical update queries for statistical data

For all files under surveillance the video information was fetched again in intervals between 30 and 120 minutes in accordance to the update windows of YouTube. Thereby the updated values of the entries that contain the total number of views, the IDs of related videos, the number of subscribers for the corresponding channel, the amount of comments, the user rating for that video and the time stamp of the update were recorded and appended to the video data.

Capturing the relevant phase in a video's lifetime

For some of the observed feeds it was expected in advance that the probability of catching a video that exhibits peaks in its access patterns was considerably low. This especially applies for the feed of most recently uploaded videos. Further videos appearing in the feeds most popular, may it be those of the day or of the week, already have a high view count. Whether or not this would further increase during the measurement was uncertain. Thus the obtained data misses the most interesting phase where the videos got popular. In order to have the guarantee that very popular videos are observed from the beginning additional video queries have been added to the list. This includes the videos in the user channels of the YouTube users *RayWilliamJohnson* and *BarelyPolitical*. Their channels are on top of the US *all time most subscribed* list. Ray William Johnson's videos reliably get at least 3 million views during the first 4 days after their publication. For BarelyPolitical, the situation is a little different. Most of his videos do not get that high view counts, usually they stay at values between 100,000 and 500,000. But there are several exceptions which have millions of views. Since the primary objective for monitoring these channels was to capture the transition from low to extremely high popularity, only videos not older than 30 days were considered. Further the search function of YouTube has been utilized to add videos to the observation list that relate to hot topics at the corresponding time.

The granularity of data updates

It is worth noting that the measurement interval—which is set to 30-120 minutes—is not as fine grained as it would be desirable. As mentioned before, this follows from the limitations of the data provided by YouTube. During early measurements requests for the video statistics have been done every 15 minutes. The results document that YouTube seldom provides data updates for the very important view counter in Intervals shorter than 60 minutes. This especially applies to very popular videos with heavy changes for these values. Thus the statistical data is coarse grained on any account. This causes the results to appear as stepped curves as we will see later in this chapter (e.g. in figure 5.3). Interestingly, the comment statistics of videos are updated much more frequently. Thus the data observed here could be used to approximate the views between the updates. However, this interpolation would have resulted in values that were not entirely authentic. Due to this reason this has not been done for this study.

5.2 Evaluation of results

Specifying the characteristics of peaks

As indicated in the beginning of this chapter the objective of this analysis is to find peaks in the request patterns of online videos. Thereby the frequency of peaks as well as their magnitude are important factors. However little can be said about both unless it has not been specified what a peak is. An important factor in this context is the number of requests R for a certain file within a specified time interval t of length T. Instead of R_{t_T} we use the short form R_t to refer to this value, using T as a static parameter. Let further be R^B be the peak defining threshold, thus $R_t \geq R^B$ indicates a peak.

Relevant peaks must exhibit high absolute view counts

In general it would make sense to set the value of R^B relative to the average over all R_t individually for each file. However, this kind of peaks are of limited use for our application. The objective here is to identify extraordinary high absolute values of R_t instead of relative ones. We recall from chapter 4 that a broadcast threshold (BT) must be reached in order to justify a satellite broadcast. Obviously in the context of this work we are interested in peaks—respectively in values for R^B that relate to an exceeding of BT. Though, as it has been stated

Category	# videos
Most popular (today)	797
Most popular (week)	134
Most recent	5248
Most responded	65
Most shared	1099
Top rated	57
Trending videos	964

Table 5.2: Feeds observed in measurement of 2011/04/05

for BT, determining a generally applicable value for R^B is impossible. Accordingly in this chapter different, considerably high numbers are used for R^B which are arbitrarily chosen. As far as not indicated differently, $R^B = 10000$ is applied in the remainder of this chapter. Further the peak period T has been chosen in a way that it reflects the YouTube update interval as far as possible and is 60 minutes for the majority of measurements. Deviations from this standard value will be highlighted when applicable.

Timely scope of the data collection

The first measurements for this study have been performed in November 2010, the last ones in February 2013. During that time, more than 100 measurements have been made, each containing up to several hundreds of thousands datasets. A fraction of these were presented and analyzed in [63]. The results shown in this chapter are also based on a subset of the data.

Analysis of first results

The analysis starts with a set of data collected on a measurement period from 2011/04/05 to 2011/04/16. The observed standard feeds as well as the number of recognized unique videos are shown in table 5.2. Besides these, also data for 16 videos from the channel of user *BarelyPolitical* and for 10 videos from the channel of user *RayWilliamJohnson* has been recorded. The first analysis concentrates on the occurrences of peaks within the different feeds respectively channels under varying values for R^B. The results for $R^B = 10,000$, $R^B = 50,000$ and $R^B = 100,000$ are shown in tables 5.3, 5.4 and 5.5.

Interpretation of the results

Since each peak potentially justifies a broadcast, the column # *peaks* is of particular interest, together with the average increase of the view counter per peak in the adjacent column. The latter depicts the maximum number of unicast transmissions that could have been saved each time in case the proposed hybrid network content delivery model was utilized. Thereby the total number

Category	**Videos with Peaks**	**# peaks**	**∅ incr. per peak**	**∅ peaks per video**
MostPopular (today)	450	2358	29278	2.9586
MostPopular (week)	53	570	20671	4.2537
MostRecent	0	0	0	0
MostResponded	2	10	13974	0.1538
Most shared	301	2567	35233	2.3358
Top Rated	44	1186	42027	20.8070
Trending videos	280	2412	33659	2.5021
RayWilliamJohnson	7	234	58696	23.4000
Barelypolitical	7	43	15478	2.6875

Table 5.3: Peaks after category. $M_B = 10,000$

Category	**Videos with Peaks**	**# peaks**	**∅ incr. per peak**	**∅ peaks per video**
MostPopular (today)	91	278	101692	0.3488
MostPopular (week)	8	25	71388	0.1866
MostRecent	0	0	0	0
MostResponded	0	0	0	0
Most shared	82	388	116010	0.3530
Top Rated	9	193	154690	3.3860
Trending videos	70	333	117314	0.3454
RayWilliamJohnson	4	76	127308	7.6000
Barelypolitical	0	0	0	0

Table 5.4: Peaks after category. $M_B = 50,000$

Category	**Videos with Peaks**	**# peaks**	**∅ incr. per peak**	**∅ peaks per video**
MostPopular (today)	25	73	196879	0.0916
MostPopular (week)	2	3	138266	0.0224
MostRecent	0	0	0	0
MostResponded	0	0	0	0
Most shared	34	161	182823	0.1465
Top Rated	4	87	262078	1.5263
Trending videos	36	121	202556	0.1255
RayWilliamJohnson	3	36	191801	3.6000
Barelypolitical	0	0	0	0

Table 5.5: Peaks after category. $M_B = 100,000$

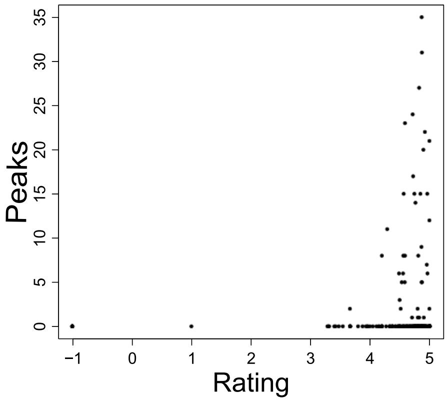

Figure 5.1: Peak-rating correlation for search string "Fukushima"

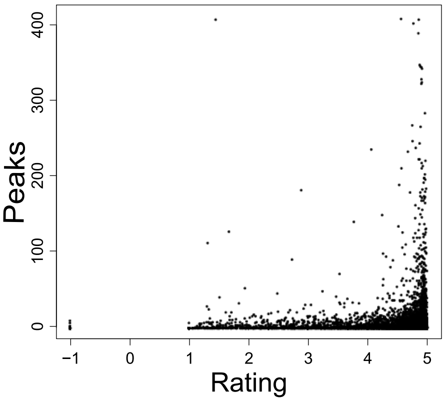

Figure 5.2: Peak-rating correlation for standard feeds

of videos with peaks in tables 5.3 to 5.5 is the minimum amount of necessary file broadcasts, which can be reached when the caches are able to store such a video for a sufficiently long time. The data of the first row in table 5.3 is used for the following example. In that case 450 broadcasts might have substituted $2358 * 29278 = 69037524$ unicast transmissions. Even assuming only a relatively small file size of 10 MByte, this would mean that the Internet traffic was reduced by more than 658 TByte. On the other side less than 4.4 GByte needed to be broadcasted. With other words each 10 MByte broadcast saved over 1.46 TByte of unicast traffic in the best case. With $R^B = 100000$ it was even more than 5.48 TByte. In this case, the broadcast efficiency would be higher, at the price of a lower total traffic reduction in the unicast network.

An example for potential savings with the proposed content delivery model

The last column of the tables 5.3 to 5.5 is interesting since the values provided there indicate the probability of a video from that particular feed respectively channel to experience a peak in user requests. This is of significant importance for a prediction of future user demand and thus for triggering broadcasts. Comparing the different feeds/channels we observe that for the *most recent* videos in that measurement no peaks occurred at all, no matter how R^B is chosen. From that it can be derived that it would be pointless to apply our content distribution model if no further information on the future peak-probability of a video was available.

Peak probability for distinct channels

Further the tables show an extraordinary high probability for peaks for videos in the feed *top rated* and those on the channel of user *RayWilliamJohnson*. The former indicates that a high rating considerably attracts potential viewers. We further find evidence for the correlation between high ratings and the probability for peaks in figures 5.1 and 5.2 [3]. Figure 5.1 is based on the

Identifying characteristics of peaky videos

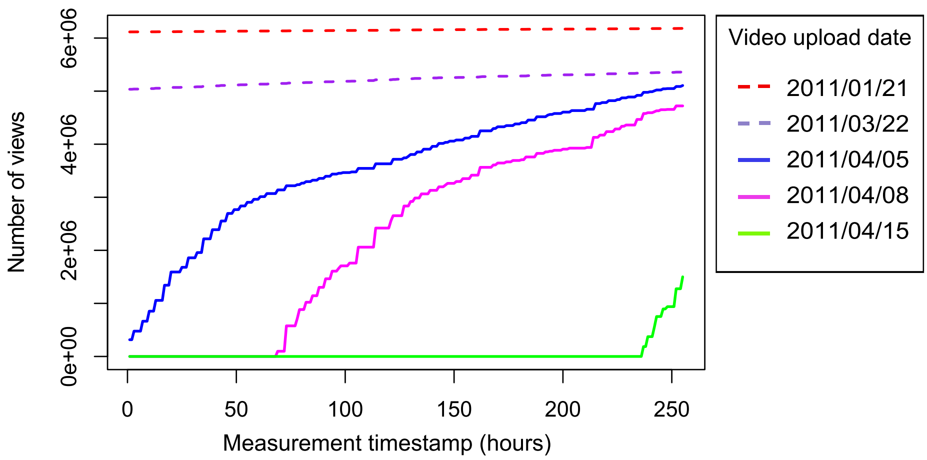

Figure 5.3: View count development for videos from user-channel of RayWilliamJohnson

data recorded for a special search string named "Fukushima" that will be further discussed later in this section. Figure 5.2 relies on data collected for the same set of feeds as shown in table 5.2 during the complete month of May in 2011.

Channel subscribers not always decisive for peak in videos

In case of videos from the channel of user *RayWilliamJohnson* the high peak probability can be partially reasoned by the high number of channel subscribers which are all informed about new channel videos at the same moment. However, the user-channel of *barelypolitical* has a similar high number of subscribers but the peak-probability for a video is much lower. The reason why *RayWilliamJohnson* videos have so many peaks can not be derived from the measurement results. However, there is a considerably easy way to predict the timely occurrence of peaks for new videos in this channel, and that is based on experience. Each video that is being uploaded to this channel experiences almost the same development as its predecessors. We see the progression of the view counter for five videos from this channel in figure 5.3. The upper two, dashed curves belong to older videos. The changes that can be observed for these mature videos are not significant anymore. For the other videos the day of their upload lies less in the past, partially it falls even into the measurement period. They all exhibit a fast increase within the first days after the video has been published and a decreasing gradient after day three. Later, this will be followed by a rather flat progression for the remainder of their lifetime, like we observe it for the two older clips. The same effect can be noticed for all videos from this channel during all measurements. Further it seems to be characteristic

Progression of videos' view counts for an exemplary popular channel

[3] In these figures a rating of -1 indicates that there was no rating information provided by YouTube for the corresponding video.

Upload date	#Views during observation	#peaks	∅ increase per peak
2011/04/15	1500502	8	187562
2011/04/08	4721440	54	87339
2011/04/05	4790816	73	64681
2011/03/22	326883	4	11971
2011/01/21	65565	0	0

Table 5.6: Peaks in Ray William Johnson's Videos during measurement period.

for channels with many subscribers, except for a varying duration of the distinct phases and different absolute view counts. For example the progression of view statistics for videos from BarelyPolitical's channel have almost the same shape, even though the maximum view count is considerably lower.

Peak occurances during video lifetime

The visual impression that we get from figure 5.3 is substantiated by the corresponding peak analysis results shown in table 5.6. The oldest video in the table does not exhibit any peaks at all, even though it still receives numerous requests. This shows that requests for older videos occur less frequently and are rather evenly distributed. All other videos in this category (not shown in the table) which had an higher age did not have any peaks either. Further it can be observed that the intensity of peaks—by means of average increase per peak—is highest during the first hours after the video upload and then decreases, which perfectly matches the progression of the curves in figure 5.3. Further it fits the observations made by the authors of [22], who report that most videos have their peak at the first day. This agglomeration of peaks in the first days of a video's lifetime is characteristic for channels with a extremely high number of subscribers. This is not surprisingly, since a subscription implies a notification about new videos which enforces early views. For the content distribution model this means that such videos are a perfect choice for broadcasts, since not only the rough number of future viewers but also their identity is known in advance.

Catching peaky videos by harnessing social networks

Another approach for finding videos with a high probability for peaks in the near future is monitoring social networks like Facebook[4], Google+[5] or Twitter[6] for the most discussed terms, the so called *hot topics*. Two of them in march 2011 were *Fukushima* and *Osama*. They have been used as a search string in the YouTube API and the results have been added to the observation list. The

[4] www.facebook.com
[5] plus.google.com
[6] www.twitter.com

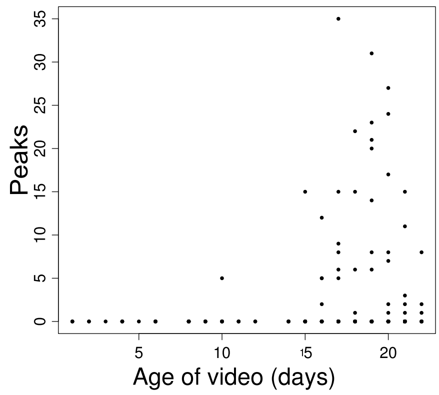

Figure 5.4: Correlation of video age and brusts

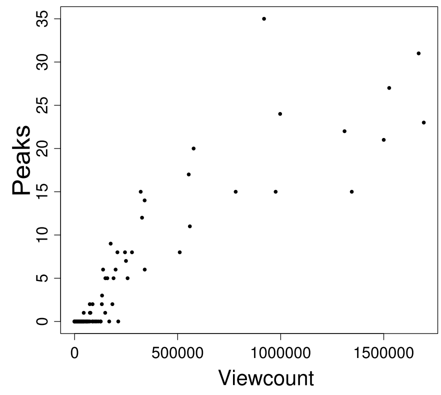

Figure 5.5: Correlation of total number of views and peaks

results of the peak analysis are shown in table 5.7. As is can be observed the average peaks per video are lower than for the most YouTube standard feeds (compare tables 5.3 to 5.5). Especially the feed *trending videos* delivers slightly better results. Nevertheless, the number of peaks as well as their intensity is notable. Thus this is further evidence that taking social network trends into consideration can further improve the content delivery efficiency.

Delay between peak occurrences and triggering event

Further analysis has been made for the Fukushima data sets. Thereby the data has been filtered for videos whose age at the end of the measurement was less than 23 days. This limits the videos to those who were uploaded after a heavy natural disaster that especially striked Japan's prefecture Fukushima. This was followed by a meltdown at the nuclear power plants located there. These dramatic events were causative for a significantly increased interest in this topic. A notable aspect here is that the peak in peak occurrences appears several days after the disaster, as it can be observed in figure 5.4. This delay would allow to broadcast such videos even before the majority of peaks have emerged and thus to reduce the induced traffic.

In figure 5.5 the correlation between the number of peaks and the total views

Search string	# videos	Videos with Peaks	# peaks	∅ incr. per peak	∅ peaks per video
Fukushima	313	38	418	33651	1.3355
Osama	348	46	339	26473	0.9741

Table 5.7: Peaks for trending Twitter topics ($M_B = 10,000$)

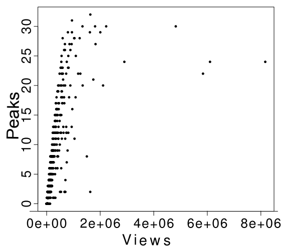

Figure 5.6: Correlation of peaks and views for data set from beginning of March 2011

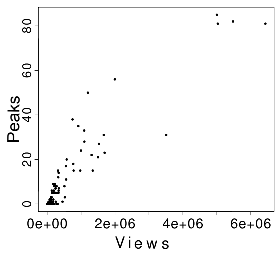

Figure 5.7: Correlation of peaks and views for data set from end of March 2011

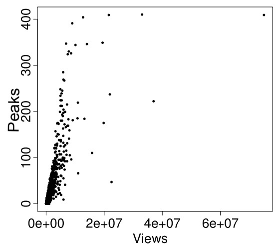

Figure 5.8: Correlation of peaks and views for data set from April 2011 (whole month)

Correlation between the number of peaks and total views

of videos from this data set is illustrated. Apparently the requests for very popular content are indeed not uniformly distributed but occur concentrated in peaks, which is a great benefit for the proposed content distribution model. In order to substantiate this finding the same analysis has been performed for other measurements. Figure 5.6 is based on a data set obtained at the beginning of march 2011. The data for figure 5.7 was recorded at the end of march 2011 while figure 5.8 relies on results collected in a measurement that lasted for the complete month of April 2011. Also here a correlation between the absolute popularity of a video and the number of peaks can be observed.

Peak occurences by categories

We conclude with an analysis of the relation between peaks and specific YouTube categories. The results are visualized in figures 5.9, 5.10 and 5.11 for varying values of R^B. A notable cumulation of peaks can be observed for the categories *Comedy*, *Entertainment* and *Music*. The significance of this effect varies with the changing R^B but the general trend remains. This knowledge can and should also be utilized when estimating the future peak–probability of videos since it allows to further narrow down the scope of videos that potentially will become eligible for being broadcasted.

Chapter conclusion

It can be summarized that peaks in video request pattern can be frequently observed. Further, they happen with increased probabilities in certain categories, feeds and channels. The provided results do not allow a precise prediction of future peaks for arbitrary videos. What they clearly disclose is that online content such as the videos offered by online video providers is highly suitable for the content distribution model proposed in this thesis, relying on a hybrid network infrastructure.

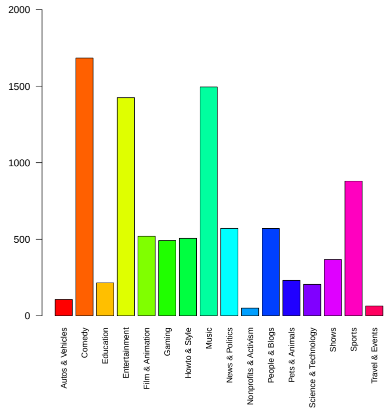

Figure 5.9: Peak occurrences by categories for $M_B = 10000$

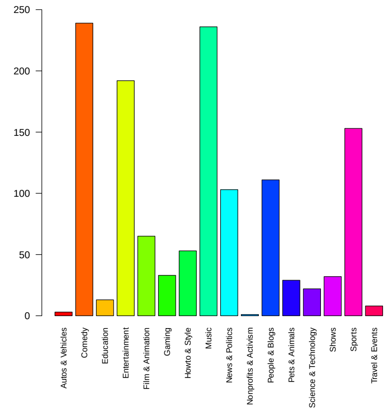

Figure 5.10: Peak occurrences by categories for $M_B = 50000$

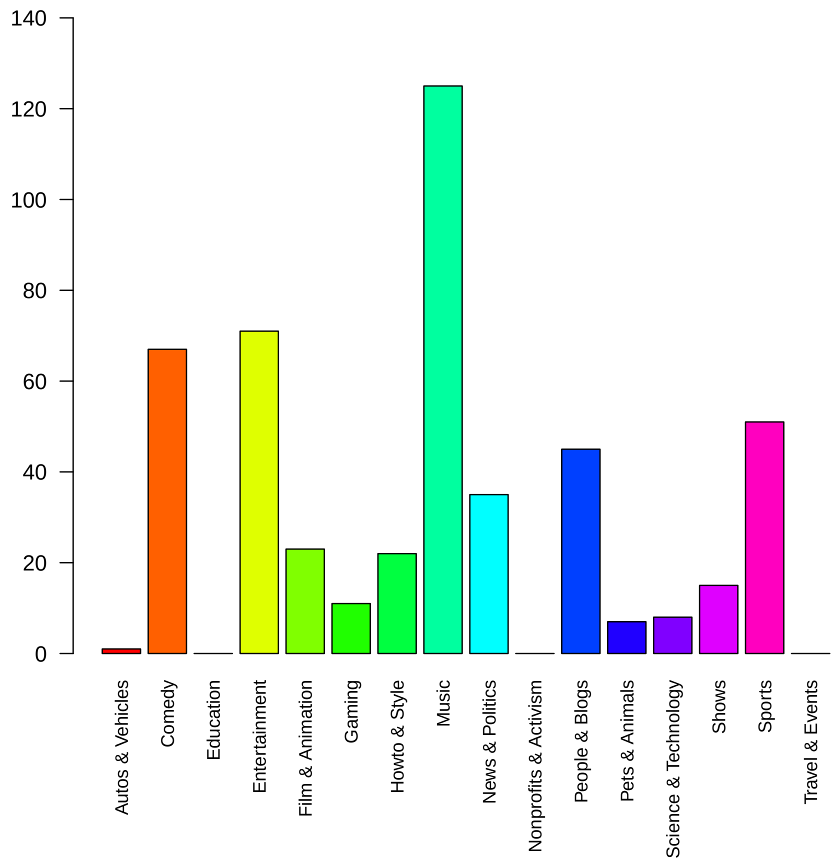

Figure 5.11: Peak occurrences by categories for $M_B = 100000$

CHAPTER 6

Analyzing Dynamic Broadcasts

Analyzing traffic savings for broadcasting pieces of concurrent downloads

In this chapter a very basic evaluation of the content distribution model is being made. The results shown herein have been published in [67]. The scope of this analysis is limited to live transmissions of currently requested files. In consequence there is no in-advance distribution, no prefetching, no P2P distribution and no caching taken into account. In this case the model solely relies on files having access patterns which exhibit a timely correlation to a high degree. In the studies presented in chapter 5 it has been shown that videos exist which show a high quantity of peaks with strong amplitudes. The analysis presented in this chapter is based on this finding and relies on the statistical information gathered during the previously presented analysis of user access patterns in YouTube online videos. In consequence the results of this study are only applicable for the distribution of video files. Since this is an application that is also very popular on mobile devices, these are especially considered here. Thereby the mobile clients may have Internet access either via the mobile network or via WiFi when they are near a home gateway. With regard to improvements in mobile satellite reception it is further assumed that the involved mobile clients can always receive satellite broadcasts with the same data rate as the HGW. Thereby it is postulated that they get the broadcasts bypassed by the HGW in case they are inside a building. For the distribution of the video files a client/server approach is being used. However, the files are partitioned into smaller pieces which are independently submitted as it is common practice in P2P approaches. In the following section 6.1 the simulation model is introduced, followed by the corresponding evaluation in section 6.2.

Putting special emphasis on mobile clients

6.1 Simulation Model

Specification of simulation parameters

For the distribution each file is subdivided into pieces of size S_P. The Internet access bandwidth for all clients is equal and referred to B_M. For both a homogeneous configuration is used, thus all clients have the same bandwidth and all files use the same piece size. Both are set to the same value thus that one piece can be submitted per second via unicast. This further implies that this denominates also the maximum encoding bitrate for videos in case a fluent playback is intended. Thus in general the same value will be also assumed for the encoding bitrate. However, in the evaluation in section 6.2 also a higher video encoding bitrates will be used during the simulations.

Regardless of a specific encoding bitrate, given a size S_F for a file F, this means that F is split into

$$|P_F| := \frac{S_F}{S_P} \tag{6.1}$$

Ratio between satellite and devices unicast download bandwidth

pieces. Let further the available satellite bandwidth be B_S. Whenever the number of active downloaders exceeds the broadcast threshold BT, a broadcast is scheduled. In the common case the satellite bandwidth is high enough to transmit several pieces within one second. Let r be defined as the ratio between the files transmitted per second via mobile network and via broadcast depends on the available mobile network bandwidth. Thus it is

$$r = \frac{B_S}{B_M} \tag{6.2}$$

Sequential download order

Due to the assumptions that have been made with respect to the clients Internet access bandwidth, the piece size and the video encoding bitrate all pieces will be downloaded in sequential order via mobile network. Thus when BT is reached the piece with the highest sequence number $piece_{max}(F)$ that at least one on the active downloaders already holds is determined. In the next step all pieces with a sequence number higher than $piece_{max}(F)+2$ and lower or equal $piece_{max}(F)+2+r$ are broadcasted. In case that

$$piece_{max}(F) + 2 + r > |P_F| \tag{6.3}$$

Granularity of satellite bandwidth allocation

we change the start sequence number for the broadcasted pieces to $|P_F| - r$ in order to utilize the full available bandwidth. This means in that special case some pieces are broadcasted even if BT is not reached for them. However, the remaining bandwidth could not be used for other files since one second is the finest granularity for which satellite capacity is being assigned. Using it this way further increases the download speed for the remaining clients.
While this broadcast is being made all clients continue their unicast download of the next piece in line as it would have been the case without a broadcast. This is why the 2 is added the highest sequence number in equation 6.3. Otherwise,

some pieces would be received twice by some clients.
Two other parameters being required are the maximum number of clients that can start watching a video per second (CpS) and the probability P that they will. Both are required to be adjusted in order to achieve realistic request patterns. The evaluation of this model in the following section will show the possible traffic savings.

6.2 Evaluation

Simulation parameters: Internet access bandwidth

The first important question concerning the evaluation of the proposed model is the choice of the parameters. For the clients the Internet connection bandwidth is chosen by the expected smallest common denominator. It is assumed that this is the bandwidth usually available in the standard *Enhanced Data Rates for GSM Evolution* (EDGE), being 236 kbit/sec. Thus let

$$S_P = 236kbit \tag{6.4a}$$
$$B_M = 236kbit/sec \tag{6.4b}$$
$$B_S = 36,000kbit/sec \tag{6.4c}$$

Simulation parameters: Client join rate

Another very critical parameter is the number of users that might request a certain file per step respectively per second. Here the results of the YouTube study presented in chapter 5 are used. There it has been shown that videos experience an average of 715.5 requests per minute during the first three days after they have been uploaded. This means there are approximately 12 requests per second in average. Due to the coarse grained measurement that has been discussed in chapter 5 no information is available how the requests are distributed within a period shorter than one hour. Thus a Poisson distribution is assumed. It is worth noting that this refers to the worst case for the objected broadcasts, since more uniform playback positions would be achieved in case the downloads did occur peaked in a short interval and thus result in a increased efficiency. According to the assumptions just made the parameters are set to

$$CpS = 7,155 \tag{6.5a}$$
$$P = 0.0017 \approx \frac{12}{7155} \tag{6.5b}$$

Simulation termination condition

which results in approximately 12 users joining the file download per second. The simulation ends after 100,000 clients have completely downloaded the file. Further an average video duration of 342 seconds is used, which relies on the results for very popular videos achieved in chapter 5. For the simulations two different video encoding bitrates are used. The first is $1,057kb/sec$ which is often used for video with the resolution 480p and an audio bitrate of $96kb/sec$.

Video enccoding bitrate and file sizes

It results in an average video file size of $\approx 48.16MByte$. These bitrates respectively this video resolution has been chosen since it is a common resolution for screens of mobile devices. The second is $236 kbit/sec$ and thus matches the assumed maximum download bandwidth of mobile clients. This will allow a fluent playback on mobile devices even without buffering and in absence of broadcasts. The corresponding file size is 10.01 MB. Figures 6.1 and 6.2 show the download

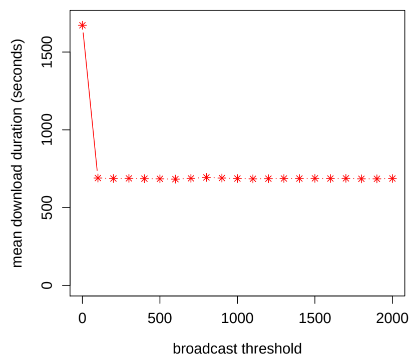

Figure 6.1: Mean download duration for varying BT and an encoding bitrate of 1,057 kbit/sec

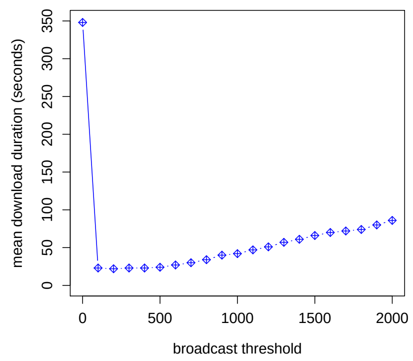

Figure 6.2: Mean download duration for varying BT and an encoding bitrate of 236 kbit/sec

Impact of BT on the distribution duration

duration for varying values of BT for both encoding bitrates. Thereby $BT = 0$ means that broadcasts are disabled. In both cases the drop in the mean distribution time is tremendous as soon as broadcasts are enabled. While for the bigger file, respectively the higher encoding bitrate, this seems to remain constant, for the smaller file slowly increasing values can be observed. In order to analyze this behavior in more detail, a simulation with a wider range of BT has been performed, with coarser steps. Figures 6.3 and 6.4 illustrate the corresponding

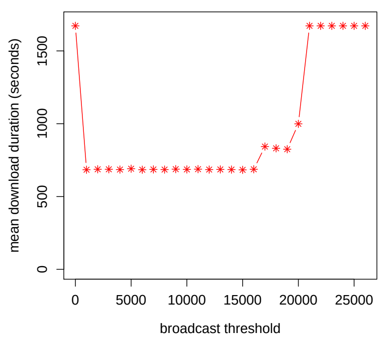

Figure 6.3: Mean download duration for varying BT and an encoding bitrate of 1,057 kbit/sec, coarse grained

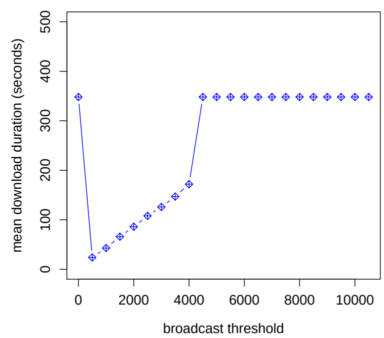

Figure 6.4: Mean download duration for varying BT and an encoding bitrate of 236 kbit/sec, coarse grained

results. It is confirmed that the duration stays on a constant level for the bigger file. However, then it experiences a rapid increase. For the smaller file this seems to be a more gradual development. However, what both results have in common is that from a certain value of BT on, the mean duration falls back on the figures that have been observed when broadcasts were disabled. This is caused by the probability of users to join per second, the maximum number of clients that can join in a second, the download bandwidth in the unicast network and the size of the file. Assuming that exactly 12 clients join each second, a file consists of 10 pieces and one piece can be downloaded per second. This means every client would finish its download after 10 seconds. During this time 120 clients would have started downloading in total. In consequence if BT is greater than 120 a broadcast can never occur. In general this means that no broadcast will be possible if

Upper bound for BT depending on client joint rate and file size

$$BT \geq \frac{S_F}{B_M} \cdot \mathrm{CpS} \cdot P \equiv |P_F| \cdot \mathrm{CpS} \cdot P \qquad (6.6)$$

The latter equivalence only exists because equal values are being used for B_M and S_P. This implies that the bigger the file size, the higher BT can be chosen if the client's download bandwidth in the unicast network remains constant. On the other hand, BT must potentially be decreased if B_M grows. However, lowering BT reduces the efficiency of a broadcast. In this context the cost for a broadcast must be compared to that of the unicast delivery. It must also be taken into account that the model as it has been defined in section 6.1 will only work reasonably if $r \leq 1$. Otherwise, broadcasts could still be used to reduce bandwidth consumption in the unicast network. However, it can be assumed that the user acceptance would be very low if they have to wait longer for the broadcasted than for the unicasted content.

Critical performance factors and constraints

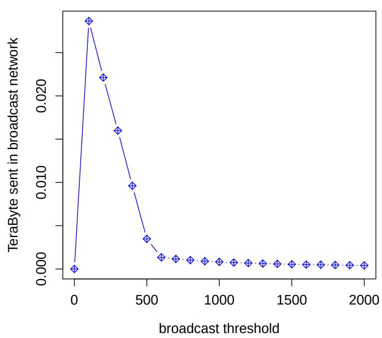

Figure 6.5: Broadcast traffic in TByte for varying BT and an encoding bitrate of 236 kbit/sec

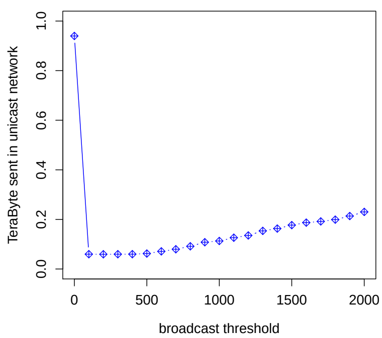

Figure 6.6: Unicast traffic in TByte for varying BT and an encoding bitrate of 236 kbit/sec

Comparison of bandwidth consumption

Next the bandwidth consumption in the unicast and in the broadcast network is analyzed and compared. As figures 6.5 and 6.6 show the demand on satellite capacity is very high for small values of BT, which is not surprising. However, it quickly drops to very low values. On the other hand the unicast traffic immediately falls when the broadcasts start and start increasing only verly slowly with growing BT. For the bigger file the situation is different, as

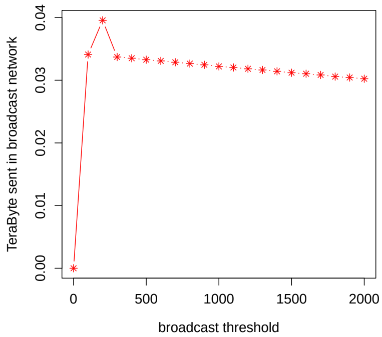

Figure 6.7: Broadcast traffic in TByte for varying BT and an encoding bitrate of 1057 kbit/sec

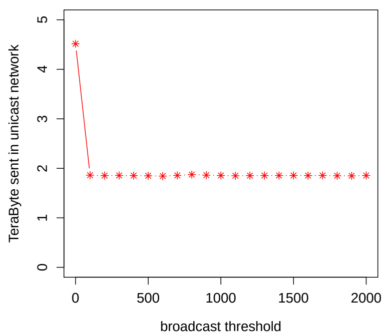

Figure 6.8: Unicast traffic in TByte for varying BT and an encoding bitrate of 1057 kbit/sec

figures 6.7 and 6.8 show. In the broadcast network the decline after the peak is not that distinctive. Though considering the much bigger file, the peak is comparatively low. While for growing BT the satellite bandwidth consumption slowly decreases, the traffic in the unicast network remains more or less constant.

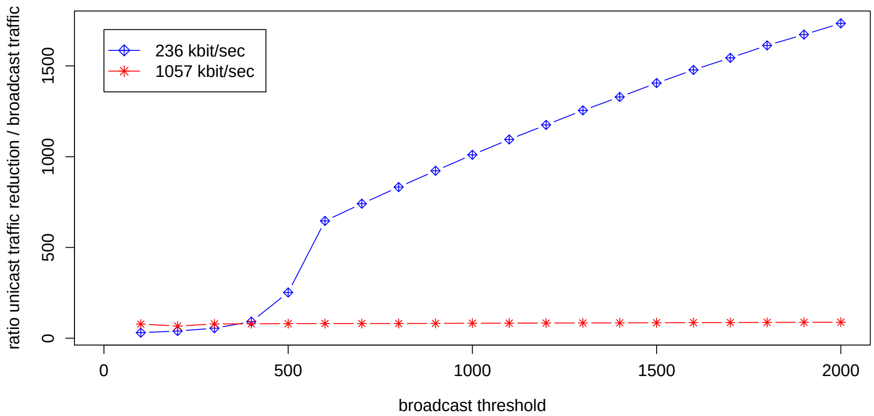

Figure 6.9: Ratio between traffic reduction withing the unicast network and bandwidth consumption in the broadcast network for varying BT

The gain of broadcasts in terms of traffic reduction becomes apparent in figure 6.9. While the ratio between bandwidth savings in the unicast and bandwidth consumption in the broadcast network remains constant at 50 for the bigger file, the smaller file shows an increasing gain. This is due to the fact that the demand for satellite bandwidth was considerably lower for this file distribution.

Chapter conclusion

In this chapter the very basic components of the proposed content distribution model have been analyzed. It has been shown that the unicast traffic can be significantly reduced depending on the parameters that are used. At the same time, the required satellite bandwidth is much lower. However, whether this difference is big enough in order to justify a broadcast must be further investigated. Further it has been shown that the integration of broadcasts can clearly reduce the average distribution time. What can be derived from the results presented in this chapter is that the broadcast efficiency benefits from a high number of concurrent downloaders. Thus the features like prefetching and caching which are discussed in the following chapters can be expected to bring a significant improvement of the content distribution model's performance.

Performance increases with rising number of concurrent downloads

CHAPTER 7

The SatTorrent Protocol

Chapter Introduction

The SatTorrent protocol is a modification of BitTorrent as mentioned previously. In this chapter a comprehensive knowledge about the latter is postulated since the details of BitTorrent will not be discussed again. This chapter concentrates on the modifications and extensions that are introduced by SatTorrent as they have been partially presented in [66].

7.1 SatTorrent Protocol

Requirements of the SatTorrent protocol

The pure SatTorrent protocol—without support for any integration of social networks—as it is described in this chapter has been first presented in [65] and [66]. It relies on a hybrid network infrastructure. The concurrent access to a unicast and a satellite broadcast network is thereby mandatory for the content provider side, in particular for the tracker. For the clients a mixed environment is supported where only a fraction of nodes has hybrid network access. This set of nodes should neither be empty nor so small that no benefit can be obtained from the broadcasts. Viable ratios between sat-enabled and non sat-enabled peers will be determined later in this chapter.

The two SatTorrent versions

Two different versions of SatTorrent have been designed. The first aims at a very low demand on satellite bandwidth. It only uses the broadcast facilities to distribute metadata to the peers and is referred to as **SatTorrent-M**. The second extends SatTorrent-M by further allowing broadcasts of P2P payload, more precisely pieces of files. This is considered as the complete version of the protocol and thus being named **SatTorrent**. Both versions are introduced and evaluated separately in the remainder of this chapter.

Name	Description
info_hash	20 byte hash value of the info part in the torrent file
peer_id	a string providing an ID for the peer
ip	the peer's IP address
port	the port number the peer is listening on
uploaded	bytes uploaded by the peer for this file
downloaded	bytes already downloaded by the peer for this file
left	bytes the peer still needs to receive in order to finish the download
location	the physical location of the peer, here the AS number
satenabled	set to true in case the peer is sat-enabled, false otherwise
event	peer status (STARTED, COMPLETED, STOPPED, EMPTY)

Table 7.1: GetRequest parameters

7.2 SatTorrent-M – The Metadata Broadcasts

Capabilities and constraints of SatTorrent

As previously discussed this approach only utilizes the broadcast channel for distributing metadata. In particular only the trackers are able to trigger broadcasts and in consequence only unicast tracker response messages can be substituted by satellite transmissions. This means that the payload transfer—by means of piece carrying peer wire messages—remains in the unicast network. It is worth noting that thereby the number of these messages, the biggest ones in size, remains unchanged since the data still needs to be sent to every single client. However, due to the increased knowledge that is provided to the sat-enabled clients an improved performance in the file exchange process is postulated. The expected gains are in the time demand for a file distribution as well as in the number of required control messages. The latter are all PWMs except those of type *piece*. In the following the modifications made to the BitTorrent protocol in order to facilitate and handle these metadata broadcasts are introduced.

Extension of tracker announce messages

As already indicated in chapter 4 the clients need to provide the tracker with additional information. Correspondingly the tracker announce messages are extended to support the exchange of these values. The complete specification of an announce message is given in table 7.1. The new fields that have been added to BitTorrent's GetRequest message are satenabled and location. The former indicated whether the sending peer is connected to a hybrid network. The latter provides information about the physical location of the peer to the tracker. However, this can be very coarse, e.g. for the current protocol version the autonomous system number is used here. This allows a very simple but effective metric that decides whether or not two peers are within the same

Location awareness in SatTorrent

AS and thus are preferable exchange partners. It is worth noting that a high throughput can also be achieved with peers in other ASs that are far away. However this might induce a higher cost for sending the data through a higher level ISP network. We recall that this aspect of high cost for ISPs is one of the biggest problems BitTorrent was opposed to, since this caused several ISPs to take measures against the thereby provoked cost explosion such as traffic throttling attempts for P2P packets as it has been mentioned in [59]. Thus avoiding such potential problems increases the chances for a broad acceptance and support of the SatTorrent protocol.

Tracker handling new downloads

At the beginning of a download, the tracker will receive such an announce message with event code STARTED. In turn it will add information to its dictionary that the announcing peer is now downloading the specific file and has no pieces so far. In case this is a resuming of an interrupted download, this entry will already be present. However, the tracker will then have to change the peers download status to active again. The further line of action taken by the tracker depends on the entry `satenabled`. If this is set to false the tracker simply composes a response message in the style of the BitTorrent protocol. The same applies in case the parameter value is true an error should have occurred during the request handling. Otherwise, a tracker response message broadcast is scheduled, in case this has not already been done for a previous request. These broadcasts are done in certain intervals (ranging from 5 to 60 seconds) and thus introduce a little delay compared to unicast responses which are always sent immediately. Making a broadcast directly after each request would solve this problem but would lead to an excess of broadcast messages. Due to the just mentioned delay the startup performance for sat-enabled peers would be degraded. This is especially critical in the beginning of a file's distribution since it would slow down the piece replication and thus would delay the load reduction for the original seeder, e.g. the content provider. Thus a threshold is introduced that causes the sending of an immediate tracker response via unicast in case the total number of sat-enabled peers known by the tracker does not exceed its value. This threshold is referred to as the **tracker broadcast compensation threshold (TBCT)**. Eligible values for it will be evaluated in the subsequent section.

Broadcast tracker messages are sent in fixed time intervals

The two types of broadcast tracker response messages

A **Broadcast Tracker Response Message** (**BTRM**) can have two different forms. The standard design contains the fields listed in table 7.2. The most notable aspect is that it contains not only a subset of all peers in the swarm—as it would be the case in a normal response—but all of them. Further, for every peer it provides additional information such as whether it is sat-enabled, its AS ID, its bitfield for this file and the bytes the peers uploaded to other peers. The latter is not only for this file but for all of its downloads and allows a rating of

field name	description
message_id	4 bit field indicating the type of the broadcast message, *0001* for BTRM
info_hash	20 byte hash value of the info part in the torrent file that identifies the file
tracker_id	a string providing an ID for the tracker
announce	the announce interval given by the tracker
peerlist	a dictionary with the following info for each peer:
peer_id	the identifier chosen by the peer
IP	the peer's IP address
port	the listening port of the peer
uploaded	the total number of bytes the peer uploaded for all its downloads
ASID	number identifying the AS the peer is located in
satenabled	whether or not the peer can receive broadcasts
bitfield	the peer's bitfield for this file, indicating complete pieces

Table 7.2: BTRM specification

its altruism. Peers might prefer to upload pieces to altruistic peers in order to speed up the file distribution. The value of the other information is explained below when we examine the peer behavior.

Extended broadcast tracker response messages

The other form of a tracker response is the **extended BTRM**, or short **BTRM-E**. Instead of containing peer list and bitfields for one file, it contains a list of file-peerlist value pairs. The structure of a BTRM-E message is shown in table 7.3. In addition to the information contained in a BTRM, it further contains the upload and download bandwidth for each peer as well as the number of bytes uploaded, downloaded and missing for each peer-file pair. Obviously this provides the recipients with more information that can be used for increasing the quality of exchange partner selection. On the other hand this increases the satellite bandwidth demand for these messages. Whether or not this is desirable depends on the quality of service the provider wants to offer.

How peers handle broadcat tracker responses

When a peer receives a BTRM or BTRM-E, all information it provides is added to the locally stored data. This means that the recipient may learn the IDs, addresses and ports of new peers as well as their download progress—by means of the bitfield vectors—of one or potentially more downloads. It is worth noting that even if the standard BTRMs are used, each client can receive and process broadcasts that do not belong to its current downloads. Storing the corresponding data can be of high value for future downloads as the evaluation in section 7.2.1 will show. Let us consider the case of a peer that just

field name	description
message_id	4 bit field indicating the type of the broadcast message, *0010* for BTRM-E
tracker_id	a string providing an ID for the tracker
announce	the announce interval given by the tracker
filelist	a dictionary with the following info for each file:
info_hash	20 byte hash value of the info part in the torrent file that identifies the file
peerlist	a dictionary with the following info for each peer:
peer_id	the identifier chosen by the peer
IP	the peer's IP address
port	the listening port of the peer
dl_bw	the peer's download bandwidth
ul_bw	the peer's upload bandwidth
bytes downloaded	bytes downloaded for this file
bytes uploaded	bytes uploaded for this file
bytes missing	bytes missing for this file
ASID	number identifying the AS the peer is located in
satenabled	whether or not the peer can receive broadcasts
bitfield	the peer's bitfield for this file, indicating complete pieces

Table 7.3: BTRM-E specification

Differences between BitTorrent and SatTorrent

started a new download. With the BTRM it receives a list of potential exchange partners. In case of BitTorrent, this would be a short list, a fraction of all peers selected randomly by the tracker. In that case the client connects to all peers on the list, a handshake is performed and bitfields—potentially empty ones—are exchanged. This is different for SatTorrent. Here the sat-enabled client has a complete list of all peers downloading that file. Connecting to all of them is not an option since such a high number of open connections is detrimental for the overall performance due to the characteristics of the underlying TCP/IP protocol stack. The SatTorrent client chooses peers for connecting according to the provided information and gives priority to peers who are known to be useful in terms of being able to provide missing pieces. Those who are in the same autonomous system and who possess at least one piece that is still needed are connected to with priority 1. Peers that are not is the same AS but can provide missing pieces are in priority class 2. Third and last, all other peers are assigned priority 3. In the style of the BitTorrent equivalent, the SatTorrent client has a limit for maximum open connections. For SatTorrent this is being referred to as **open connection limit (OCL)**. Connections to peers in priority class

SatTorrent's enhanced peer connecting

The open connection limit

2 and 3 are only established with a certain probability ρ. If we refer to the number of existing connections as C, then $C - \mathrm{OCL}$ denominates the amount of connections that can still be initiated. Then ρ is determined by the quotient of $C - \mathrm{OCL}$ and OCL and a parameterizable factor f_{conprob}. The latter allows a further adjustment of the probability for connecting to this kind of peers. Thus we get

$$\rho = \frac{(C - \mathrm{OCL}) \cdot f_{\text{conprob}}}{\mathrm{OCL}} \tag{7.1}$$

In other words, connections to other nodes who are less desirable as exchange partners get more restricted as the number of existing connections grows. The purpose is to restrict such connections in case a sufficient number of class 1 peers are available while avoiding starvation in case none or too little of this kind are present.

Bitfield suppression for Sat-Peers

As for BitTorrent, each connection starts with a handshake (see chapter 2.6.1). However, in contrast to BitTorrent, a sat-enabled client will not request a bitfield from the remote peer if this has been received via broadcast. Further it will only send its own bitfield in case the connection partner can not receive broadcasts. Since the bitfield messages are the second biggest PWMs after the piece messages, this is expected to significantly reduce the traffic. This SatTorrent feature is referred to as **bitfield suppression**

Location aware unchoking

Restricting distant peers

The location information is further used for SatTorrent's unchoking algorithm. However, in this case the peers in the proximity are not exclusively served first. This would potentially hinder a distribution of files respectively of pieces across AS boundaries. Instead, in case more candidate peers are available than can be unchoked, nodes in a foreign AS are considered with a reduced probability which can be adjusted in the client settings. This is done for the normal unchoking as well as for the optimistic unchoke. In this context the bitfields are not incorporated. Except for the optimistic unchoke, only peers that announced their interest in a piece—by means of a PWM of type *interested*—are considered. Thus it is in the responsibility of the connected peers to state which are eligible for being unchoked. The broad availability of bitfields would allow a reversal of this process, letting the unchoking client also determine the peer's piece demand respectively it's interest. However, the other approach must still be supported for non sat-enabled peers. Further, due to the broadcast interval there is a slight delay in the information distribution. Thus the data the interest is ascertained on might not be up to date. As a consequence, a peer might be unchoked that cannot be provided with a missing piece anymore. This would cause confusion and delays unckoking of peers with real demand and therefore is not applied.

For BitTorrent the extension that is referred to as *have-message suppression* has been developed. SatTorrent adopts this feature and adds a further enhancement, the **strict have-message suppression**. It eliminates all *have* messages to sat-enabled peers and only sends one to the tracker, who updates the corresponding bitfield which is then distributed with the next broadcast. The only exception from this rule is made for peers that are currently downloading from the client who just finished the piece download. They still get a unicast *have* message. The reason is that downloaders, which are unchoked peers, should have the opportunity to download as many pieces in a row as possible. Since the newly downloaded chunk might be the only one such a peer still requires, the instantaneous status update is very important. Further, all non sat-enabled peers still must be informed about the successful piece download by means of an immediate have message. In spite of these exceptions this approach can significantly reduce the number of messages. However, it also introduces a delay in information of most sat-enabled peers. The evaluation in the following section will show that this still allows SatTorrent to perform better than BitTorrent.

Strict have-message suppression

7.2.1 Evaluation of SatTorrent-M

In order to facilitate the evaluation of SatTorrent and allow a performance comparison to BitTorrent, a Simulator has been implemented that is capable of simulating both under different conditions. It must be admitted that such simulators have a high memory consumption. This is due to the fact that every client must store the bitfields of all known peers. At least in case of sat-enabled peers, this corresponds to the total number of clients in the network. Further, since there is an practical upper bound of 512 kByte for the chunk size[102], the complexity of bitfields is determined by the size of the corresponding file. Thus both, file size and number of nodes in the network cause an exponential growth of memory demand. In consequence these two parameters can not be increased arbitrarily. In particular this does not allow simulating networks with hundreds of thousands of peers. Nevertheless the simulation results show a clear tendency towards increasing performance gains of SatTorrent for rising numbers of involved nodes and file sizes. This will be further discussed when the corresponding results are presented.

Concerning limitations of the simulation

File size and network order are limiting factors

Before starting the analysis of the various simulations, an introduction of the general simulation parameters is necessary. The assumptions that have been made for the content distribution model that is subject of this analysis lead to simulation parameters that are different from common BitTorrent simulations. The HGW, where the SatTorrent client is running on, is considered as an always on device. In consequence churn—an important aspect in most BitTorrent simulations due to a high peer fluctuation—is insignificant in this

Specifying general simulation parameters scenario. In consequence and due to the relatively short simulated time spans, churn is not being considered in the simulations presented in this section.

Seeding time in the proposed model

A further specialty in the context of the proposed hybrid network based content distribution model is the increased piece availability at the peers. Since the HGW is always on, always connected and due to the caching strategy, pieces that are once downloaded are available for other peers for a long time until they are purged as a consequence of the cache replacement strategy. This applies for completed downloads as well as for those that are aborted. For the simulations in the context of SatTorrent-M where one single file is being distributed among all peers this means that all clients remain as seeders after finishing their download until the end of the simulation. The latter is reached when all clients completed the file download. The time that elapses until that point is also a measure for the protocols content distribution performance.

Figure 7.1 shows this duration for varying ratios of sat-enabled peers. In

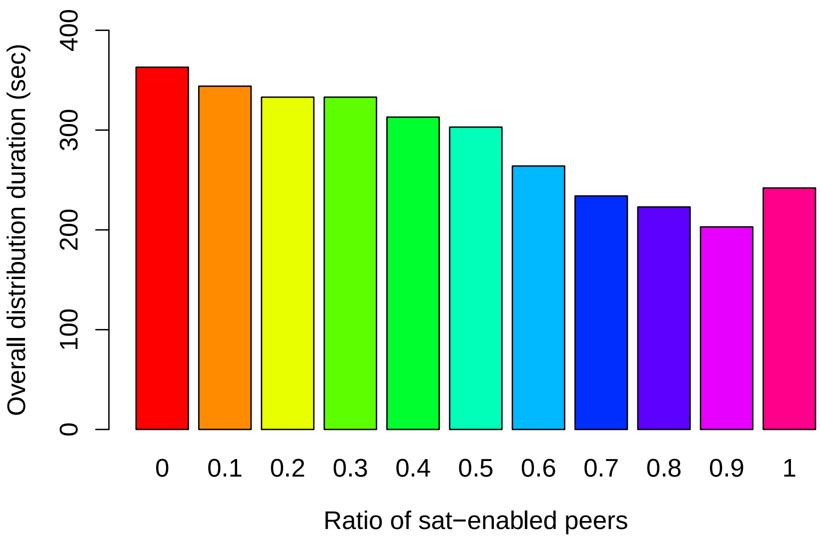

Figure 7.1: Distribution duration for varying values of SPR

Comparing distribution time for varying amounts of sat-peers

each simulation run a file of 20 MByte size is distributed among 2000 peers, one of them being the original seeder. Further all clients have the same asymmetric Internet connection bandwidth of 6 Mbit for incoming and 2 Mbit for outgoing traffic. The result on the very left for **sat-peer ratio (SPR)** set to zero reflects the time demand for BitTorrent where no satellites are involved at all. For all other results the clients implement the SatTorrent protocol, with SPR indicating the amount of peers connected to a hybrid network. We observe a reduction of distribution time as the portion of sat-enabled peers increases with just one exception. For a setting where only sat-enabled peers are in the network, the distribution time experiences an increasing again. This is an indication for a symbiosis between sat-peers and those who have no access to a hybrid network. The reason is twofold and reasoned by the little delays sat-enabled peers are exposed to as well as their restrictive behavior against peers that are not is the same AS or cannot be ensured to be valuable as exchange partners. This causes some peers to experience relatively long download durations. Surprisingly very little ratios of non sat-peers can fully compensate this effect in a heterogeneous network. This increase and measures to avoid it will be further investigated in section 7.3 when the full SatTorrent protocol is evaluated.

Increasing duration for settings with sat-enabled clients only

No negative effects for $SPR = 1$ are observed with regard to the amount of PWMs. As figure 7.2 shows, their number monotonously decreases as SPR

Message count for varying ratios of sat-enabled peers

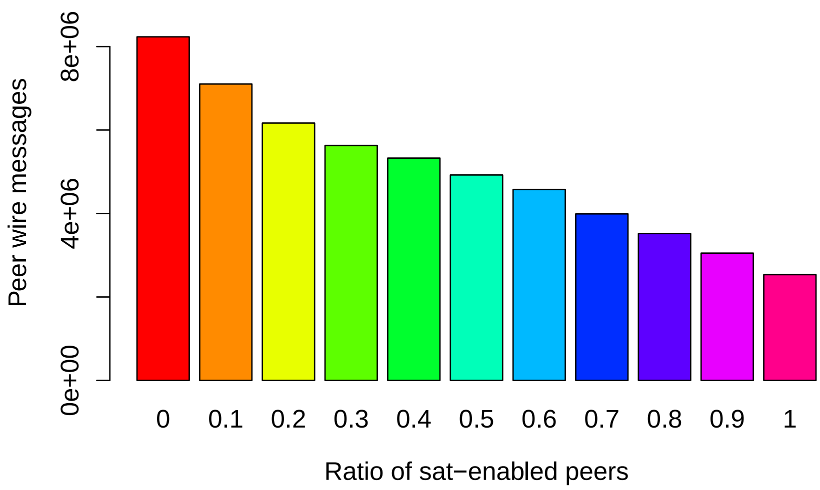

Figure 7.2: Peer wire messages for varying values of SPR

grows. Thus when the objective is maximizing the message savings, a higher portion of sat-enabled peers is always desirable. When distribution time is more important, a little share of non sat-peers should be aspired. It is worth noting that in all these simulations the number of required BTRMs never exceeded 17 and was at 13.6 on average. Thus the price for the benefits of SatTorrent is considerably low.

For the remainder of this evaluation we focus on optimizing the message complexity. Under these conditions we next analyze how the total number of messages evolves for BitTorrent and SatTorrent as the distributed file increases in size. For these simulations a network size of 1000 nodes has been used with

Analyzing the impact of increasing file sizes

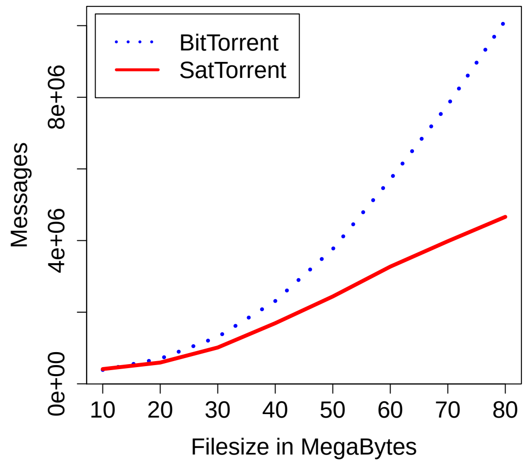

Figure 7.3: Message counts for varying file sizes.

High potential for message reduction observable

$SPR = 1.0$. The results are illustrated in figure 7.3. It can be observed that the total number of unicast messages in dependence of the file size is growing significantly slower for SatTorrent. It is worth noting that the file sizes that have been analyzed here are comparatively small, considering the size of movies and software packages offered online. This reduction cannot solely be justified by the tracker response messages that are substituted by BTRMs. In order to gain a better understanding of the particular changes in message complexity the various PWMs have been monitored isolated. Furthermore, different simulations have been made with certain protocol features of SatTorrent being disabled. As an experimental feature for the evaluation the **extended tracker announce messages** have been implemented. Using them will cause the peers to send information about all their downloads—including finished ones—such as bytes downloaded, uploaded and missing as well as the corresponding bitfields to the tracker with each announce message. This causes a lot of traffic which makes this approach unalluring for practical use. Especially since all the information in these extended messages can also be obtained by the tracker by consequently collecting from other sources (e.g. have messages, normal announce messages) and keeping peer information persistent even after a disconnect. Thus this feature is exclusively used in order to determine whether or not this data collection is working reliably.

Examining the tracker's functioning

In figure 7.4 we see the changing quantities of all aggregated unicast messages (total messages), *have* and *bitfield* messages. The different simulation settings for the individual results in the chart are as follows: Beyond this, for

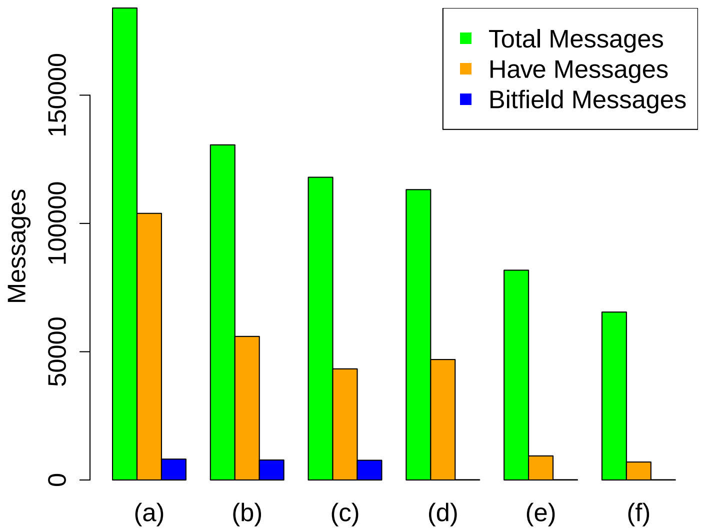

Figure 7.4: Amount of *have* and *bitfield messages* and total message count for varying combinations of protocol features.

all these simulations a file size of 10 MByte and a network size of 2000 nodes have been applied. As we observe in 7.4(a) and (b), the have message suppression greatly reduces the number of have messages—and the total messages to the same extent—already in BitTorrent. Comparing these values to those in 7.4(c) only a slight improvement is achieved. This is not surprising since all measures of SatTorrent that potentially can reduce the amount of messages are not applied in these settings. The small decrease is a result of a lower value for the OCL. Having less connections to other peers necessarily reduces the number of required have messages when a piece is received.

Interpretation of simulation results

For the results in 7.4(d) several features have been merged, in particular these are the *BitField Suppression* (BS), *Extended GetRequest* (EGR) and *Extended Tracker Response Mode* (BTRM-E). This is due to the fact that the activation of EGR and BTRM-E did not lead to any notable changes. Considering EGR this indicates the correct functioning of the tracker. With respect to the BTRM-E messages it shows that the information provided by the standard BTRMs is sufficient for achieving good results. What can further be observed in figure fig:modecomp1(d) is that the bitfield suppression significantly reduces the bitfield messages. More precisely they are eliminated. This is the expected behavior since all clients are sat-enabled and with bitfield suppression no bitfield messages are sent to such peers. However, at the same point we observe a slight

Correct functioning of the tracker confirmed

Complete avoidance of bitfield messages achievable

ID	description
a	BitTorrent without extensions
b	BitTorrent with Have-Message Suppression (HMS)
c	SatTorrent with HMS
d	SatTorrent with HMS, BitField Suppression (BS), Extended GetRequest (EGR) and Extended Tracker Response Mode (BTRM-E)
e	SatTorrent with HMS, BS, EGR, BTRM-E and Strict Have-Message Suppression (SHMS)
f	SatTorrent with HMS, BS, EGR, BTRM-E, SHMS and location awareness

Table 7.4: Legend for simulation parameter settings

increase of *have messages*. According to the protocol *have message suppression* avoids this type of PWMs to peers who already possess the corresponding piece. However, broadcasted bitfields introduce a certain delay, potentially as long as the broadcast interval of the tracker. Thus it might happen that a peer decides to send a *have message* based on an outdated bitfield. However, this can only be the case shortly after a new connection has been established, since otherwise the remote peer would have sent a *have message* already. Considering the bigger size of a bitfield message compared to a *have message*, the reduction of the former justifies this slight increase of the latter. This holds all the more with regard to the development of *have messages* in 7.4(e) where these messages are only send to the tracker and to non sat-enabled peers, which are not present in this simulation settings at all.

Priority for the reduction of big messages

The last feature that has been included in the simulation on which the results of 7.4(f) are based on is the location awareness. We observe a further reduction of total messages that can only be partly explained by the decreasing number of *have messages* that is shown in 7.4(f). In figure 7.5 it can be seen that there is further a considerable reduction of unchoke messages. What are the reasons for this behavior in consequence of location awareness? Many parts of the client—such as unchoking, connecting to new peers, piece requests—are much more restrictive when this function is enabled. This leads to a situation where clients tend to have less open connections, potentially also permit less unchokes and in consequence less concurrent downloaders. The latter is causative for the slight reduction of *have messages* that can be observed in figure 7.4 (f), since with HMS these are only sent to downloaders and to the tracker. All other variations in figure 7.5 are of low significance and not further discussed. Rather the implications of the location awareness will be investigated in more detail. What has been stated before, a potentially lower number of downloaders and

Evaluation of location awareness

Cautious behavior causative for further message reduction

Impact of cautious clients on download duration

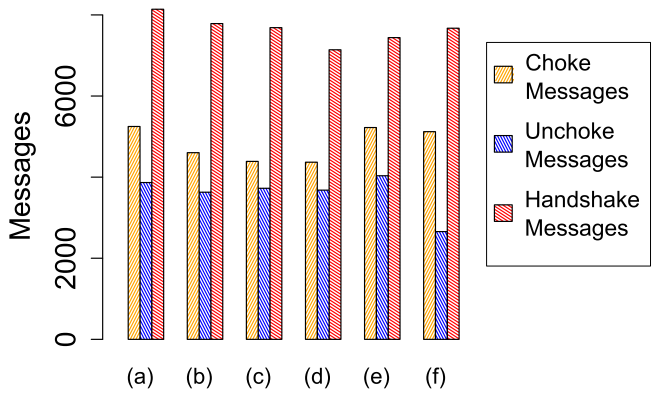

Figure 7.5: Amount of choke, unchoke and handshake messages for varying combinations of protocol features

more restrictive behavior in general indicates that the download duration might suffer from this. However this is not the case as the simulation results revealed. In simulations with $SPR = 1.0$, 2000 peers and a 20 MB file, where location awareness has been enabled and disabled in different runs, we detect an average total time demand of 242 seconds for the former and of 303 seconds for the latter. Thus at least for higher numbers of involved peers and well crowded AS this effect can not be observed. It must be admitted that in case there might be one or more AS with only very few peers, these still will experience a longer download duration. The probability for this issue increases inversely to the ratio between total peers and number of autonomous systems. This is the price of consequently avoiding inter AS traffic. It is worth noting that SatTorrent's purpose is to take load from the Internet by means of shifting transmissions of highly popular files to the satellite broadcast. Thus it can be assumed that in practice this inauspicious constellation will seldom occur. But does location awareness fulfill its objective of reducing inter AS traffic? When we have a look at figure 7.6 we observe that this reduction is not that distinctive for small amounts of involved clients. However, as mentioned before and as these curves indicate, the performance as well as the gain of location awareness increases with growing number of peers and might become substantial in extremely large networks. This assumption is further backed up by the results of [59] who state that location awareness can lead to 50% faster downloads for approximately 24% of peers.

Potential discrimination of isolated clients

Concluding on the performance of location awareness

After the studies presented in this section it can be concluded that SatTorrent-M has a great potential to reduce unicast messages while at the same time having only a negligible demand for broadcast messages. The protocol can be tuned towards maximum message saving or minimal distribution time as well as to a

Summarizing section results

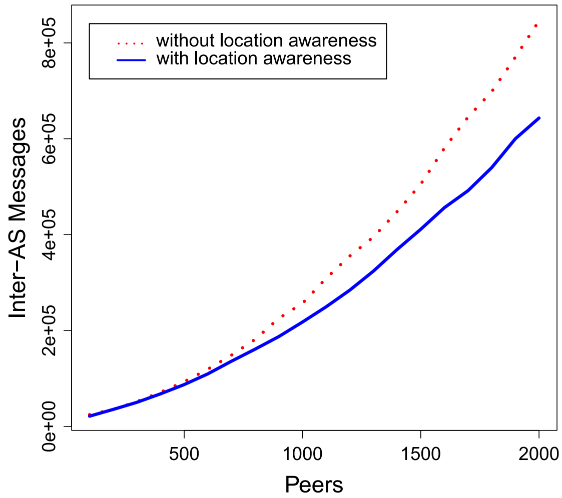

Figure 7.6: Inter-AS Messages with and without location awareness.

trade-off between both.

7.3 SatTorrent with Payload Broadcasting

SatTorrent combines payload and metadata broadcasts

The full SatTorrent protocol that is presented in this section combines metadata and payload broadcasts. It fully integrates SatTorrent-M as it has been described in the previous section. In principle it would also be possible to realize the payload broadcast without doing any metadata broadcasts. However, the tracker still needed the additional information provided in SatTorrents announce messages (see table 2.2), needed to process and to store the corresponding data. Thus the savings that could be expected from excluding this aspect were negligible. Further, since BTRMs are only sent occasionally and due to the positive results we have seen in the preceding section, it would make no sense to renounce this aspect and the resulting benefits.

Modification of tracker an clients required

As discussed in chapter 4 during the introduction of the content distribution model, the broadcasts are initiated on the content provider side, more precisely by the tracker. In order to facilitate this extension an upgrading of the tracker's functionality is mandatory. Further the clients must be modified to handle the broadcasted pieces correctly. All these changes are described in the following section before we start the evaluation of the protocols performance based on simulation later in section 7.3.2.

7.3.1 The SatTorrent Protocol – Payload Extension

In order to explain and to understand how the tracker determines what pieces of which files should be broadcasted we will have a closer look on how it stores information about peers and their downloads. Thereby identifiers will be defined which facilitate the specification of rules and of an algorithm that describe the broadcast decision making process. First, let the number of files managed by a specific tracker be N and the corresponding set of files F be defined as

$$F = \{f : f \text{ is handled by this tracker }\} \tag{7.2}$$

Each file $f \in F$ consists of a specific number of pieces so that

$$f = \{c : c \text{ is a piece of } f\} \tag{7.3}$$

Further for each $f \in F$ the corresponding tracker holds a list of peers which is defined as

$$P^f = \{p : p \text{ is currently downloading } f \in F\}. \tag{7.4}$$

Tracker maintains extended peer information records

Further, for each peer known by the tracker, the latter stores a vector V^p containing the necessary information about peer p. This information includes the peer's IP address, the port it is listening on and its ID, the amount of bytes uploaded, downloaded and still missing in order to complete the download. So far, this is identical to the information that the BitTorrent tracker stores. As we know from the preceding section, for a SatTorrent tracker each V^p contains additional information about the corresponding peer, including location, whether or not the peer is sat-enabled and the bitfields for all active and inactive downloads. In this context we define the following function:

$$s(p) = \begin{cases} 1, & \text{if peer } p \text{ is sat-enabled} \\ 0, & \text{otherwise} \end{cases} \tag{7.5}$$

Let further be B_p^f the bitfield of peer p for file f. We define the function

$$b(p, f, c) = \begin{cases} 1, & \text{if } p \in P^f \wedge p \text{ has piece } c \in f \\ 0, & \text{otherwise} \end{cases} \tag{7.6}$$

Now we are able to express the broadcast decision metric of a tracker. A broadcast for piece c of file f is triggered only if

$$\sum_{p=0}^{|P^f|-1} b(p_i, f, c) \cdot s(p_i) \geq BT \tag{7.7}$$

Factors influencing the broadcast threshold

Thereby BT is the broadcast threshold introduced in chapter 4.2. It denotes the required number of peers that potentially, according to the tracker's information, have a demand for the specific piece. The exact value of BT depends on numerous parameters such as for example the available satellite bandwidth, the cost comparison between Internet delivery and broadcast and whether there is a competitive situation of several broadcast attempts that can not all be served

Algorithm 1 Determining piece broadcasts

for all *files* $f \in F$ **do**
  for all *pieces* $c \in f$ **do**
    $counter \leftarrow 0$
    for all *peers* $p \in P^f$ **do**
      if $b(p, f, c) == 0 \ \& \ s(p) == 1$ **then**
        $counter \leftarrow counter + 1$
      end if
    end for
    if $counter \geq BT$ **then**
      broadcast piece c
    end if
  end for
end for

within an acceptable time frame. As depicted in chapter 4 narrowing down this threshold for a specific scenario is not subject of this work. Instead in the following section the performance of SatTorrent will be evaluated under varying values of BT. The procedure that a tracker periodically executes in order to determine the pieces that are suitable for a broadcast is illustrated as pseudo code in algorithm 1. This algorithm is mainly based on equation 7.7. For each piece of every file the tracker counts the number of peers that are downloading the corresponding file but still do not possess that piece. After this has been evaluated for all files, those pieces where the number of demanding peers exceeds BT are passed to the uplink station for broadcast.

Determining broadcastability on piece level

Identifying broadcasted pieces

In principle each of the broadcasted pieces is received by all sat-enabled clients. When pieces are exchanged via peer wire protocol between peers, the TCP connection facilitates the correlation between piece and file. This is not the case for broadcasted pieces. Thus in order to determine what file an incoming piece belongs to, a broadcast piece message must contain an identifier of the corresponding file. For that purpose the info hash also used in tracker announce messages is being used which leads to the specification of a **broadcast piece message** (BPM) shown in table 7.5.

Client strategies for processing broadcasted pieces

In order to make the broadcast more efficient the message overhead is reduced as far as possible. In consequence a BPM contains a list of pieces rather than one single piece. A client can use different strategies for handling incoming BPMs. The situation for pieces belonging to files that are currently downloaded is straightforward. This message is processed and the contained pieces are stored, the local data structures updated accordingly (e.g. bytes downloaded, missing, bitfield). Further it must be checked whether any PWMs of

field name	description
message_id	4 bit field indicating the type of the broadcast message, *0100* for BPM
info_hash	20 byte hash value of the info part in the torrent file that identifies the file
piecelist	a dictionary with the following info for each piece:
piece_id	an identifier for the piece
piece_data	the actual payload containing the piece's data

Table 7.5: BPM specification

type *interested* are now obsolete and thus must be revoked by means of an *uninterested message* to the corresponding peers. Also it is possible that this specific piece currently downloaded from another peer. In that case this download should be canceled immediately. Since the client is unchoked by that peer, another piece must be requested instantaneously if applicable in order to make use of this situation in terms of downloading as many pieces as possible before being choked again. The remaining procedure is equivalent to the standard routine when a piece download is completed.

Handling of unrequested pieces

Next the question is considered how clients should handle BPMs for files that are not part of an ongoing download. Possible approaches are to discard the message, to generally store the contained pieces for all BPMs for potential future demand or to store only those pieces belonging to files that have a high probability to be subject of prospective needs. In case simply everything is stored in the cache, files are replaced by means of a least recently added strategy in case of insufficient cache capacity. In case of taking future demand probabilities into consideration additional information and effort is needed in order to make the corresponding decision. In this chapter respectively in the evaluation of SatTorrent in the subsequent section such a prediction based approach will not be considered. The reason is that for the meantime only the distribution of one single file is analyzed and thus such strategies are irrelevant in this context. The subject will be revisited in the next chapter when social awareness is added to the protocol. Then the clients will be able to choose their downloads from larger repositories of files.

Caching requires prediction of future demand

Piece selection strategy

Another important aspect for all P2P file exchange protocols is the piece selection strategy. It specifies the way clients decide on which of the missing pieces should be downloaded first from a peer. In the official BitTorrent protocol specification a *random piece selection* is proposed [29]. However, for many application scenarios the commonly used *rarest first* approach is the preferable choice, since it promotes an equal piece replication rate and thus improves the piece availability as it is shown in [74]. Other application areas might need specialized strategies. For example when P2P distribution is used for live video streaming the clients must receive the pieces before the corresponding playback position is reached. In case of small buffers and a download bandwidth only marginally above the video encoding bitrate, this suggest a sequential piece download order. Corresponding alternative piece selection strategies have been designed and investigated by [107] and [43]. As these examples show there is not one optimal solution to this problem, rather every application has its own, specific demand. For SatTorrent in principle a sequential order would be the best choice in order to exceed BT as early and often as possible. However, not all participating clients can be expected to be sat-enabled. Further it can hardly be judged by a client whether or not pieces of a downloaded file might be broadcasted in the future. In case no broadcast occurs, a sequential piece download would constrain the replication rate and thus slow down the distribution process. In the evaluation presented in the following section we will stick to a random selection. Since the objective is primarily to compare the performance of SatTorrent and BitTorrent with each other it is better than using a strategy that would exclusively promote one of the protocols.

Application specific piece selection strategies

7.3.2 Evaluation of SatTorrent

In this section we will evaluate SatTorrent's performance under varying conditions in comparison to BitTorrent. This will provide knowledge on how parameters should be adjusted to obtain the best attainable results for a specific purpose. Therefore the simulator used in section 7.2 has been extended in order to comply with the SatTorrent protocol with payload broadcasts. For the purpose of ensuring reproducibility of the achieved results, each simulation is unambiguously specified by a configuration file.

As described in section 7.2 the scalability of the simulator is mostly limited by its memory consumption. In order to cope with this problem the simulator must employ a certain level of abstraction to limit the simulation's complexity and its memory demand. In this regard in the simulation only full pieces are exchanged between peers instead of transferring numerous blocks for each piece. This means in the simulation only one PWM of type *piece* is sent where in reality several would have been needed. That becomes important at this point since we are about to compare the traffic caused by PWMs—and *piece messages* in particular—in the unicast and in the broadcast network. For a better distinction between real world *piece messages* carrying blocks and those in the simulator, we refer to the former as *block messages*. The rate in which *piece messages* are being sent in the simulation has been adjusted in order to reflect this merging. However, the simulator only counts the number of *piece messages* that are exchanged. Thus for the analysis of the results and a comparison of

Increasing simulation performance by means of astraction

Name	Description	Size
id	the message type id number	32 bit
index	zero based piece index	32 bit
begin	zero based offset within the piece	8 bit

Table 7.6: Block Message Overhead

Overhead of unicast block messages

bandwidth consumption it must be considered that each *piece message* in the unicast network implies several *block messages*. In this context it is necessary to know the overhead of the latter. The common block size used in most current BitTorrent client implementations is 16 kByte per block [102]. Considering the prevalent piece size of 512 kByte—which is also used by the simulator—this results in 32 blocks per piece. As it can be seen in table 7.6 the overhead per block message is 72 bit or 9 Byte. This means each *piece message* in the simulation generates a network traffic of 512 kByte for the payload plus 32 times 9 Byte for the overhead equals 524,576 Byte. The results presented in this section are adapted accordingly.

Global parameters used for SatTorrent's analysis

As far as nothing else is specified, the parameter settings introduced in the following remain unchanged for all simulation runs. First to mention is the satellite transponder bandwidth, which is set to 36 Mbit. As discussed in chapter 2.3 this is a bandwidth that even older communication satellites used for analogous television broadcasts are capable of after deducting error correction (net bandwidth). One of these transponders is available for the broadcasts. Further, as already mentioned above, a piece size of 512 kByte is used. According to [102] this is a common value for BitTorrent and represents a good trade off between torrent-file size and efficiency for that protocol. In order to provide a fair comparison between BitTorrent and SatTorrent, respectively the solely unicast delivery and a satellite broadcast aided approach, it is vitally important to apply the best possible settings for BitTorrent. The tracker announce interval is set to 300 seconds, the broadcast interval for BTRMs to 20 seconds. The client nodes as well as the original seeder are distributed to 12 autonomous systems. The TBCT has initially been set to 20, a comparison of the protocol performance when using other values follows later in this section. Further the open connection limit (OCL) has been set to 80, which is the value also commonly used for BitTorrent clients. The maximum number of peers that can join the network per step is set to 30.

Providing an equitable protocol comparison

As a first performance evaluation the distribution time and the aggregated bandwidth demand for SatTorrent and BitTorrent are compared. The latter is

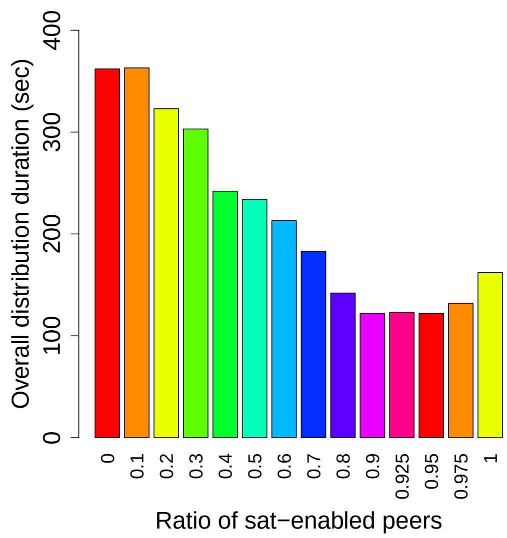

Figure 7.7: Distribution time under different proportions of sat-enabled peers

Comparing distribution time and bandwidth demand

equivalent to SatTorrent without any sat-enabled peers. Thus whenever we have $SPR = 0.0$ in the charts presented in this sections, this means that these are the results for BitTorrent. As mentioned before, it can hardly be expected to have 100% of sat-enabled peers in a real world scenario since not all households are willing or allowed to install a satellite dish. Plus, according to results of SatTorrent-M this seems not even desirable. Here we analyze the distribution of a file of 20MB size among 1000 peers with $BT = 150$ under varying values for SPR. Figure 7.7 shows the distribution time for SPR ranging from 0.0 to 1.0. Due to the change in the progression of distribution time that has been observed for SatTorrent-M in the range of SPR from 0.9 to 1.0 a more fine grained stepping has been applied within this interval. Within the range from 10% to 90% a reduction of the distribution time for increasing ratios of sat-enabled peers can be seen. However, between 90 and 95% it remains more or less on the same level until it starts increasing again beyond 95%. Further there is a very small increase in the step from $SPR = 0$ to $SPR = 0.1$.

Reduced gain under extreme conditions

It can be observed that the distribution time becomes negatively influenced if one of the two groups—sat-peers or others—becomes too small. In case an overwhelming majority of clients is sat-enabled (more than 95%), it is more likely that the increased distribution time is driven by the restrictive behavior of sat-peers that prioritizes those peers that are most supportive. Now we have a situation where all sat-enabled peers receive the broadcasted pieces at the same time, which narrows down their missing pieces homogeneously. Since these clients concurrently still download pieces from peers via unicast, the probability that a sat-enabled peer will receive a piece also needed by another sat-peer is relatively high. In contrast, non sat-enabled clients might download pieces first that the sat-peers already received via broadcast. This would even be enforced if a rarest first piece selection strategy was applied. On the one hand the previously rarest pieces are those who become well distributed after the broadcast. On the other hand non sat-enabled clients only have knowledge about the bitfields of a small fraction of all peers. In consequence, they might not be aware of the changed global situation and thus continue to concentrate on the alleged rarest pieces. Due to the broad availability of these pieces, getting them became easy. Thus the probability for requesting and getting a piece that was previously broadcasted is even increased for non sat-enabled clients. And indeed during the simulation it could be observed that in these cases only a few peers (less than 0.2%) were causative for the increased total duration. These have still been downloading while all other peers had finished. None of these clients was sat-enabled.

Impact of piece selection on this behavior

Regarding the slight increase of distribution time when the SPR changes from 0.0 to 0.1, another reason for this behavior can be derived from the results presented in figure 7.8. There the consumed satellite bandwidth for different

Number of sat-peers should at least exceed BT

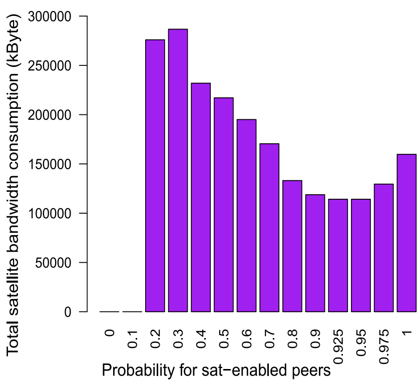

Figure 7.8: Satellite bandwidth consumption for varying SPR

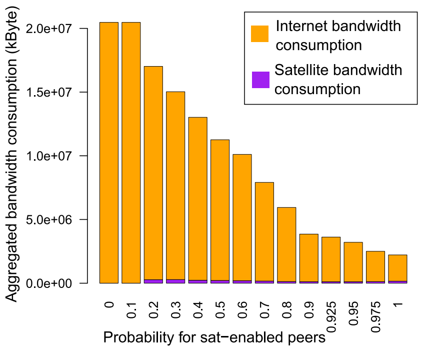

Figure 7.9: Aggregated bandwidth consumption for varying SPR

proportions of sat-peers is illustrated. If SPR is set to 0.0, in other words for BitTorrent, there is obviously nothing broadcasted. This remains even if SPR equals 0.1 since the amount of sat-enabled peers (100 in this case) prohibits that the condition given in equation 7.7 is ever satisfied, as we are using a BT of 150. Going back to the observed increase in distribution time at SPR set to 0.1 in figure 7.7, this means that the sat-enabled peers suffer from the aforementioned delays of satellite broadcasts while the major corresponding benefit does not occur. That is the increased download speed due to broadcasted file pieces. At the same time, there are too little sat-enabled peers in order to take a considerable advantage of the metadata broadcasts. It is worth noting that the difference in distribution time is only one second and thus probably not that meaningful. Nevertheless, we will evaluate better client and tracker parameters in order to achieve better results later in this section.

Significant reduction of aggregated network traffic

Having a look at figure 7.9 we observe the most significant advantages of payload broadcasts and thus of the proposed content distribution model. The chart visualizes the aggregated bandwidth consumption for broadcast and unicast piece messages. What can be observed is a monotonic decrease of total bandwidth demand for increasing values of SPR. At the same time the fraction of satellite bandwidth is considerably small. Accordingly there is a great potential for reducing the unicast network traffic by more than 89%. The stagnation at the beginning is self-evident since a perceptible traffic reduction cannot be expected as long as not payload broadcasts occur (see figure 7.8). As soon as piece messages start to be distributed via satellite the Internet traffic experience a nearly linear decrease with a mean gradient of approximately $-35818, 75$. The described effect holds up to SPR set to 1.0. The unicast traffic reduction is not affected by the longer duration of the distribution process. We recall that the by far biggest messages are those carrying file pieces. SatTorrent has no measures to reduce the piece messages that must be sent to non sat-enabled peers since they can only receive the corresponding data via unicast. Thus savings are made mostly at the sat-peers. Since, as stated above, the peers that cause the increased time demand are not sat-enabled the traffic reduction is not influenced by their longer lasting downloads.

Traffic decreases monotinically with increasing SPR

Optimizing SatTorrent parameters

Next the previously recognized issues concerning the increased distribution time for extreme values of SPR will be addressed. This includes an evaluation of how this behavior can be avoided. An easy approach to do this would be to generally remove the enhanced selection process for sat-peers and to change the way the tracker is handling get-requests from sat-enabled peers. In particular this would mean that in any case a unicast response would be sent also to sat-enabled clients in addition to the BTRMs and that clients treat all peers equally. This results in an increasing number of messages and would revert many of the advantages of SatTorrent which is not desirable. In the following an optimization of the results is being approached by means of minor changes to the parameters. The simulations have been repeated with the modified settings. In the following the corresponding results are being presented, showing that an increased performance has been achieved.

Introducing the new settings

The limit for open connections has been changed for sat-enabled peers. This is based on the assumption that the same performance can be achieved with SatTorrent even with a smaller number of connected peers. Thus an OCL of 40 has been used for sat-peers, while it remained unchanged (80) for the others. The TBCT has been raised from 20 to 100. This shall further reduce the impact of the delay in tracker responses that sat-peers experience and that can potentially increase the overall distribution time. At the same time, it is still small enough to keep the amount of additional unicast messages in reasonable bounds. Also the value for f_{conprob} has been raised to 1.0 in order to relax the restrictions against potentially less useful peers. It is worth noting that this still means that priority is given to the best qualified peers. The factor just changes the strength of this effect (see equation 7.1). The new configuration of parameters is referred to as *alternative parameter set 1* (APS1). In figure 7.10 we see the

Less restrictive connection making

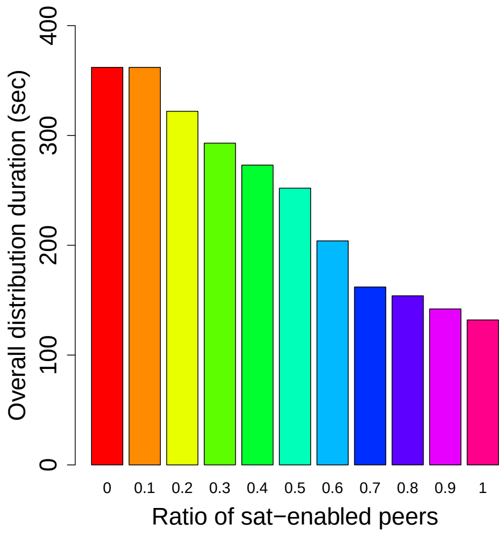

Figure 7.10: Total distribution time for varying SPR with APS1

resulting distribution times for different values of SPR. The small increase at $SPR = 0.1$ disappeared in consequence of the adjusted TBCT. Further there is no more increase for very high values, in particular for $SPR = 1.0$. This shows that the assumption about the causes for this effect and the proposed measures are correct. The corresponding demands for satellite and Internet capacity are illustrated in figures 7.11 and 7.12. Regarding the Internet traffic we have the same behavior as documented before. While minor changes can be observed for the satellite usage, they are so small that they are not significant. In particular since they are alternating which means that there is no obvious advantage for one of the settings.

Issues with increasing distribution time solved

In another approach a different combination of parameters is used which is referred to as APS2. The download bandwidth for all clients is increased from 6 to 8 Mbit, all peers join the network in the first step of the simulation instead of arriving successively. Further the number of available satellite transponders is increased from 1 to 4 and the TBTC has been reset to 20 again. The con-

Evaluation with changed client parameters

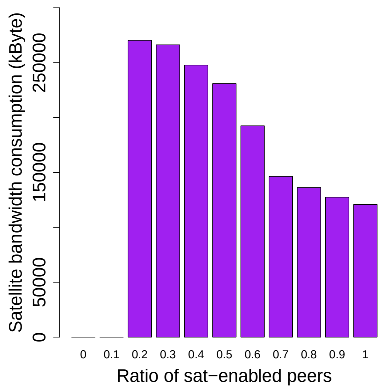

Figure 7.11: Satellite bandwidth used for varying SPR with APS1

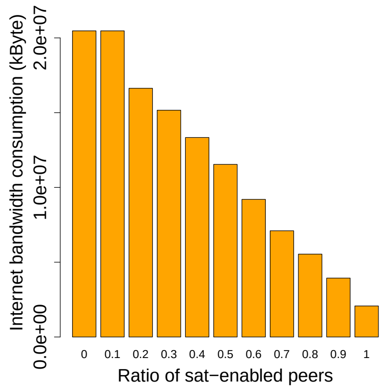

Figure 7.12: Internet bandwidth used for varying SPR with APS1

sequence of the latter can be observed in figure 7.13. Although it is shifted

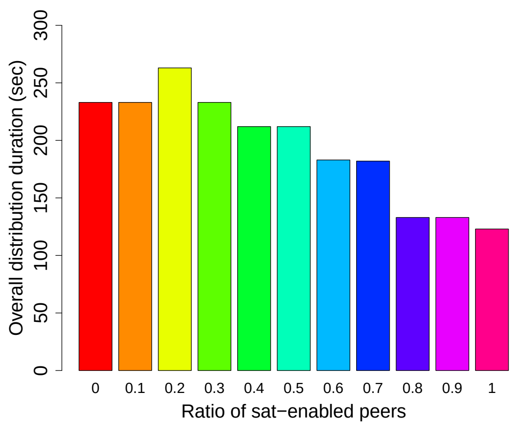

Figure 7.13: Distribution time for varying SPR with APS2

Impact of increased client bandwidth

towards $SPR = 0.2$ there is a visible increase in the distribution time for the small number of sat-enabled peers. The shift is a result of the changed peer join behavior and the increased download bandwidth. The latter is further causative for the smaller amplitude of duration changes. The higher bandwidth allows an increased download speed for the pieces exchanged via unicast network, even without changing the upload bandwidth. The satellite capacity utilization illustrated in figure 7.14 is slightly lower than in the previous evaluations. This has two reasons. On the one hand the higher Internet access bandwidth lets pieces be faster distributed via unicast, thus some broadcasts are substituted by that. Further, the fact that all peers join at the very beginning of the simulation allow

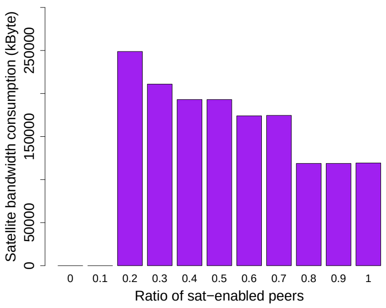

Figure 7.14: Satellite bandwidth used for varying SPR with APS2

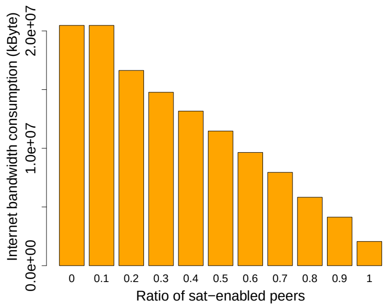

Figure 7.15: Internet bandwidth used for varying SPR with APS2

more clients to be reached with one download. In contrast to that, with subsequently joining peers the firsts broadcast might have already taken place before all peers arrived. Thus those who missed this broadcast might trigger another one. In the Internet traffic chart (figure 7.15) no mentionable differences to the first approaches can be observed.

Direct comparison of all results

In order to allow a better comparison of the different results they have been merged in one chart for each category. This is shown in figures 7.16, 7.17 and 7.18. Most noteworthy here is, as mentioned before, the fact that the same ex-

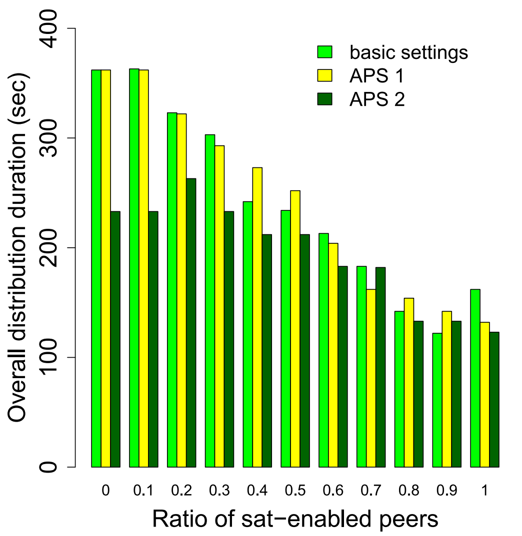

Figure 7.16: Distribution time under varying SPR for three different parameter configurations

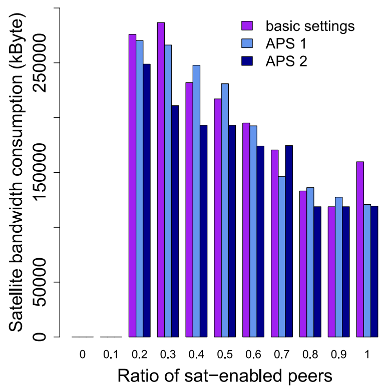

Figure 7.17: Satellite bandwidth consumption for varying SPR

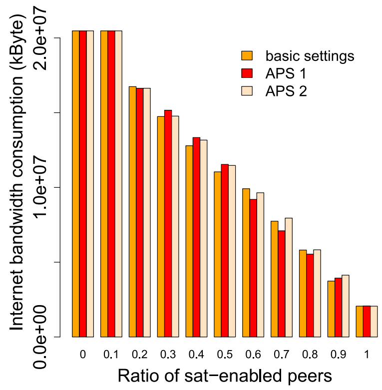

Figure 7.18: Internet bandwidth consumption for varying SPR

cellent results for Internet traffic reduction could be obtained with all settings. In another set of simulations this has been analyzed for a network consisting of 2000 nodes. The rate at which increasing numbers of sat-enabled peers are able to reduce Internet traffic is even higher. At the same time, the mean satellite capacity demand is lower than in the simulations with 1000 peers that have been presented above. This confirms the aforementioned expectation that SatTorrent performance will ever increase with a growing network size and thus it represents a scalable solution for content distribution.

SST under varying network sizes

In order to further augment our understanding of how SatTorrent performs for different network sizes the same file distribution using the standard settings introduced in the beginning of this section has been analyzed for varying numbers of peers. In figures 7.19 and 7.20 we see the bandwidth consumption for different network sizes in the Internet and in the satellite network respectively. The behavior with respect to the unicast messages (figure 7.19) is straightforward and satisfies our expectations: The bandwidth demand for payload delivery is continuously decreasing with the proportion of sat-enabled peers. The horizontal line segments in the plot reveal the only condition: The total number of sat-enabled peers must at least be equal or greater than the broadcast threshold, which is not surprising at all since otherwise broadcasts are not possible.

Presentation of simulation results

Even though the curves in figure 7.20 all exhibit a similar shape, their progression is by far less self-evident than the ones for Internet bandwidth consumption and takes a deeper investigation. First we observe a steep increase immediately after the SPR reaches a value that allows the number of sat-enabled peer to exceed the broadcast threshold BT. The fact that we observe a peak at $SPR = 0.3$

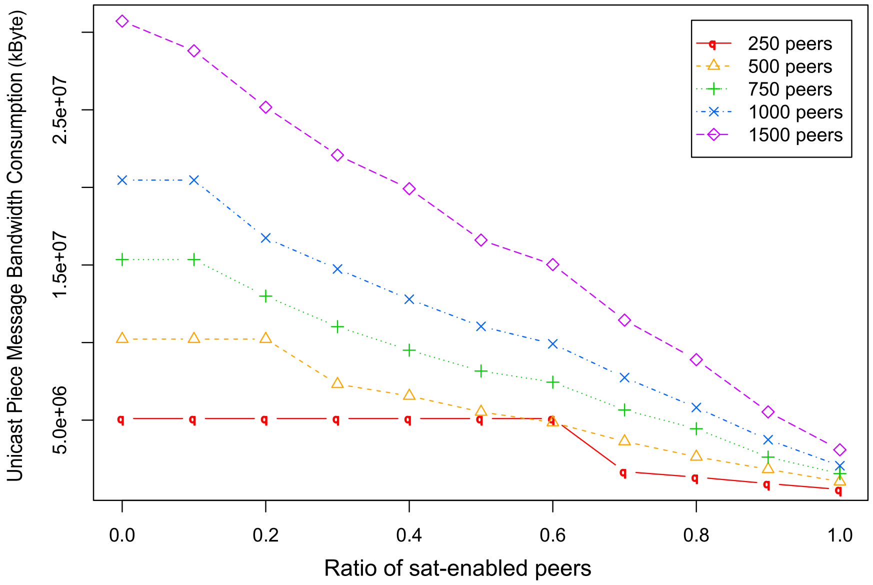

Figure 7.19: Internet bandwidth consumption for different values of SPR

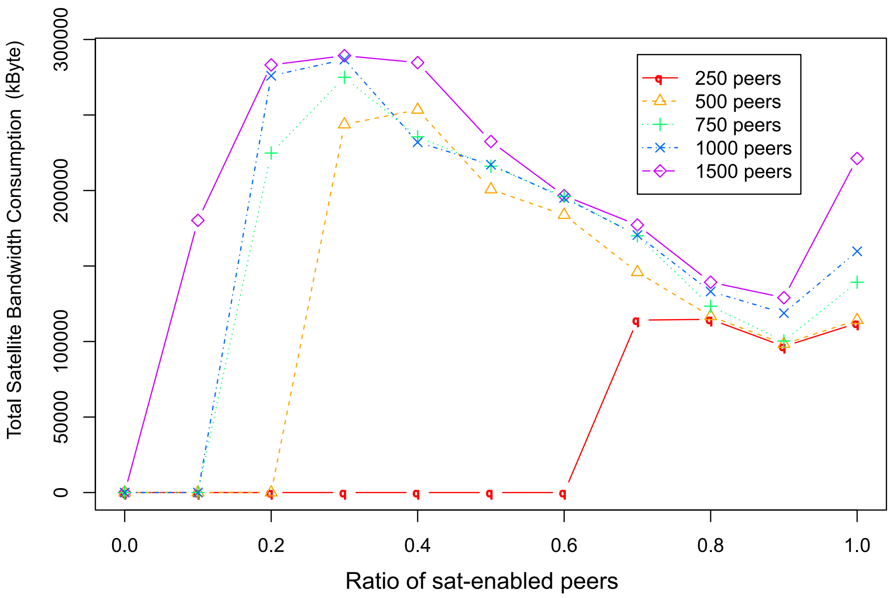

Figure 7.20: Satellite bandwidth consumption for different values of SPR

Interpretation of simualtion results

for network sizes of 750, 1000 and 1500 peers indicates that this high demand for satellite bandwidth is not depending on the absolute number of sat-enabled peers in relation to BT. The reason seems to be in the ratio between sat-enabled and non-sat-enabled peers. On the one hand peers that are able receive pieces by broadcast do not wait idle for satellite transmissions but continuously exchange pieces with other peers in the network. On the other hand, the piece selection strategy leads to a uniform distribution of pieces at the peers. In consequence the condition given in equation 7.7 might in many cases only be scarcely fulfilled even though the number of sat-enabled peers is twice or thrice as much as BT. With an increasing density of sat-enabled peers the probability that each piece broadcast is received by a larger number of peers that still need that specific part of the file grows, and thus the efficiency of the broadcast increases. This results in the decreasing curves which can be found after the peak.

Confirmed the issues when all peers are sat-enabled

This is true except for situations where all peers are sat-enabled. The main reason for this is the complete absence of non sat-enabled peers which are less restrictive in their behavior. The restrictive behavior of sat-enabled peers has been discussed above. In case f_{conprob} is set to a low value, this leads to an increased time demand until the data has been completely delivered to all peers as we have already observed in figure 7.7. The positive aspect, which is not reflected in this figure, is that it also further reduces the unicast traffic. However, there are two aspects that mitigate this effect. The first is that, as discussed earlier, in a real life environment we will hardly ever reach a configuration where all peers are sat-enabled. Second, we have seen in the optimization approaches above, particularly in figures 7.12 and 7.15, that small changes in the settings can eliminate this increase for $SPR = 1.0$ without negatively influencing the other results.

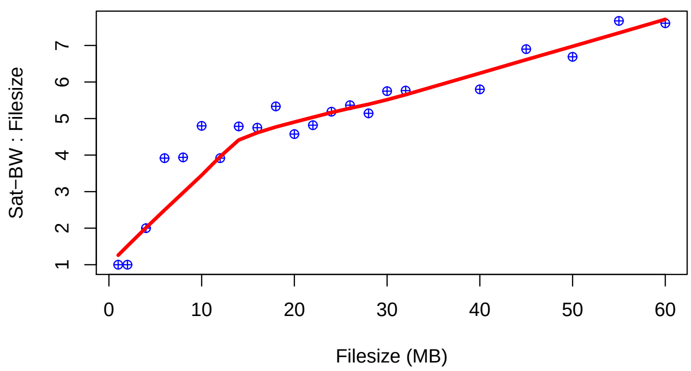

Figure 7.21: Satellite bandwidth consumption relative to filesize

Next the impact of the file size on the performance of SatTorrent is investigated. For the corresponding simulations the results presented here rely on a distribution among 500 peers has been used with BT set to 150 and SPR to 0.9. The latter has been chosen since it has shown to be a good choice with respect to the time demand for the file distribution as well as concerning the satellite bandwidth consumption. In general SatTorrent's performance can be expected to increase with the size of the files being distributed since that also augments the probability to have many concurrent downloads. In consequence each broadcast can be received by a potentially higher number of peers which constitutes an elevated efficiency. By applying parameters that are nearly optimal also for smaller files we reduce the influence of other factors except file sizes in this measurement. The results can be seen in figures 7.21 and 7.22.

Impact of file size on SST performance

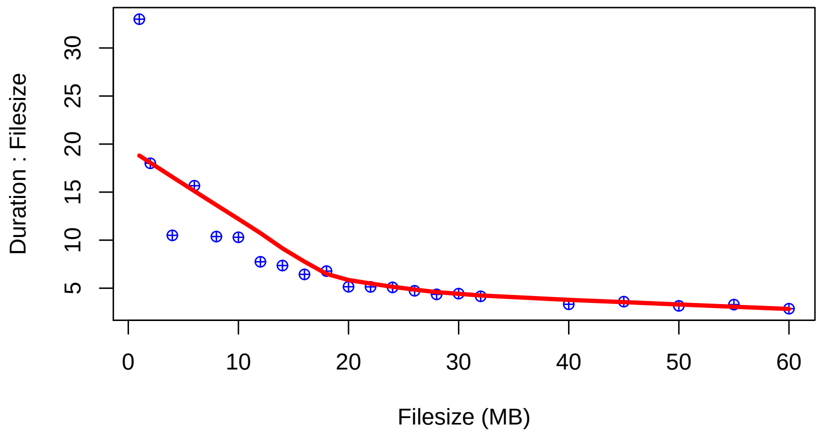

Figure 7.22: Overall delivery duration in relation to filesize

The correlation between file size and the relative satellite bandwidth demand becomes obvious in figure 7.21. The decreasing gradient that we find here confirms our assumption of an increasing SatTorrent efficiency for larger files. The same applies for the relative distribution time that is decreasing with growing file sizes (see figure 7.22).

At last SatTorrent has been analyzed under varying values of BT. The expectation that altering this parameter would have a significant effect on the distribution time and on the satellite bandwidth demand has not been confirmed. The latter is displayed in figure 7.23. The reduction in bandwidth demand is marginal. The outlier on the very right is irrelevant since here BT was higher than the maximum number of sat-enabled peers. Thus is was simply impossible

Analyzing SST under different BT values to have any broadcasts and consequently no bandwidth has been allocated.

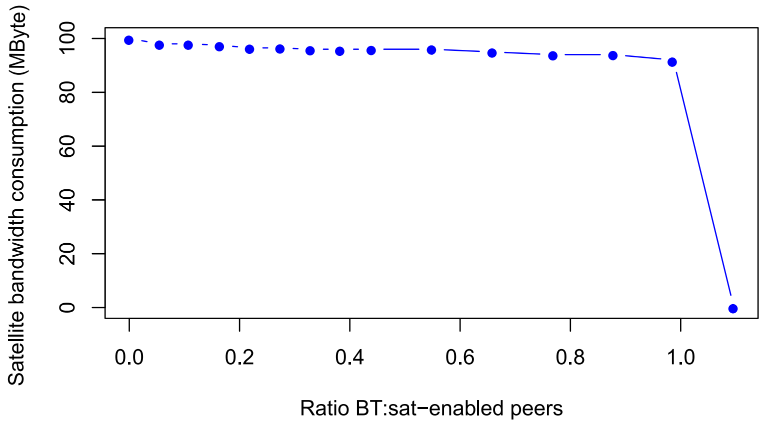

Figure 7.23: Satellite bandwidth demand for relative BT values

Chapter conclusion

Summarizing it can be stated that SatTorrent with payload broadcasts satisfies the expectations of significantly reducing the unicast network traffic while on the other hand only demanding a relative small amount of satellite capacity. Evidence for the scalability of the content distribution model has been provided in this chapter. Further it has been shown that the distribution of files with SatTorrent is significantly faster than the conventional approach of using BitTorrent even if the number ob sat-enabled peers is very small. Last but not least the results presented above reveal that SatTorrent, due to its comprehensive set of adjustable parameters, has a great flexibility that allows it to be optimized for various usage scenarios.

CHAPTER 8

Social SatTorrent

Social SatTorrent implmenets remaining model aspects

In the preceding chapter the SatTorrent protocol has been introduced and analyzed with respect to its performance. However, several important aspects of the content distribution model as it has been proposed in chapter 4 have not been considered in the evaluations so far. These are basically caching of broadcasted data for potential future demand, prefetching and especially everything related to the social augmentation. As is has been previously motivated each client is equipped with a large persistent storage that is used as a cache. Its primary purpose is to have as much data as possible cached already prior to the actual demand. Thereby the user client, more precisely the HGW, is responsible for the caching. The first component needed in order to facilitates this is a user profile. Thereby it is irrelevant whether this is explicitly defined by each user or implicitly created and maintained by the home gateway. The existence of such a user profile already facilitate in-advance caching on a very basic level for certain types of content. An example are subscriptions, e.g. for software updates or online video channels. These can also be either explicit or implicit. If we imagine that a user has Microsoft Windows 7 installed on his PC as well as the World of Warcraft (WoW) client. Then it can be assumed that he will install almost all of the recommended operating system patches and also all WoW client updates. In these cases the "prediction" is very straightforward. Other assumptions of future demand can be made by means of personal preferences provided by the profile. If we imagine a user likes the actor Clint Eastwood and also western movies, caching "A fistful of dollars" will probably result in a cache hit. However, this way of prediction is by far more difficult and error prone than the first mentioned. And there are others that can even be

Profiles enable caching measures

Measures to detect and handle traffic caused by memes

considered as being nearly impossible. Internet memes fall into this category, experiencing an outstanding popularity often without any identifiable rational cause. In that case an a priori prediction of the incident itself based on local information is a goal which is very difficult to achieve. Having complete knowledge about the development of content request might allow a prognosis after monitoring the process of rapidly increasing popularity for a while. However, it that case much time is lost before a reaction to this event is possible. The key to an earlier detection of meme-like behavior is analyzing online social networks. A sudden occurrence of link sharing for a specific piece of content can be used as an indicator for a emerging meme. But there are also other ways of capitalizing social network structures for an estimation of future demand of individual users. For example, if several friends of a specific user have already watched a certain video, potentially also gave it a high rating or shared it in an OSN, there is a significantly increased probability that this user will also access this video, as the results of [18] indicate. Thus for the realization of prefetching and caching functions for SatTorrent the use of social networks is self-evident. As pointed out in chapter 4 the HGW, in this case the SatTorrent client, does not necessarily need to implement the online social network. It can be assumed that the client has knowledge about the immediate neighborhood in the social graph of each registered user. Between these direct neighbors, in OSNs often referred to as friends or buddies, an increased level of trust can be assumed. This property allows sharing of potentially private information among close social ties that users might not disclose to other, less trusted individuals. According to that, access to these contacts preferences is postulated. This further allows each sat-enabled client not only to cache broadcasted content matching the demand of connected users, but also for their close social contacts. This amplifies the benefit that non sat-peers gain from SatTorrent. The results presented in this chapter have been partially published in [68]. In the next section the corresponding extensions to the SatTorrent protocol are specified, which turn it into **Social SatTorrent (SST)**. This is followed by an evaluation of the resulting performance assets in section 8.3.

Making use of increased trust among close social ties

8.1 Social SatTorrent Protocol

In the following the protocol extensions that distinguish SST from SatTorrent are introduced prior to their evaluation in the section 8.3

8.1.1 Persistent global upload rewarding

When investigating the causes of BitTorrent's great success, one of the most essential facets is the tit-for-tat policy. It ensures that clients upload pieces preferentially to altruistic peers while penalizing free riders. The latter have been a major vulnerability in previous P2P protocols, as documented in [60], [5], [88], [61]. However, in BitTorrent this rating of peers' cooperativeness is always done on a per file basis. This means if peers A and B are both downloading the files f_1 and f_2, and A solely uploaded 50% of f_1's data to B, B would still punish A for not uploading data of f_2. The consequence is that there is little incentive for clients to keep seeding a file after the download has been completed since there is no benefit from that. Even worse, uploading pieces of the finished file will reduce the available upload capacity for other, incomplete downloads and thus have a negative impact on their performance. Considering that most P2P protocols aim at preserving the users anonymity as much as possible, a broader approach that would deliver more fairness was simply not possible for BitTorrent. However, SST has a different objective. It aims at maximizing performance for downloads from authorized sources only and implies persistent user profiles. This facilitates the provision of a global rewarding scheme based on credits. At the same time a much more strict tit-for-tat policy is applied, that always requires a mutual contribution from two exchange partners. Thus when a client wants to receive data from a peer, it either must in turn send other data required by that peer or give credits. Every new registered user starts with zero credits. However, since these new participants neither have pieces to share nor any credits, all users are allowed to have a certain amount of negative credits in order to avoid ostracism of new participants. Let the corresponding threshold for negative credits be m_C. This concept facilitates another element of SatTorrent, the donation of credits to other peers which is introduced in the following section. How user credits are managed, where they should be stored as well as measures against manipulation must be considered for an operational implementation. However, for the scope of this thesis and the corresponding simulations it will be assumed that the credits are stored on the HGW in the user profile, under the presumption that all users are trustworthy. For a real world implementation however, this data must be protected from fraud.

Limitations and performance issues of tit-for-tat

The global credit system implemented by SST

8.1.2 Social Aid in SST

The idea of buddies supporting each other in their download attempts by means of donating idle capacities, mainly bandwidth, is utilized by the 2Fast protocol, as described in chapter 1.2. The motivation for this approach is the observation that often clients can not saturate their download link capacity in BitTorrent networks. The reason is, according to [42], the often asymmetric Internet connection bandwidth in combination with the tit-for-tat policy. Due to the increased download speed that has been reached with the 2Fast protocol as reported in [42], SatTorrent takes up this concept and adapts it to the special characteristics of hybrid networks. In particular this means that sat-enabled

Download support in the 2Fast protocol

Friendly aid in SST

priority	task	peer set
1.	request piece	friends who are downloading the file
2.	request piece	non-friends who are downloading the file
3.	request piece	friends who are seeding the file
4.	request piece	non-friends who are seeding the file
5.	request support	sat-enabled friends
6.	request support	non sat-enabled friends

Table 8.1: Piece source priority list for SST

peers cache broadcasted files that are downloaded by their friends even if they should not match their own preferences. Subsequently they are able to support these social contacts in their downloads. In contrast to 2Fast, in this case they can allocate their full upload capacity for sending pieces to the supported peer. Consequently, in the best case SST only needs half the helpers that 2Fast needs to saturate the Internet connection downlink capacity of the assisted peer. In case the required file has not been broadcasted before, the friends will start downloading it on their own and forward the received pieces to the client that asked for support. In this case additional unicast traffic is produced. Therefore this friendly aid is only applied in case all other measures did not result in a full utilization of the downlink capacity.

Due to the knowledge of sat-enabled clients about their social environment and the bitfields of all peers active in the same torrent (downloading or seeding) they can use the following priority for requesting pieces: Thereby sat-peers who have cached previously broadcasted pieces of a file they are not actively downloading belong to class 3. Each client first attempts to get pieces from peers in priority class 1, then 2 and so on. This commences until either its download capacity is completely allocated or no peers are left. The sets of clients defined by these classes are disjunct. Thereby the assignment follows the same priority. Thus a downloading sat-enabled friend will be in class 1 and not in 5. The priorities are ordered for best overall distribution performance. By asking downloaders first, an exchange of pieces is promoted and thus a better balance of credits preserved. The reason why always the friends are asked first is twofold. On one hand, according to [94] and [22], by fostering the exchange between neighbors in the social graph the probability increases to keep traffic in the close physical proximity. The corresponding advantages have been discussed in previous chapters already. On the other hand friends will still provide pieces even if no credits can be transmitted in return. The latter affects priority classes 3 and 5 only, since friends that are currently downloading data (class 1) may be hindered since they would spend upload capacity without receiving data in return.

Priority classes for exchange partner selection

Idle friends may provide support without reward

The specific form of support received by peers of class 5 may vary. In case the help-requesting peer A has less than $\frac{m_C}{2}$ credits left while the supporting peer B is possessing more than $|m_C|$, B donates $\frac{|m_C|}{2}$ credits to A. This prevents A from running out of credits and thus supports its download. In any case B starts downloading the corresponding file too, focusing its download on those pieces A is still missing. All downloaded data is immediately forwarded to A. Since B is sat-enabled, the fact that it is now downloading the file as well increases the probability that it will be broadcasted.

The Different forms of support

The procedure is similar for a non sat-enabled client except that it will usually know less peers, thus the choice within the classes will be smaller. Furthermore, it will not be able to profit immediately from a broadcast. However, there will be an indirect benefit since its sat-enabled friends will be able to cache the data and thus become an additional source for pieces as described above. The details of caching will be further discussed in section 8.1.4.

8.1.3 Rating probability of future demand

For several aspects of SST it is necessary to know the probability of a certain file for being required in the future. For the corresponding metric several parameters are taken into consideration. These are the preferences of the corresponding user itself as well as the preferences of his immediate neighbors in the social network graph, their history of downloaded files, potential ratings and recommendations. The latter may also include links to content shared in OSNs. Predictions and recommendations based on collaborative filtering, ontologies and social ties have been extensively studied, e.g in the work of [53], [51], [49], [20], [92], [52], [62], [95] and [70]. The selection and evaluation of a qualified approach for SST is not subject of this work. For the evaluation in this chapter a simple metric is used that allows a rating of future demand probability on a file-category level. Thereby the user profile contains a quantifier for all content categories the corresponding user is interested in. Its value rates the degree of interest in the specific category.

Comparing future demand probabilities of different files

This requires that for any incoming piece, a client has to know the corresponding file's category in order to make a match with the user's preferences. The category of a file is provided in its torrent file. In case of a user initiated download this file has been received prior to the download. For pieces received via broadcast this is not necessarily the case. Downloading the missing torrent files from the corresponding servers would be possible. However, since a vast number of files might be broadcasted, this would cause a high amount of additional unicast traffic. Having a hybrid network at hand makes broadcasting them an appealing alternative. In case of a content provider with a relatively small set of files—e.g. a video on-demand provider or a software distributor

Matching pieces and files to categories offering updates—simply all torrent files might be broadcasted. However, if we imagine YouTube as a content provider, the number and the aggregated size of all torrent files would be enormously large. In consequence broadcasting all of them makes no sense, the more since 90% of all YouTube videos are only seldom requested [25]. In that case the content provider was required to make a pre-selection of popular torrent files that should be distributed. The clients would then keep those that match their users' preferences. However, this still might result in a significant number of unnecessarily broadcasted files. Further, deciding which files will belong to the most popular 10% in nontrivial. Though there is a much easier solution.

Distribution of relevant torrent files

If a piece of a file is broadcasted for the first time, prior to that broadcast the torrent file is being distributed via satellite. The same is done in case the period since the last broadcast of that torrent exceeds a specified threshold. The clients will then decide for the torrent file whether to keep it or not.

Corresponding metric is not part of the protocol

However, this is solely an implementation detail and may even be different for distinct clients. For the protocol it is assumed that a corresponding metric exists that allows a comparison of the probabilities for future demand for distinct files. Let $P_{\mathrm{FD}}(f, u)$ be the probability that a file f will be required by a user u in the future. This incorporates also the probability that a certain file might be needed by a neighbor in the social graph. However, the own demand it given priority.

8.1.4 Caching

Every HGW, and thus every client, is equipped with a persistent storage that is used as a cache. SST considers each piece of a file that is received for caching, no matter whether it arrives via unicast or broadcast. In case there is still enough cache capacity available pieces of all files are stored without further inspection.

Caching in the face of limited resources

Let the remaining cache capacity be C^R. If however no or not enough free storage space is remaining, it must be checked whether the pieces should be withdrawn or be stored by means of replacing an existing cache entry. For this purpose P_{FD} is computed for the file the pieces belong to.

As described in the previous chapter a broadcast piece message (PBM) contains a list of pieces for a given file while a unicast peer wire message carries always just one piece. For the sake of simplicity it is assumed that in any case there is a list of pieces which in case of a unicast PWM contains only one element. We refer to this list as P. Let the file that corresponds to the incoming piece list be f_P and the **cache index** containing the identifiers of all files of which pieces are cached be defined as

$$C^I = \{f|\text{pieces of } f \text{are stored in the cache}\} \tag{8.1}$$

Furthermore the following functions shall be defined:

$$c(p) := \begin{cases} 1, & \text{if piece p is in the cache} \\ 0, & \text{otherwise} \end{cases} \tag{8.2}$$

$$s(P) := \{\text{storage capacity required to cache P}\} \tag{8.3}$$

$$min_{\text{PFD}} := min(P_{\text{FD}}(f_1), ..., P_{\text{FD}}(f_n))\forall f_i \in C^I, n = |C^I| \tag{8.4}$$

Searching a candidate for replacing

The latter returns the smallest probability for future demand for all cached files. For equation 8.3 it is worth noting that in case some of the pieces contained in P are already cached, only the size respectively the storage demand for those pieces in P that are not yet cached is returned. Let further be a function defined that returns the set of all files that have the lowest probability for future demand of all cached files.

$$min(C^I) := \{f|f \in C^I \wedge P_{\text{FD}}(f) = min_{\text{PFD}}\} \tag{8.5}$$

Then the algorithm shown in 2 describes how the client handles an incoming list of pieces. In case there is enough capacity left to store all the pieces, they are simply added to the cache. If not, files with the lowest probability for future demand are removed until either enough capacity is free or non such files remain. Then as many pieces contained in P as possible are added to the cache.

Cache hit rate increase by keeping once downloaded files

The just described caching strategy is supposed to deliver good cache hit rates for two reasons. First many HGWs should be used by several users, e.g. a family living together in one household, potentially also by friends who often come for a visit and connect with their mobile devices. People using the same HGW usually will have some social tie since the collective usage requires them to be in a relatively close physical proximity. In consequence there is an increased probability for the involved individuals to mutually influence each other. Hence the likelihood for a file previously requested by one user registered at a specific HGW to be demanded by another user connected to this same HGW is elevated. Further the one user might want to download the same file on several of his own devices. Both aspects make caching of already downloaded content a reasonable approach.

Benefits of caching broadcasted data

Caching broadcasted content is interesting since only very popular files are being broadcasted. Due to their high popularity, their chance for being recommended by other users (potentially friends) is increased. This also results in a growing probability for becoming subject of the locally managed users' demand in consequence of the mutual influence among people and in particular between

Algorithm 2 Cache decision for incoming pieces

if $C^R \geq s(P)$ **then**
    add P to cache
else
    while $P_{\text{FD}}(f_P) > min_{\text{PFD}} \wedge C^R \leq s(P)$ **do** ▷ Remove files with lower probability for future demand from cache
        for all files $f \in min(C^I)$ **do**
            remove f from cache
        end for
    end while
    for all pieces $p \in P$ **do**
        if $c(p) == 0$ **then** ▷ is the piece not cached already?
            if $C^R \geq s(P)$ **then** ▷ check remaining cache capacity
                add p to cache
            else
                break; ▷ the cache is full, stop caching
            end if
        end if
    end for
end if

friends as stated in [55]. Since caching relies on the probability metric for future demand, which further incorporates social networks, this cached content can potentially be used to support downloads of friends in terms of the social aiding described previously. In particular this lets non sat-enabled peers take profit of the hybrid network and the satellite broadcasts.

8.1.5 SocioAware Prefetching

Limitations of the functions introduced so far

The just described caching is a strategy that can potentially fill the local cache of sat-peers with files that have a good chance for being needed in the future. However, the scope of files that can be cached by means of the functions described so far is limited. Especially niche content would never find its way into the local cache by these approaches. The same applies for preferences of non sat-enabled peers that none of their social contacts share. This means for all clients there is no chance of ever having an unpopular—or at least not yet popular—file in the cache that has never been requested by any attached user so far. However, the are several reasons that make it desirable to have also these files cached in advance to the actual demand. On one hand this allows immediate access even in case the Internet connection bandwidth of an HGW is very small. On the other hand, considering the typical day-time curve of Internet activity that has been described in chapter 1, downloading a file in advance allows avoidance of traffic peaks. SST's strategy to close this gap is socio-aware prefetching which is described in the following.

The prefetching procedure on detail

When a peer is idle it checks the file repositories of the known content providers for potentially interesting files. Thereby for a detected file f the probability $P_{\mathrm{FD}}(f, u)$ is determined for every user managed by the HGW. In case the resulting probability is above a **prefetching threshold (PT)** plus exceeds min_{PFD} (see section 8.1.4) a prefetching download is started. This is in general performed as any other download except a few differences. First it will immediately be postponed if a download resulting from an explicit user request is started. Secondly, the exchange partners considered for piece downloads are restricted. In comparison to the priority list shown in table 8.1, the only peers in priority classes one to three are used for prefetching. The motivation for this constraint is to avoid prefetching to negatively influence other downloads. Peers in class 1 and 2 are currently downloading the same file. Thus there will be a mutual profit here when pieces are exchanged. In class 3 are those friends who are seeding the file. According to the results of [94], [12] and [93]—who show that the majority of social connections exhibit a small geographical distance—this causes the prefetching to happen mainly in the near physical proximity of the client. This comes with the great advantage that the major part of the induced traffic is kept within a comparatively small geographical region and thus potentially inside of a single Internet service provider's (ISP) network. This is supposed to increase the acceptance of SatTorrent among the ISPs, who commonly dislike P2P file exchange protocols due to the costly inter-ISP traffic their applications produce. Additionally, due to the increased similarity and the mutual influence among these peers, there is a high chance of finding an exchange partner in this set of peers. Both also applies for nodes in class 1. Non-friend seeders of the file (class 4) are more urgently needed by other downloaders who are not prefetching or should be left alone in order to allow them to do their own prefetching. Classes 5 and 6 are not considered since otherwise this would be an exploitation of helpfulness. Instead it is preferred to give also these peers the opportunity to do prefetching. This is not purely altruistic since a file cached by a friend might be serviceable for future downloads due to the mutual influence.

Prefetching must not hinder any downloads

Keeping prefetching traffic local

Certain classes of peers are not considered for prefetching

8.2 Simulation conditions

Considerations for the SST simulations

The results of the SatTorrent analysis in the preceding chapter enabled the development of an improved simulator. Relying on the gained insights a higher level of abstraction can be used that facilitates simulating significantly higher amounts of users, files and file sizes. The integration of social networks further raised the demand for integrating new functions. Besides the obvious components like a social graph structure, user profiles and cache management, there are further aspects related to the social awareness which must be included. Most notably among these is the influence among individuals in a social network. In order to do a plausible analysis of P2P downloaders organized in a social network and thus interfering with each other, mutual influence must be taken into consideration. In the following section this topic is discussed before more aspects of the SST evaluation are introduced.

8.2.1 Mutual influence model

Modelling mutual influence in social networks

There is no doubt about the mutual influence between individuals in general and in particular between users in online social networks. While several studies analyze these incidents and develop numerous models to describe them, there is no model that can be considered as the ultimate characterization of the real world procedures. Considering this in combination with the finiteness of resources for a simulation based analysis, a decision in favor of a simple approach—in terms of the computational complexity—has been taken. In order to achieve a broader view and to allow a comparative analysis, various models have been implemented which are described in the following. For all of them the mutual influence model for every node is applied in discrete intervals with a probability p_{MI}. In order to facilitate mutual influence, each user u has a list of preferences $P(u)$. The set of all categories is referred to as C. The list of preferences contains entries which consist of an integer that unambiguously identifies the category c and a quantifier $q(c, u)$ that specifies the importance of this category for the corresponding user u. Thereby, a higher quantifier indicates a higher interest in a category, with $0 \leq q(c, u) \leq 1$. In case a category is not listed in a user's preferences, the corresponding quantifier is defined as being zero. Further the set of nodes that have a common edge with node u – in other words the direct neighbors – is referred to as the *buddies* or the *Buddy-List* $B(u)$ of this node.

Global parameters of the mutual influence models

Buddies and the buddy-list of a user

Let the following function be defined that identifies whether a category $c \in C$ appears in the properties of a node u with a quantifier greater zero.

$$a(c, u) := \begin{cases} 1, & \text{if } q(c, u) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{8.6}$$

Now the following function can be defined that counts for all neighbors $b_i \in B(u)$ of a node u the frequency of occurrences of a specific category c.

$$freq(c, u) := \sum_{i=0}^{|B(u)|-1} a(c, b_i) \tag{8.7}$$

Based on this four different models for mutual influence can be described in the following.

Three Most significant (MI1)

In this mutual influence model one of those categories with the highest aggregated quantifiers of all neighbors is used for influencing. Therefore the preferences of all direct neighbors $b_i \in B(u)$ are analyzed for a user u. For each category found in these preferences the corresponding quantifiers are summed up in

$$s(c,u) := \sum_{i=0}^{|B(u)|-1} q(c,b_i) \tag{8.8}$$

which gives us the following set containing pairs of categories and aggregated quantifiers

$$Q_u := \{(c,q_a) : q_a = s(c,u)\} \tag{8.9}$$

Further let

$$M^3(u) := \{c : (c,q_a) \in Q_u \wedge q_a \text{ is one of the three highest values in all pairs }\} \tag{8.10}$$

In other words the set consisting of those three categories that exhibit the highest aggregated quantifiers. Then one element $r \in M^3(u)$ is picked with a probability proportional to the number of occurrences $f(c,u)$. In case the user u does not have this category in his preferences, it is added with the mean quantifier. Otherwise, the new quantifier $q'(c,u)$ of peer u for category $c \in C$ is determined by modifying the old. Both variants are calculated by means of the following formula:

Applying the mutual influence model

$$q'(c,u) := q(c,u) + \frac{s(r,u)}{freq(c,u)} \cdot (1 - q(c,u)) \tag{8.11}$$

Three Most Frequent (MI2)

This approach uses as criterion the absolute occurrences of the specific categories in the neighbors' preferences to select the three candidate categories. The set defined in equation 8.12 contains all pairs of categories and their frequencies of occurrence in the preferences of all neighbors of u

Disregarding quantifiers for selecting influencing category

$$F_u := \{(c,f) : c \in C \wedge f = freq(c,u)\} \tag{8.12}$$

Now let

$$F^3(u) := \{c : (c,f) \in F_u \wedge f \text{ is one of the three highest values in all pairs}\} \tag{8.13}$$

Then one element $r \in F^3(u)$ is randomly selected. Then the new quantifier $q'(c,u)$ of peer u for category $c \in C$ is determined by the formula given in equation 8.11

Most significant only (MI3)

This model is very similar to MI1, except that the category $r \in C$ which is selected for being influenced is chosen from

$$M^1(u) := \{c : (c, q_a) \in Q_u \wedge q \text{ is the maximum of all value pairs}\} \quad (8.14)$$

In most cases this set should contain only one element. However, in case there are several categories which exhibit exactly the same maximum quantifier, one among them is randomly selected. Thus in general it can be stated that a category r is randomly chosen from $M^1(u)$. The modification of the corresponding quantifier follows equation 8.11.

Random (MI4)

Random selection from all neighbors prefences

From $B(u)$ one neighbor b_i is selected randomly. Then again a random selection is applied to pick one of this nodes preference categories c, which is then used for influencing u. In case $c \in P(u)$ the new quantifier $q'(c, u)$ is determined as shown in equation 8.15.

$$q'(c, u) = q(c, u) + q(c, b_i) \cdot (1 - q(c, u)) \quad (8.15)$$

and by

$$q'(c, u) = 0.5 \cdot q(c, b_i) \quad (8.16)$$

otherwise.

8.2.2 Experience Based Preference Feedback

Personal experience further changes user preferences

Despite the mutual influence between individuals, the opinion of users is also influenced by personal experiences. Thus whenever a user completes the download of a file in the simulation it will influence the quantifier of the corresponding category in his preferences. With a defined probability (simulation settings) this change will be negative. The magnitude of the modification is depending on the variety of interests and the current quantifier value.

8.2.3 Social Graph Modelling

The social graph models used in the simulations

The simulator supports different graph modeling algorithms. For the evaluation the Barabasi-Albert, Watts-Strogatz and Toivonen models have been used since all of them result in graphs that exhibit characteristics similar to real world social network graphs. The Toivonen model is less popular and not as extensively analyzed as the others. The graphs generated by this model do not exhibit hubs with extremely high node degrees as they can be found in BA graphs.

This is an interesting property since the graphs of those P2P networks that consist of only private users might be missing such super hubs. However, the P2P overlay networks for the proposed content distribution model also contain the seeder nodes of the content provider, which usually should have very high degrees. Nevertheless, including the Toivonen model into the analysis might provide valuable information about SSTs performance in a scenario where no super-hubs are involved.

The Watts-Strogatz model is well studied and a good choice for generating small world networks. However, the resulting graphs are missing the scale free property that is commonly observed in real world social network graphs. This suggests that the Barabasi-Albert model, despite its known deficiencies, should be the most qualified approach for this evaluation. More details on the properties and analysis results will be provided later in section 8.3.

No scale-freeness in WS graphs

8.2.4 Static Simulation Parameters

For the sake of comparability as many parameters as possible are kept unchanged throughout the evaluation phase. These are summarized in table 8.2. There are ten initial seeders with each of them providing all the files. All participants—including the seeders—share the same asymmetric connection bandwidth. After a peer has finished a download, before starting a new one it waits for a random time whose global average is set to two hours. The remaining parameters are introduced later in section 8.3 with their particular values.

The fixed paramters for all simulations

Parameter	value
File Size	100 MB
Categories	100
Seeders (initially)	10
Download Bandwidth	8 Mbit
Upload Bandwidth	1 Mbit
Sat-Enabled Peers	30%
Avgerage download wait time	2h

Table 8.2: Static Simulation Parameters

8.3 Evaluation

In this section the performance of SST under varying conditions is measured and compared. Before analyzing the file distribution with this protocol in particular, the impact of the applied social graph modeling algorithms on the distribution of sat-enabled peers in the social network is examined in section 8.3.1. This is followed by a survey on the effects that the different mutual influence models introduced in section 8.2.1 have on the overall network traffic in section 8.3.2. Then we conclude with a comparison of SST to SatTorrent without social enhancements and to BitTorrent in section 8.3.3.

8.3.1 Dish Distribution Analysis

Realistic percentages of sat-enabled peers

As the results of preceding SatTorrent evaluations in chapter 7 show, the protocol's performance increases with rising ratios of sat-enabled peers. Even though a change in that trend has been observed for fractions of sat-enabled peer of more than 95% under certain circumstances, it has been shown that this effect can be leveraged by optimizing SatTorrent's parameters. Though, in a real world scenario it might happen that the ratio of sat-enabled peers is relatively small. For example in the USA, according to statistics published in [82] and [81], at most 40% of all households with broadband access also have satellite reception hardware. Even though this proportion might increase in case the herein proposed content network model should become available and thus the parallel installation of both more appealing for the time being using a lower value is more authentic. In consequence, for the remainder of this chapter a considerably lower value of 30% of sat-peers will be assumed. As the preceding SatTorrent evaluations revealed the protocol's performance significantly increases in case the relative number of sat-enabled peers is higher. Thus the results presented later in this chapter can be considered as a lower bound that corresponds to the current situation.

Probability for having sat-enabled neighbors

However, for the overall performance of SST not only the number of sat-enabled peers is important. An additional crucial factor is the number of nodes that can immediately take profit of a sat-peer. In particular this refers to the question how many clients have a sat-enabled direct neighbor in the social graph. Or inversely beheld, how many do not. We refer to the probability of an arbitrary node in the network for having no sat-enabled direct neighbor as P_{NSN}.

Parameter selection for graph models

In this context the distribution of sat-enabled peers across social network communities is analyzed. Therefore the social network graphs are generated according to the models introduced in section 8.2.3 and analyzed afterwards. For the Barabasi-Albert (BA) model the initial graph is composed of 5 nodes ($m_0 = 5$) with a minimal node degree of 3. Each node added to this graph connects to 5 existing nodes via preferential attachment ($m = 5$). While this results in scale-free small world graphs, the average clustering coefficient is significantly lower than those that can be observed in real social networks. This behavior is well-known for BA graphs and is analyzed in [40]. The clustering declines as the network grows. In order to achieve a higher clustering coefficient another BA graph with $m_0 = 50$ and a well connected initial graph has used for comparison. For this graph, referred to as BA2, each new node connects to 4 existing ones.

For the Toivonen graph (TO) the settings $m_r = 2$ and $m_s = 3$ are used, for the Watts-Stragatz graph (WS) it is $beta = 0.2$ and $K = 10$.
The resulting graphs are illustrated in figures 8.1 to 8.4. In figure 8.1 one can observe the typical shape of a BA graph with very few nodes showing an extremely high degree while it is very small for the majority of nodes. This is different when the initial graph for the BA model is changed, as figure 8.2 shows. Here the number of nodes with a high degree is visibly increased but those nodes with a moderate degree are almost missing completely. This is a consequence of the large well connected graph used initially in the creation process. This effect can also be observed in the degree distribution plotted in figure 8.6, especially in comparison to figure 8.5. This indicates that the settings used for the BA2 graph are not suited for the purpose of analyzing SST.

Characteristics of the resulting BA graphs

The Toivonen graph in figure 8.3 exhibits fewer nodes with a very high degree. Figure 8.7 confirms this and shows a break in the distribution for very low values. Thus the number of nodes with only one or two neighbors—friends in the objected use—is significantly lower than for the BA models. Further also the highest observed degrees are much lower in the TO graph. The latter is a declared objective of the model. While the former is debatable, it is not necessarily undesirable. In the scenario of a content distribution network based on SST, there is an incentive for users to have more friends in order to maximize the corresponding benefits.

TO graph shows less dominant hubs

The Watts-Strogatz graph (WS) illustrated in figure 8.4 considerably differs from the previous three. This is due to different degree distribution where most node degrees accumulate around K. This is confirmed by the Poisson like shape of the plot in figure 8.8. These are typical properties of a WS graph and thus not surprising. Albeit all these differences to the previously discussed ones, this is also a small world graph.

Peculiarities of WS graphs

The later can best be observed in tables 8.3 and 8.4 There several important statistical properties of the generated graphs are listed, once for graphs consisting of 1000 nodes, once for the order of 10000. Most notable are the average path lengths, which are very short despite the little differences between the various graph models. With regard to the Toivonen model this—together with the other observed properties discussed above—confirms the results presented in [103]. The previously mentioned decreasing clustering in BA graphs for rising graph orders is distinct for both graph settings. Even though the unusual increase of the initial graph resulted in a relatively strong gain of the average clustering coefficient, its absolute value is still very small for the higher order graph. Especially, it is considerably lower than those values observed in an OSN [104]. Whether or not this has a considerable impact on the performance of SST

Comparing the graph analysis results

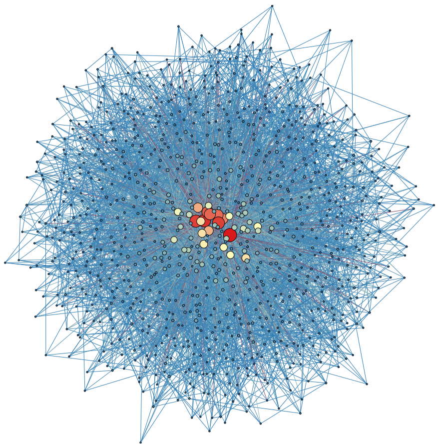

Figure 8.1: BA graph (order=1000)

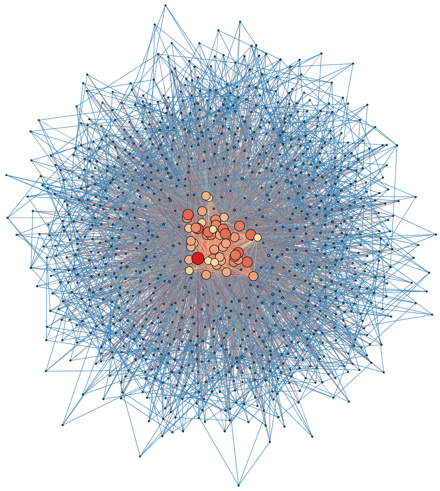

Figure 8.2: BA2 graph (order=1000)

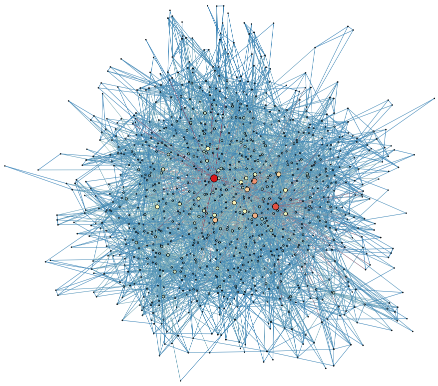

Figure 8.3: TO graph (order=1000)

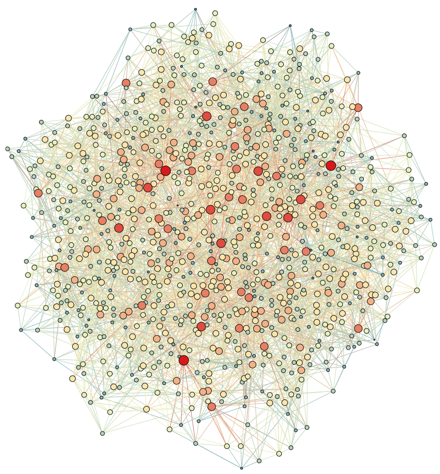

Figure 8.4: WS graph (order=1000)

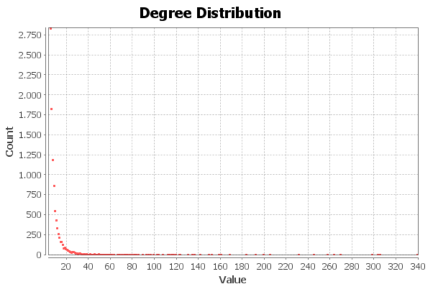

Figure 8.5: Degree distribution of BA graph (order=1000)

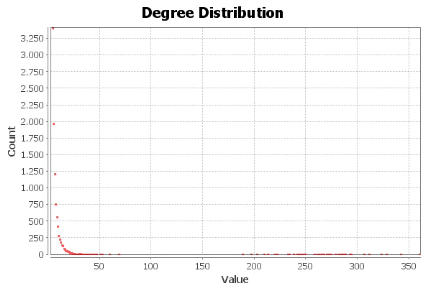

Figure 8.6: Degree distribution of BA2 graph (order=1000)

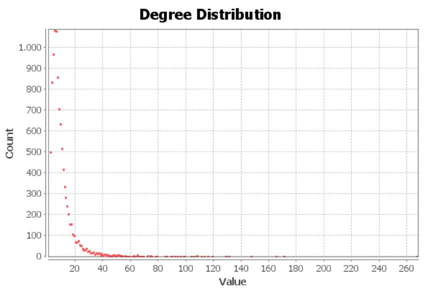

Figure 8.7: Degree distribution of TO graph (order=1000)

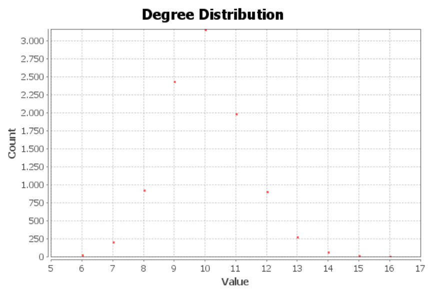

Figure 8.8: Degree distribution of WS graph (order=1000)

is analyzed in section 8.3.3.

Analyzing probabilities for sat-enabled neighbors under varying SPR

Next P_{NSN} is examined under different proportions of sat-enabled peers, varying from 0% up to 100%. The graph order used for this analysis is 100,000. The results are visualized in figure 8.9. All the four curves are rapidly decreasing, while for the Watts-Strogatz graph the probability recedes most quickly. It can be stated that for settings with more than 50% of sat-enabled peers the choice of the graph modelling algorithm in completely negligible. In the interval between 20 and 30 percent the highest differences can be measured. Thus for the further evaluation, picking a value within this range will give the most interesting results.
The small difference between the models in this interval can also be observed in figure 8.10 where P_{NSN} is compared for the different graphs under varying network orders and a constant sat-peer ratio (SPR)of 0.3. First to mention is the fact that within the distinct networks the results for P_{NSN} remain more or

	BA	BA2	TO	WS
Edges	4981	4876	4864	5000
Average degree	9.962	9.752	9.728	10
Network diameter	5	5	7	6
Avg. path length	2.969	2.790	3.511	3.856
Avg. clustering coefficient	0.042	0.246	0.544	0.344
Total triangles	978	15557	6355	5068

Table 8.3: Comparison of graph properties for network order=1,000

	BA	BA2	TO	WS
Edges	49981	40876	48414	50000
Average degree	9.996	8.175	9.683	10
Network diameter	6	6	9	8
Avg. path length	3.653	3.427	4.426	5.241
Avg. clustering coefficient	0.008	0.037	0.536	0.347
Total triangles	2185	19407	58976	51419

Table 8.4: Comparison of graph properties for network order=10,000

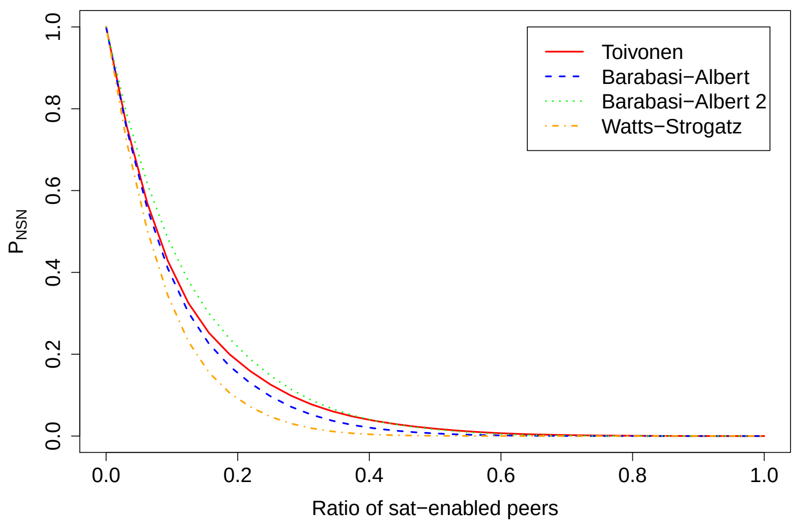

Figure 8.9: P_{NSN} for varying dish ratios (order=100,000)

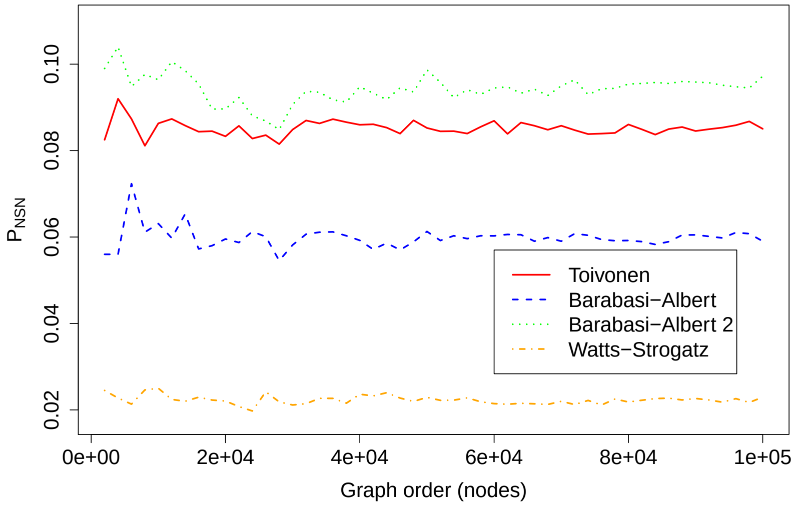

Figure 8.10: P_{NSN} for varying graph orders (SPR $= 0.3$)

less constant for growing orders. Only for small numbers of nodes an identifiable fluctuation is present.

Determining best model for SST analysis

Even though the observed differences between the various graphs are small in absolute values, they are significant. Especially in the WS graph the risk of not having at least one friend who is sat-enabled is very low. Thus when using the WS model for generating the graphs for the analysis of SST's performance, the results will potentially be better than those that would be expectable in real networks in case they rather resemble one of the other graphs. The BA graph with the settings used for BA2 should not be considered either for the reasons mentioned previously in this section. Therefore, in the remainder of this chapter the BA model will be preferentially used for further analysis of SST, supplemented by a direct comparison of BA, TO and WS graphs.

8.3.2 Impact of Mutual Influence Model

Estimating the impact of different mutual influence models on SST

In section 8.2.1 different ways of modeling the mutual influence among users have been introduced. Since the prediction of future user behavior and thus the prefetching, caching and cache replacement strategies rely on the analysis of the user profiles and those of the corresponding buddies, the mutual influence model is crucial for the reliability of the simulation results. However, it is impossible to define one universal mutual influence model that would correctly reflect the behavior for all possible applications of SST. The comparison provided in this section will show how much the performance of SST is affected by variation of the applied model. In this context first the average download durations for the different models are being compared. Besides the three realistic models (MI1, MI2, MI3) this analysis is additionally performed for a fourth, random approach (MI4). The latter is considered being not realistic and thus its application is expected to deliver rather undesired results by leading to cache misses and failed prefetching attempts. Using MI4 will indicate the performance of the social features in SST in case no assumptions about the process of mutual influence can be made. Figure 8.11 shows a comparison of average download times. While

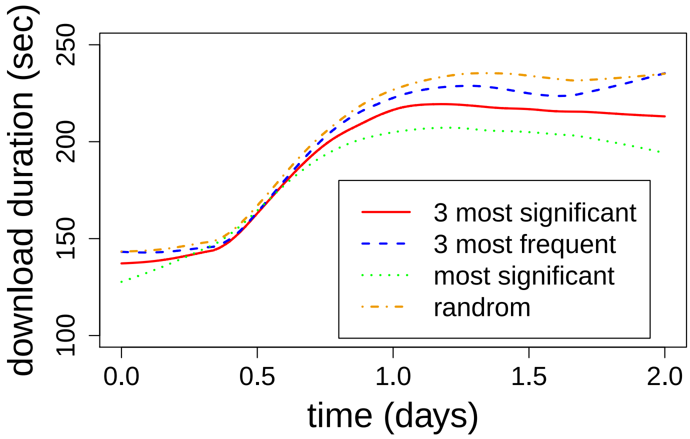

Figure 8.11: Average download duration over time compared for the different mutual influence models.

Random approach as expected gives worst results

all the four models are close together, indeed the worst results can be observed for the random approach. The common increase in the download times after 12 hours is caused by the increasing number of concurrent downloads paired with a high prefetching activity. Thus a point is reached where insufficient sources are available to satisfy the demand. This is due to the asymmetric Internet connection with an download/upload ratio of 8:1, which makes it simply impossible to have all peers concurrently downloading at full speed. Whether or not this means that SST with prefetching leads to a better or maybe worse overall performance than non-social SatTorrent or other P2P protocols will be investigated later in section 8.3.3.

Another important factor for the benchmarking of SST is the number of bytes that are exchanged between peers that are not neighbors in the social graph, since this traffic is assumed to be less local and consequently to be more costly for ISPs. The results of the corresponding analysis are visualized in figure 8.12. While MI1 to MI3 again show only very little differences, the random approach exhibits a different behavior with lower values at the start. The reason for this effect is that at the beginning the differences between neighbors are still more

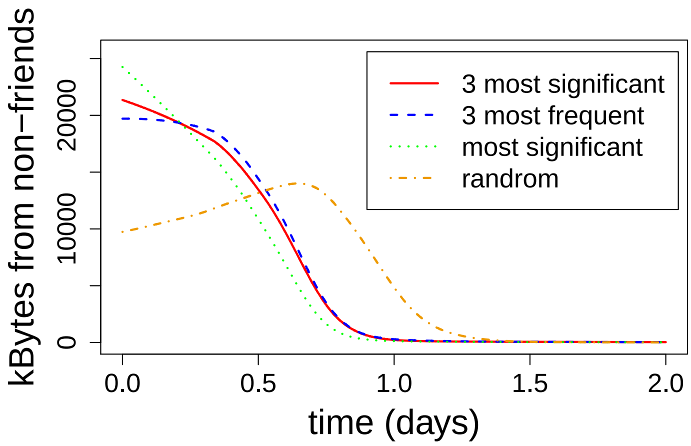

Figure 8.12: Unicast network traffic between non direct neighbors in the social graph over time compared for the different mutual influence models.

Realistic models need time to outperform random approach

significant, while they assimilate more and more in the course of the simulation. Thus a random approach can perform better at the start since the changes are unbiased. However, the random approach performs worse after approximately half a day. Then the other models led to an assimilation of neighboring nodes which allows better prediction and improved synergy effects. Interestingly, all curves quickly converge to zero after a relatively short period. This is a result of the satellite broadcasts, the caching and the prefetching. The effects of these features will be investigated in more detail in the next section 8.3.3.

The harder the prediction, the more important the broadcasts

When we have a look at the correlation coefficients of download duration and being sat-enabled respectively the number of sat-enabled friends (table 8.5), a better understanding of this difference can be achieved. For MI3, having a dish is still important, but not as much as for the other models. The same relation is observed for the number of sat-enabled friends. The reason is that MI3 leads to user preferences where a few categories reach very high ratings while most others remain low. Thus less files are sufficient for a good cache hit rate. What

	Sat-Peer	#Sat-Peer Friends
3 most significant (MI1)	0.4613411	-0.1048858
3 most frequent (MI2)	0.499472	-0.1194338
most significant (MI3)	0.3898186	-0.09718519
random (MI4)	0.5298729	-0.1245606

Table 8.5: Comparison of correlation coefficients (duration, sat-enabled)

can be derived from this is that the less exactly the future demand can be predicted, the more important the satellite broadcasts are. Also the more diverse the user preferences, the higher the benefit gained from broadcasts.

Section conclusion

Concluding it can be stated that be best results have been achieved by means of the mutual influence model using the most significant category in the neighborhood for the influencing of user preferences. However, all models, even the random approach, exhibit such a high similarity that it can be affirmed that the details of the driving forces behind mutual influence in social network is not decisive for SST's performance.

8.3.3 Content Distribution Efficiency

Analyzing the performance gain of social network integration

In this section the performance of SST with regard to distribution time and network traffic is analyzed. Thereby a special attention is paid to exchange of data respectively pieces which are transferred between clients that are no direct neighbors in the social graph. This will be an indicator of the successful design of the SST protocol respectively of the proposed content distribution model. For the latter, one objective was to reduce inter-AS respectively inter-ISP traffic. Since human beings tend to have most social contacts in their near physical proximity, a low ratio of traffic between non friends would indicate that this aim has been achieved.

Specifying simulation parameters

For first set of experiments in this section one satellite transponder is utilized with a net bandwidth of 36 Mbit (error correction already deducted). Further the social network is based on a BA graph with $m_0 = 5$, $m = 5$ and a minimal node degree for the initial graph of 3. Thereby for all of the corresponding simulations the same set of SST specific parameters has been used which is shown in table 8.6. These settings can be divided into two distinct blocks, one with and another without support for payload and metadata broadcasts. Each item in the latter has a broadcast enabled equivalent in the former. Thereby a corresponds to i, b to g, c to f and d to e. The only broadcast entry that has no counterpart is h. This correlation is considered in the subsequent figures by means of coinciding colors. Thereby the broadcasts use dashed, the pure unicast approaches solid lines.

Comparison of simulation results for download duration

Figure 8.13 shows the mean download duration for these different settings. Thereby for a and b a significantly increasing duration can be observed. This is caused by a high number of downloaders joining within a relatively short time frame. This leads to a shortage of file sources as more and more clients start downloading. According to the asymmetric Internet connections that have been assumed for all peers, a contemporaneous download of all clients will not be able

id	description
a	BitTorrent simulation mode; all social features disabled
b	download support by friends enabled (social aid)
c	download support by friends; prefetching enabled with prediction of future popularity
d	download support by friends; prefetching enabled with prediction of future popularity but limited to an average of 10 concurrent prefetchers in the entire network
e	broadcasts enabled (payload and metadata); download support by friends; prefetching enabled with prediction of future popularity but limited to an average of 10 concurrent prefetchers in the entire network
f	broadcasts enabled (payload and metadata); download support by friends; prefetching enabled with prediction of future popularity
g	broadcasts enabled (payload and metadata); download support by friends enabled (social aid)
h	broadcasts enabled (payload and metadata); prediction of future demand; other social features disabled
i	broadcasts enabled (payload and metadata); all social features disabled

Table 8.6: SST specific simulation parameters

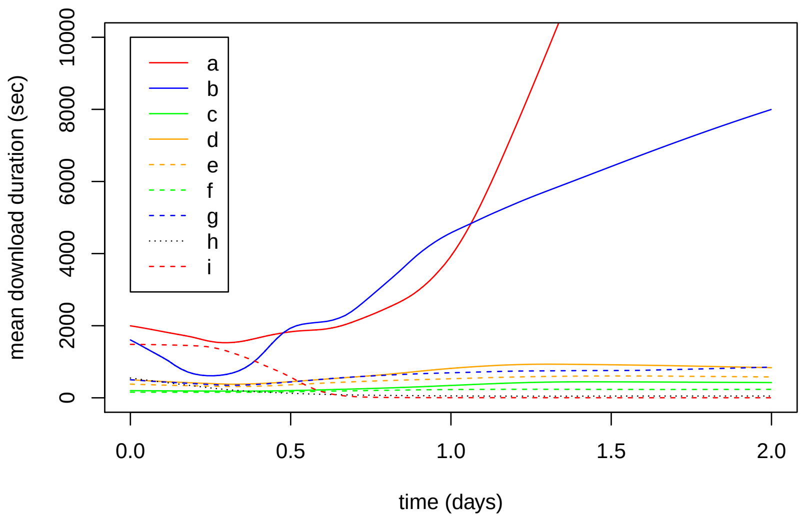

Figure 8.13: Mean download duration over time for different configurations

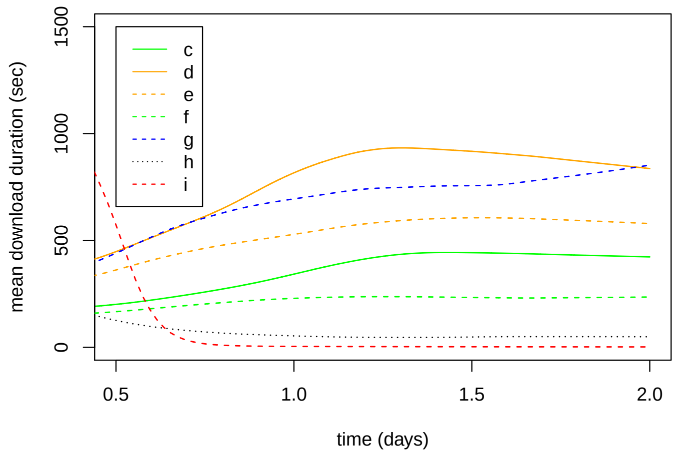

Figure 8.14: Magnification of figure 8.13

to exceed download rates equal to the mean upload bandwidth of all nodes. The remaining configurations which make better use of the social network structure or even use broadcasts all perform much better. Thereby it can be observed that the broadcasting counterparts always delivers better results than the corresponding unicast-only approaches. The difference might look small, but the broadcasts reduce the mean duration up to one third. This becomes more visible in the magnification of the lower right corner of this plot provided in figure 8.14. The results for **h** and **i** seem to be especially impressive, at least on the first view. However, figure 8.15 reveals that the average number of files downloaded by each user is much lower for these configurations. What happened here is that sat-enabled peers—due to cached content—often have download times of zero seconds since the files are already completely cached when they start the

Socially augmented models exhibit faster downloads

Higher throughput for sat-enabled peers

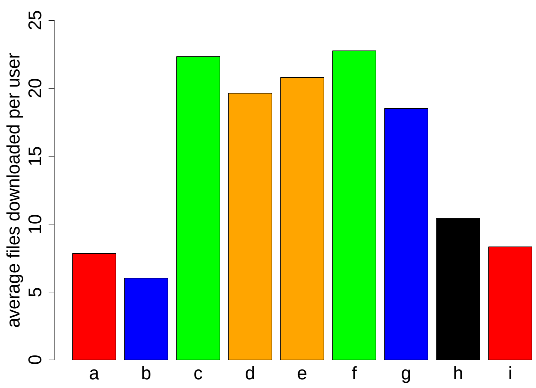

Figure 8.15: Number of files downloaded by each client on average.

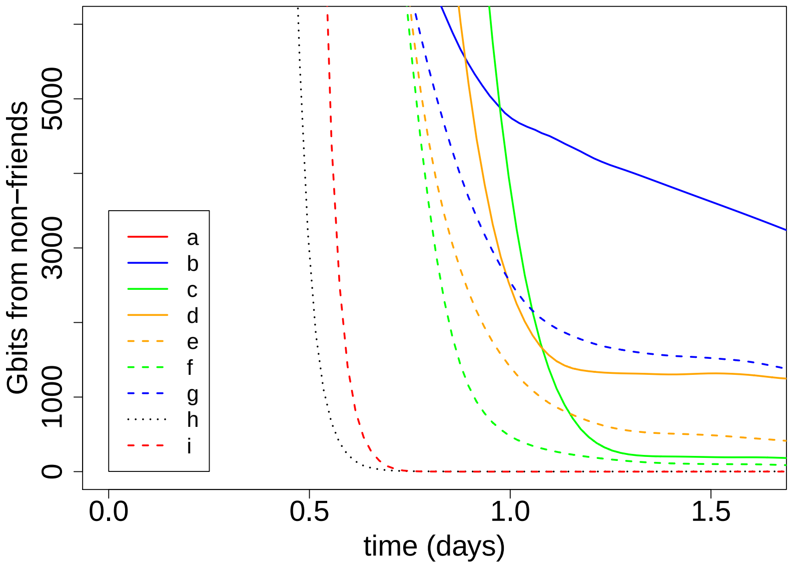

Figure 8.16: Evolution of traffic between nodes that are no neighbors in the social graph.

corresponding download. However, non sat-enabled peers exhibit much longer download durations. Since the fast downloads resulting from cache hits decrease the mean duration, the very slow unicast downloads have a negligible influence the average values. Thus, this combination of a relative small number of very long lasting unicast downloads and a high number of extremely fast broadcast transfers leads to the curve progressions for h and even more for i in figure 8.13 and 8.14 with very low average values.

Cache hits increase total downloads and reduce average duration

In figures 8.16 and 8.17 the amount of data received from peers that are not in the downloader's buddy list (not friends) is compared for the various configurations of table 8.6. Figure 8.17 shows absolute values and exhibits high results for setting a. Due to that reason the corresponding curve does not even appear in figure 8.16 where the range has been limited to the most important section. According to these and the previous results it can be stated that BitTorrent can not compete with the social network enhanced approaches. What can be observed again in figure 8.16 is that those simulations that use broadcasts converge to lower values than their non-broadcast counterparts. Considering that the so far observed absolute unicast traffic is only slightly lower for the broadcast approaches, this is an important factor since it documents an increased locality for the distribution supported by broadcasts.

SST outperforms BitTorrent, broadcasts further reduce long distance traffic

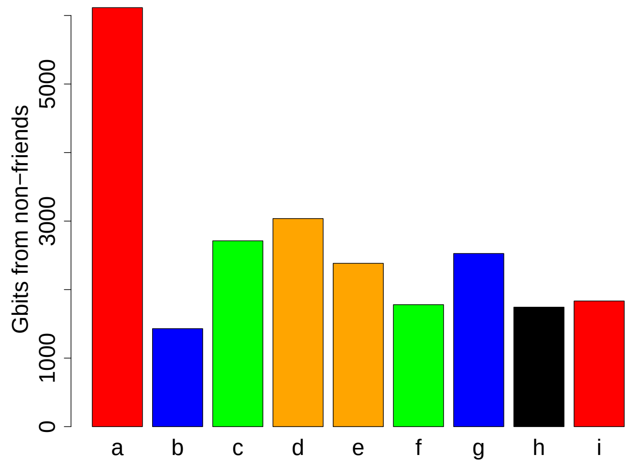

Figure 8.17: Aggregated traffic between non-friend nodes.

Good results even with thin content provider seeder nodes

For the next analysis, which focuses on the comparison of the various protocol functions in particular for broadcast mode, a longer time period of 12 entire days is considered. Thereby the simulation uses 1000 peers that may select their downloads from a set of 1000 files initially seeded by 10 content provider seeders. It is worth noting that the latter use the same Internet connection bandwidth as all other peers (8 Mbit download, 1 Mbit upload). In consequence the results provided in the following can be achieved without huge investments into infrastructure or hardware. Particularly this means that the presented results can be achieved even for end users sharing their own user generated content from their private devices. Further all files have an equal size of 100 MByte. The clients posses caches of homogeneous sizes with capacity for 100 complete files. For the broadcasts now five transponders are available, each with a bandwidth of 36Mbit. Still the Barbasi-Albert model with the same settings as before is used for generating the social network graph.

Broadcasts have large impact on distribution time

Again we start with the inspection of average download duration and its evolution over time. The results are illustrated in figure 8.18. Not unexpectedly a much higher average time demand for the distribution of files can be observed when no broadcasts are enabled and further neither prefetching nor help from friends is being used ($a2$). This significantly decreases as soon as broadcasts are enabled. However, the performance boost by the prefetching is still distinctive when comparing the results for parameter set $b2$ with those of $c2$, $d2$ and $e2$. For the very last mentioned the Toivonen model has been used for the generation of the social graph. According to the results of section 8.3.1, the TO and the BA model are the most suitable graph modeling algorithms for a simulative analysis of SST. Nevertheless, later in this section also the WS model will be examined

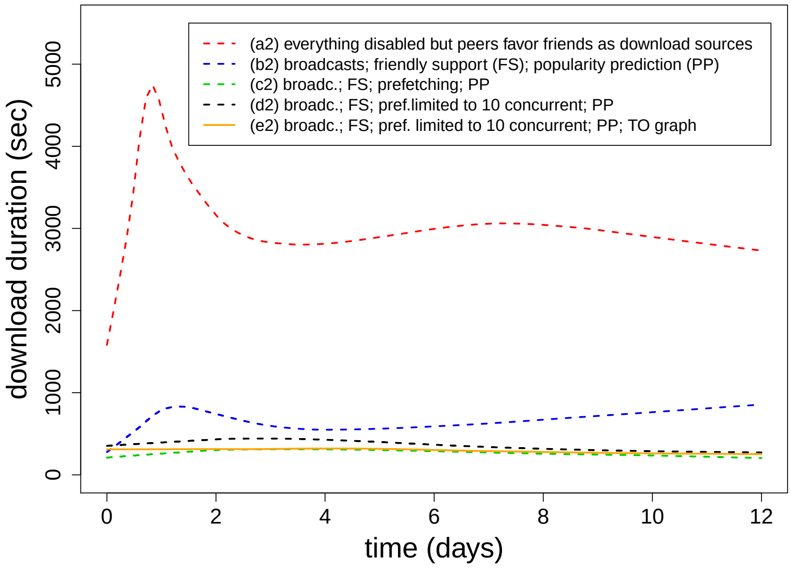

Figure 8.18: Download duration over time for different settings.

and compared to the others.

Unrestricted prefetching allows fastest download times

For a better distinction of the results for $c2$, $d2$ and $e2$ a magnification of the most relevant part of the plot is provided in figure 8.19. There it can be observed that the permission of unlimited prefetching—allowing clients to spend all idle time on it—the download duration for requested files is the shortest. Considering the two sets where the prefetching has been restricted by means of allowing only 10 concurrent prefetching clients in average, better results are achieved for the Toivonen model. Thus in order to determine a lower bound for SST's performance it is preferable to work with the BA model, at least in correspondence to results shown on figure 8.19. The outcomes of the Internet traffic comparison illustrated in figure 8.20 reveal the prize of the shorter download duration achieved by unrestricted prefetching. This relativizes the advantage of this approach observed in 8.19. The induced traffic is more than four times higher than for those cases where prefetching is limited. Considering the relatively small differences in average download duration, according to the results achieved so far it might be put into question whether it is worth this high traffic increase. At least using it as the general approach should be reconsidered. However, under certain conditions and for specific applications this might be a viable option. E.g. for the distribution of video on-demand content to nodes that have an Internet connection bandwidth lower than the video encoding bitrate—thus downloading in advance is essential—the gain in convenience legitimates an in-

Unrestricted prefetching induces high unicast traffic

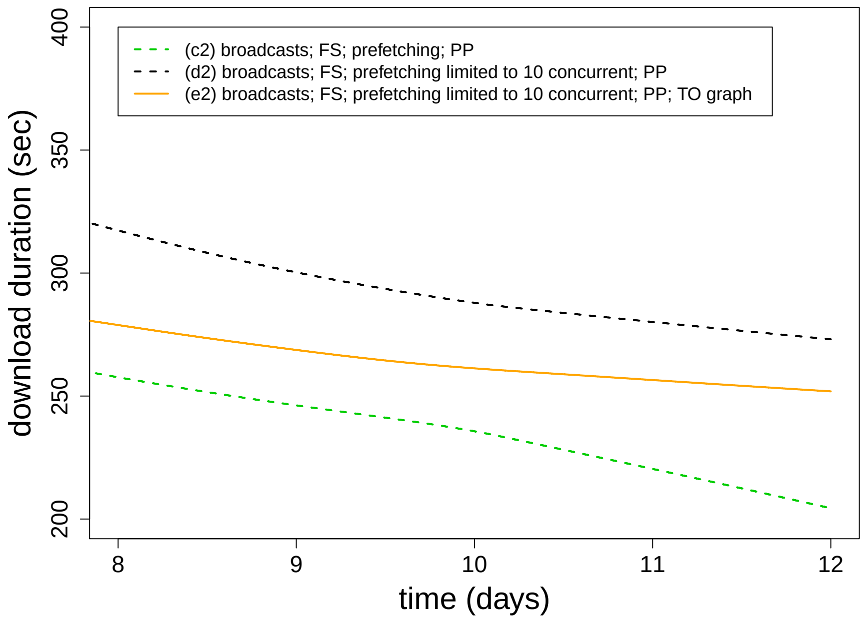

Figure 8.19: Magnification of figure 8.18.

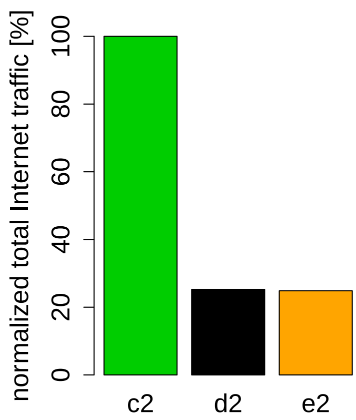

Figure 8.20: Internet traffic compared for selected SST broadcast settings.

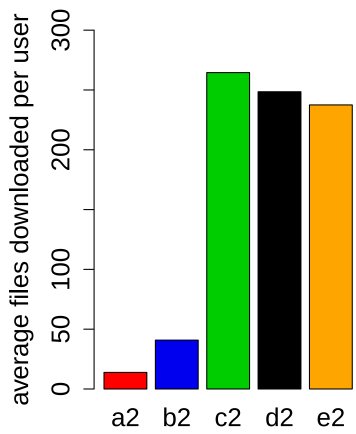

Figure 8.21: Number of files downloaded on average by each user.

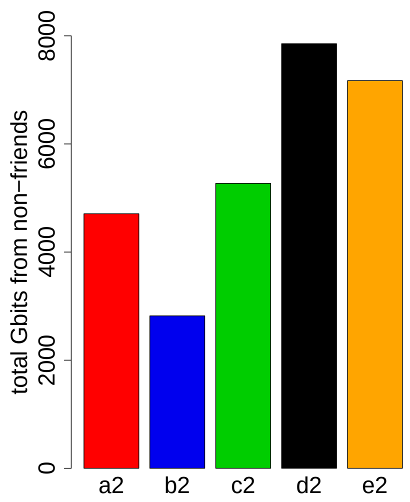

Figure 8.22: Gbits received from non-friend peers in total.

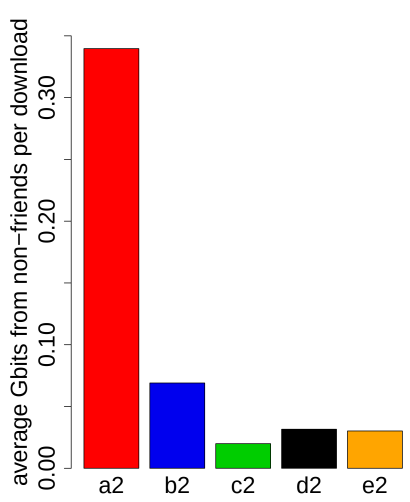

Figure 8.23: Average traffic per file between non-friend nodes.

creased traffic. Further figure 8.20 shows that in terms of Internet traffic there is no perceptible difference between the BA and the TO model.

Analyzing traffic between non-direct neighbors

The last aspect that is examined for this set of configurations is the traffic between nodes that have no direct connection in the social graph. In this context first the absolute traffic is considered. The corresponding results are illustrated in figure 8.22. While it must be admitted that very low values for $a2$ and especially $b2$ are achieved, seen at the background of the very low file download quote discussed above, both configurations are not competitive. The result for $c2$, which is also very good, completely changes the interpretation of the high network traffic caused by this unrestricted prefetching approach. Considering the fact that the biggest fraction of this traffic occurs between friends while the amount of data exchanged between nodes not adjacent in the social graph is even lower than for the restricted prefetching, this configuration becomes very appealing. The conclusions just made are further confirmed by figure 8.23, where the mean non-neighboring traffic per file is illustrated . Especially it highlights the disproportionately high transfer volumes between non neighboring clients for $a2$.

High unicast traffic for unrestricted prefetching relativized by locality

Comparing resutls for different social graph models

In the following the impact of the different graph models—in particular Barabasi-Albert, Toivonen and Watts-Strogatz—is analyzed. In this matter simulations for each of the three models have been performed with the configurations c, d, e and f of table 8.6. The results are shown in figures 8.24-8.27. What can first be observed is that again the unrestricted prefetching results in shorter download times. Second, the TO model delivers slightly better results

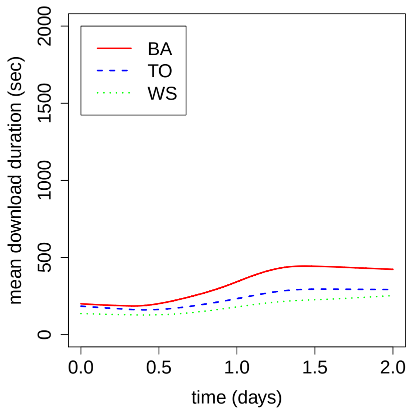

Figure 8.24: Progression of mean download duration over time for configuration c of table 8.6.

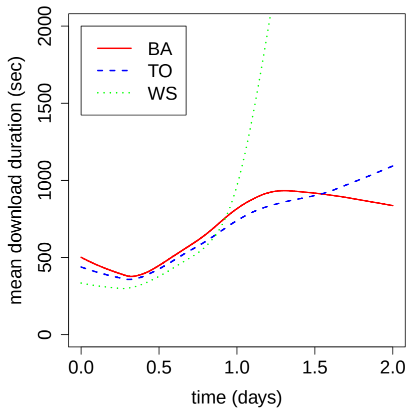

Figure 8.25: Progression of mean download duration over time for configuration d of table 8.6.

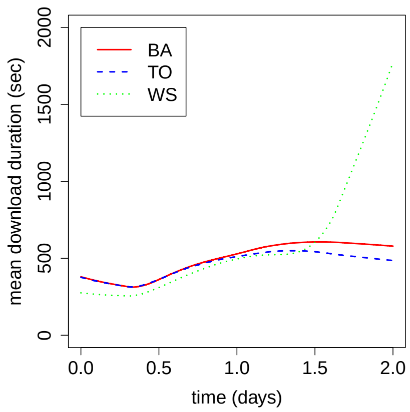

Figure 8.26: Progression of mean download duration over time for configuration e of table 8.6.

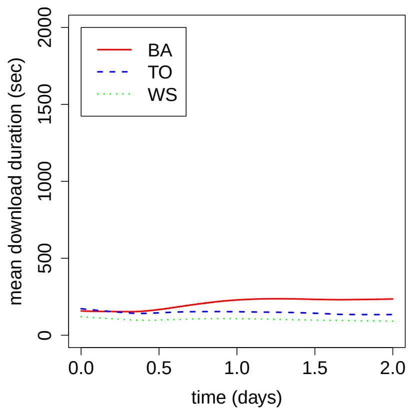

Figure 8.27: Progression of mean download duration over time for configuration f of table 8.6.

Differences between the models are small

than BA at almost any time. While WS sometimes delivers the shortest distribution times, it also exhibits very strong increases of distribution time, particularly for those settings where prefetching is limited. However, the other models also show an increase here, even though it is much less pronounced. Despite the just described outliers the results for all models are relatively close. What has been mentioned before is again confirmed in these figures: The best choice for an investigation and estimation of SST's performance is the BA model. The reason is that is bears the smallest risk of achieving better results than in reality.

Next the average number of files downloaded by each user is given a closer look. While differences are observable in figures 8.28-8.31, they are small enough in order to allow stating that the details of the underlying social network graph are not distinctive for the performance of Social SatTorrent and thus also not for the proposed content distribution model. This is again confirmed by figures 8.32-8.35 where no differences between the different models can be discovered.

No considerable impact of graph model on distribution throughput

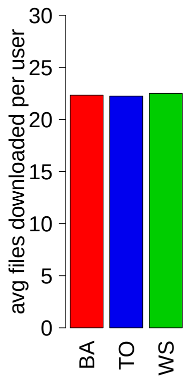

Figure 8.28: Avg downloads for config. c of table 8.6.

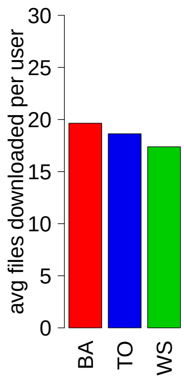

Figure 8.29: Avg downloads for config. d of table 8.6.

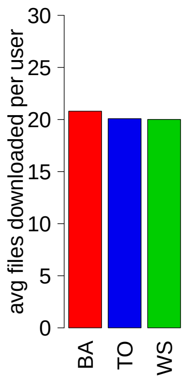

Figure 8.30: Avg downloads for config. e of table 8.6.

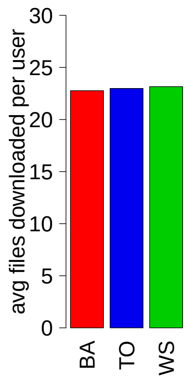

Figure 8.31: Avg downloads for config. f of table 8.6.

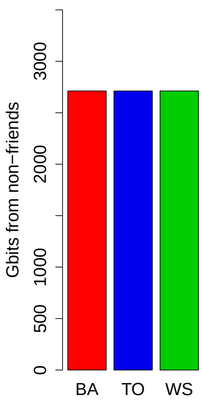

Figure 8.32: GB from non-friends for conf. c of table 8.6.

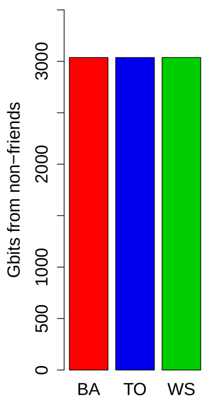

Figure 8.33: GB from non-friends for conf. d of table 8.6.

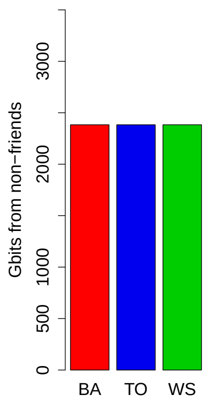

Figure 8.34: GB from non-friends for conf. e of table 8.6.

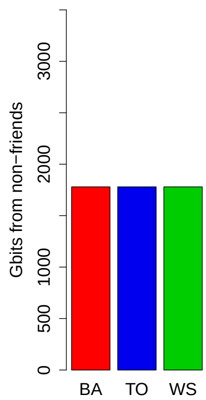

Figure 8.35: GB from non-friends for conf. f of table 8.6.

Chapter conclusion

Concluding it can be stated that social features, most notable social prefetching, implemented in SST have a large impact on all performance aspects. Furthermore it is affirmed that the better the prediction of future demand, and thus the better the cache filling and prefetching, the smaller the impact of the satellite broadcasts on the download duration due to better cache hit rates for demanded files. However, there still is a notable difference. Despite that, whenever solely the unicast network is used for delivery, all pieces of every transfered file have to traverse the network at least once for each transmission. In consequence a significant reduction of the overall Internet traffic can only be achieved by means of broadcasts.

CHAPTER 9

Conclusion

This dissertation explores the applicability of socially augmented content distribution in hybrid networks in order to maintain functionality and performance of the Internet under future online traffic situations. In this regard a model and a corresponding implementation based on a peer-to-peer concept is provided. Thereby a two layered approach has been taken that consists of a highly reliable static part, maintained by the content and satellite providers, and a dynamic part consisting of user devices where no preconditions on the node reliability are required. The decision for a P2P based model is motivated by the inherent scalability for the source side of the content delivery chain. Layering this on top of a hybrid network brings scalability for the delivery network that interconnects content providers and consumers.

Synopsis of this thesis

It has been stated in the beginning and substantiated in the course of this thesis that the social augmentation of the content distribution process is a key enabler for the good performance. The results of the comparative analysis' of the distinct aspects that have been specified for the content delivery model show that the more of its components are implemented, the better is the performance which supports coherence of the model. Since a separate evaluation has been made for the key components, an estimation of the trade-offs that were connected with a partly implementation is possible. The individual achievements from the studies of the divers facets are discussed separately in the following section.

Socially augmentation facilitates performance gain

9.1 Summary

The hybrid network based socio-aware content delivery model

In chapter 4 a novel content distribution model based on a hybrid network has been proposed. For the latter a definition has been given prior to the specification of the content distribution model. Besides the new infrastructure where packets can be dynamically routed via either of both networks a key aspect of this model is the capitalization of social networks for an increased efficiency. For this thesis the hybrid network has been further narrowed down to the combination of the Internet and satellites. The latter have been chosen for reasons of scalability, coverage and bandwidth. The objectives of the content delivery model are a reduction of Internet traffic, particularly at peak loads, scalability with respect to the number of users and files well as a reduced distribution duration compared to other approaches. The strategy taken in order to achieve this goal is a shift of redundant traffic from the unicast into the broadcast network. For substituting several unicast deliveries by one broadcast numerous requests must be aggregated. In order to avoid delays caused by this aggregation a prediction of future demand is utilized that facilitates an in-advance distribution to local user caches. For estimating future demand the social connections of the involved users have been capitalized.

The main actors of the model

The proposed model involves three main entities that implement this delivery scheme. These are the home gateway, the content provider and the satellite uplink station. The specific tasks each of the involved entities has to accomplish have been described in detail. All requests and thus the prediction of future demand are managed by the HGW. using this distributed approach gives more control about their private data to the users and further increases the scalability by keeping all centralized components of the distribution network as simple as possible. The files are not only being transfered from the content provider to the clients. For the entire distribution process a P2P technique is being used that allows the files further to be exchanged between the users. In particular a modification of the BitTorrent protocol has been suggested because its utilization of a tracker is an ideal precondition for the proposed model.

Peak occurances in online video access patterns

In chapter 5 the access patterns for online video on demand content have been analyzed. In this context the popular video platform YouTube has been monitored for a cumulative duration of more than 4 months. Over 500,000 videos have been observed and changes in view counts, rating and other relevant statistical data been recorded. The objective was to get information about timely correlations of request for the online videos, the peakiness of views. In addition this study aimed at detecting properties of highly popular videos that allow a coarse estimation of the future popularity and peakiness of distinct classes of videos. The evaluation of the measurement results that have been presented in this thesis has shown that not only heavy peaks in the requests for videos exist but further for certain classes of videos and for particular channels the peakiness is significantly increased which allows a good estimation of the future popularity of certain videos and of the distribution of their view counts over time. These results have shown that content exists whose access patterns and popularity exactly meet the requirements of the proposed content distribution model.

Analyzing the performance of dynamic network switching

In chapter 6 a first analysis of the basic components of the hybrid network based content distribution model has been performed. Thereby only actual downloads of online videos resulting from immediate user requests have been considered. For this study mobile devices have been particularly taken into consideration. In consequence, the size of the regarded videos was rather small. The simulation for this analysis used the results of the YouTube presented in chapter 5 for the adjustment of the corresponding parameters. The results have shown that even without prefetching, caching, prediction of future demand and social network integration the bandwidth consumption in the unicast network can be significantly reduced. Up to 80% and more depending on the broadcast threshold, the size of the files being distributed and the Internet access bandwidth of the involved clients. Further it has been shown that the download duration can be reduced to the same extent. However, in this case BT must not exceed the limit where broadcasts would be prohibited due to a insufficient number of concurrently downloaders. In consequence broadcasts must be made for relatively small numbers of recipients, e.g. for 2000 in case of a file size of 10.01 Mbyte and 100,000 views in total for the corresponding video. For a file of approximately 48 Mbyte BT can be increased to 15,000 already. In general the results meet the expectation that the gain of the proposed model by means of traffic savings increases with growing file sizes.

Potential for reduction of traffic and download time

The SatTorrent protocol facets

In chapter 7 SatTorrent, a P2P file exchange protocol based on BitTorrent, has been introduced. The protocol specification as well as the performance evaluation have been split into two domains. One considering the broadcasting of SatTorrent related metadata exclusively and the other additionally supporting payload broadcasts. For both approaches the principle of broadcasting data that is of interest for a broader audience while unicasting individual messages has been taken into account. The tracker has an central role in the hybrid network delivery management. It is responsible for selecting the most appropriate delivery network. In order to do this it only needs little more information than a BitTorrent tracker.

The evaluation of the metadata broadcast

For the metadata broadcasts a notable reduction on the number of unicast messages has been observed in comparison to BitTorrent. At maximum the achieved savings were approximately 50%. This is not only due to the replacement of unicast tracker responses by broadcast messages. Far more important is the increased knowledge about other peers the sat-enabled clients are provided with. This allows them to precisely select the best peers for file exchange and thus reduces especially the amount of peer wire messages.

Interpretation of the measured savings

Nevertheless, considering the high effort of satellite broadcasts that are required for this approach, savings of 50% might seem sparse. However, it is worth noting that especially the biggest messages, *have messages* and *bitfield messages*, have been abated. Further it must be considered that even in case of a small number of 2000 involved nodes and a file size of 20 Mbyte the number of peer wire messages (PWM) has been reduced by four million. On the other hand less than 30 metadata broadcasts have been required to achieve this. Location awareness has shown to have an impact on the number of messages crossing ISP boundaries, even though this was less significant than expected. However, the gain of this strategy largely depends on a high number of peers within the same autonomous system. Thus in an environment with 100,000 or more users this can be assumed to deliver much better results.

The concept of payload broadcasts

For the payload broadcasts of SatTorrent it has been shown that minor modifications, most notably having all clients sending it *have messages* after a successful piece download, enable the tracker to manage payload broadcasts. After having introduced the tracker's procedure for making a broadcast decision, the client side treatment of broadcasted piece messages has been described. Subsequently the common piece selection strategies and their applicability for SatTorrent clients have been discussed. Due to the problems of other approaches and to the heterogeneous types of content a random piece selection has been preferred.

Interpretation of the simulation results for SatTorrent

The evaluation has shown that the payload broadcasts can reduce the distribution duration by more than 50%. At the same time it has been possible to reduce the bandwidth consumption in the unicast network by over 75%. Further results have been presented documenting an increase of this effect's strength with a growing number of clients. Also is has been shown that the latter has only a barely perceptible influence on the required satellite bandwidth. Another very important aspect is the progression of relative distribution duration in relation to the size of exchanged files. In this context is has been shown that the former decreases with growing file sizes. This indicates that for files that measure several Gbytes or more the performance of SatTorrent should be much higher.

Social SatTorrent and its key aspects

In chapter the SatTorrent has been extended by means of integration of knowledge on social network8 structures of the participating users. All features that have been proposed for the content distribution model that have not been discussed until that point rely on these social networks. Download support by friends, caching of broadcasted files for the potential demand of friends, prefetching which relies on the prediction of future demand, same as the caching for personal demand depend on or take profit of the knowledge about the social graph structure. As another important aspect a persistent global rewarding scheme for altruistic behavior has been developed that elevates BitTorrents tit-for-tat policy on a higher level and in first line increases the fairness of the system in terms of remembering upload effort beyond the scope of a single file download. As a side effect, the *credits* which are used for this purpose can be traded and exchanged, and thus also used for a very easy way of supporting other peers in their download attempts.

Mutual influence and graph models

Prior to a presentation of simulation results the most important aspects of the simulation model and its parameters have been discussed. This includes the mutual influence models, which determine how users organized in a social network take mutual influence on each other's preferences. Since this is crucial for the reliability of the simulation results not only one but four different models have been introduced and used later on. The same applies for the layout of the social network graph. In consequence three different graph models have been used, in particular Barabasi-Albert, Watts-Strogatz and Toivonen.

Analisys of dish distribution and mutual influence

Then the results of the simulations have been presented and discussed. First the distribution of satellite dishes has been taken into account. Besides being sat-enabled, the newly introduced social aspects of SST further allow clients to indirectly take profit of broadcast. This is achieved by having sat-peers caching content potentially required by their social contacts. Thus special attention has been paid on the probability of having friends that are sat-enabled. This has been analyzed for the different graph models. The differences that have been observed are small. However, the highest probability for having sat-enabled friends was given for the Watts-Strogatz graph. In order to avoid idealized results the subsequent simulations concentrated more on the Toivonen and the Barabasi-Albert model. Next the impact of the different mutual influence models has been analyzed. The corresponding results have revealed that this impact in small and that the performance of SST is good even if the exact way how individuals mutually influence each other are not known.

SST's content distribution performance

Further, the performance of content distribution with SST has been evaluated, showing that the distribution time can significantly reduced by using the full potential of SST. It has also been observed that the traffic locality mostly depends on the degree of utilization of social networking features and only to a lesser extend on the broadcasts. It has been shown that by regulating the prefetching effort of clients the gains of SST can be adjusted in order to put a greater emphasis on unicast traffic savings or on the time demand for file distribution. Finally the impact on the different graph models on the simulation results has been studied. The corresponding figures have shown that this influence for the realistic, and thus relevant, models is marginal even though there are significant differences in the graph properties.

9.2 Perspectives

Market relevance of the proposed model

This thesis has a practical orientation in terms that it aims at an actual implementation of the proposed content distribution model. In this context the first steps have been taken already. As a result of the research performed in the scope of this thesis, a business related study has been started performed by a consortium of three Luxembourgish institutions. In particular by the *Puplic Research Centre Henri Tudor*, *Cybercultus* and *SES TechCom*. That study is funded by *Luxinnovation*, the *National Agency for Innovation and Research* in Luxembourg and examines the market demand and acceptance of a service implementing the proposed model. Depending on the results, as the next steps the development of a corresponding prototype might be considered.

Optimizing piece selection strategies for different purposes

Further investigations should be made considering the piece selection strategies. Here a dynamic approach depending on the nature of the currently downloaded file might be preferable. Especially if in the future the percentage of sat-enabled clients should increase, it might become interesting to use a sequential piece download order for all files that have a high probability for being broadcasted. For informing peers about the latter, the tracker could send a corresponding strategy instruction via broadcast to all sat-enabled peers.

Simulations of the model at a larger scale

During the specification of the content distribution model it has been highlighted that its general conception suggests an increasing performance gain with growing numbers of users and files as well as file sizes. Many of the results presented in the course of this work support this assumption. However, such a complex system of networks might show unexpected behavior at larger scales. Thus a further investigation with significantly higher values for these parameters is recommended. However, considering current hardware and the finiteness of the resource time this is a challenging, potentially even impossible task. At least if millions of users and files in combination with file sizes up to tens of gigabytes are intended to be analyzed.

Immediate implementation depending of economical aspects

On the other hand the results presented in this thesis might be sufficient to justify the implementation of this hybrid network content distribution model. This largely depends on the economical appraisal of these figures and on the relation of unicast to broadcast message cost, which has not been in the scope of this thesis. A technical implementation involves building a prototype of the HGW as well as the implementation of the Social SatTorrent protocol by means of client and tracker development. Alternatively also other, newly developed protocols can be used that comply with the model.

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

API .... application programming interface

AS .... autonomous system

BS .... bitfield suppression

BTRM .... broadcast tracker response message

BTRM-E ..... extended broadcast tracker response message

CDN .... content delivery network

CP .... content provider

DNS .... domain name system

DTH .... direct to home

EGR .... extended GetRequest

FNR .... National Research Fund

GEO .... Geostationary Earth Orbit

GPS .... Global Positioning System

HGW .... home gateway

IANA .... Internet Assigned Numbers Authority

ISP .... Internet service provider

IXP .... Internet exchange Points

LAN .... local area network

LEO .... Low Earth Orbit

MEO .... Medium Earth Orbit

OCL .... open connection limit

OSN .... online social network

P2P .... peer-to-peer

PBM .... broadcast piece message

PT .... prefetching threshold

PWM .... peer wire message

SPOF .... single point of failure

SPR .... sat-peer ratio

SST .... Social SatTorrent

SUS .... satellite uplink station

TBCT .... tracker broadcast compensation threshold

UGC .... user generated content

WWW .... World Wide Web

Bibliography

[1] Akamai's technology overview: Web application performance monitoring tools for online businesses. http://www.akamai.com/html/technology/.

[2] Bittorrent - wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/BitTorrent.

[3] Cisco forecast Widget, 2012.

[4] Abdolreza Abhari and Mojgan Soraya. Workload generation for YouTube. *Multimedia Tools and Applications*, 46(1):91–118, June 2009.

[5] Eytan Adar and BA Huberman. Free riding on gnutella. *First Monday*, pages 1–22, 2000.

[6] Vaneet Aggarwal, Robert Caldebank, Vijay Gopalakrishnan, Rittwik Jana, K. K. Ramakrishnan, and Fang Yu. The effectiveness of intelligent scheduling for multicast video-on-demand. *Proceedings of the seventeen ACM international conference on Multimedia - MM '09*, page 421, 2009.

[7] Adrian Andronache, Matthias R Brust, and Steffen Rothkugel. Multimedia content distribution in hybrid wireless networks using weighted clustering. In *Proceedings of the 2nd ACM international workshop on Wireless multimedia networking and performance modeling - WMuNeP '06*, page 1, New York, New York, USA, June 2006. ACM Press.

[8] Andrea Apolloni, Karthik Channakeshava, Lisa Durbeck, Maleq Khan, Chris Kuhlman, Bryan Lewis, and Samarth Swarup. A Study of Information Diffusion over a Realistic Social Network Model. *2009 International Conference on Computational Science and Engineering*, pages 675–682, 2009.

[9] M.F. Arlitt and C.L. Williamson. Internet Web servers: workload characterization and performance implications. *IEEE/ACM Transactions on Networking*, 5(5):631–645, 1997.

[10] Arthurs C. Clarke. Proposal for Geostationary Satellite Communications. *Wireless World*, 1945.

[11] Sitaram Asur and Bernardo A. Huberman. Predicting the Future with Social Media. *Applied Energy*, March 2010.

[12] Lars Backstrom, Eric Sun, and Cameron Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *WWW '10 Proceedings of the 19th international conference on World wide web*, pages 61–70, New York, NY, USA, 2010. ACM.

[13] Fred (Cisco Systems) Baker. RFC1812 - Requirements for IP Version 4 Routers, 1995.

[14] A.L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211, 2005.

[15] Albert-László Barabási, Réka Albert, and Hawoong Jeong. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 281(1-4):69–77, June 2000.

[16] B. Beverly Yang and H. Garcia-Molina. Designing a super-peer network. In *Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405)*, pages 49–60. IEEE, 2003.

[17] Ashwin Bharambe, John R. Douceur, Jacob R. Lorch, Thomas Moscibroda, Jeffrey Pang, Srinivasan Seshan, and Xinyu Zhuang. Donnybrook: enabling large-scale, high-speed, peer-to-peer games. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication - SIGCOMM '08*, page 389, New York, New York, USA, 2008. ACM Press.

[18] Robert M Bond, Christopher J Fariss, Jason J Jones, Adam D I Kramer, Cameron Marlow, Jaime E Settle, and James H Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295–8, September 2012.

[19] Jean Botev, Alexander Hohfeld, Hermann Schloss, Ingo Scholtes, Peter Sturm, and Markus Esch. The HyperVerse: concepts for a federated and Torrent-based '3D Web'. *International Journal of Advanced Media and Communication*, 2(4):331, 2008.

[20] JS Breese, D Heckerman, and C Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth ...*, 1998.

[21] Matt Britt. Partial map of the Internet, 2005.

[22] Anders Brodersen. YouTube Around the World : Geographic Popularity of Videos. In *WWW '12 Proceedings of the 21st international conference on World Wide Web*, pages 241–250, New York, NY, USA, 2012. ACM.

[23] Vincent Buskens. The social structure of trust. *Social Networks*, 20(3):265–289, July 1998.

[24] Claudio Castellano, Santo Fortunato, and Vittorio Loreto. Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2):591–646, 2009.

[25] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 1–14, New York, NY, USA, 2007. ACM.

[26] Xu Cheng, Jiangchuan Liu, and Haiyang Wang. Accelerating YouTube with video correlation. In *Proceedings of the first SIGMM workshop on Social media - WSM '09*, page 49, New York, New York, USA, 2009. ACM Press.

[27] Cisco. Cisco Visual Networking Index : Forecast and Methodology , 2011 - 2016. Technical report, Cisco, 2012.

[28] T Clausen and P Jacquet. RFC3626 - Optimized Link State Routing Protocol (OLSR), 2003.

[29] Bram Cohen. The BitTorrent Protocol Specification, 2009.

[30] European Commission. Galileo - What do we want to achieve ?, 2012.

[31] Richard I Cook. How Complex Systems Fail Cognitive technologies Laboratory. Technical report, Cognitive technologies Laboratory, University of Chicago, 2000.

[32] Riley Crane and Didier Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences of the United States of America*, 105(41):15649–15653, 2008.

[33] M.E. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1997.

[34] DE-CIX. IXP Traffic Statistics, 2012.

[35] Benjamin Doer, Mahmoud Fouz, and Tobias Friedrich. Why rumors spread so quickly in social networks. *Communications of the ACM*, 55(6):70, June 2012.

[36] Fady Draidi, Esther Pacitti, and Bettina Kemme. P2Prec: a P2P recommendation system for large-scale data sharing. In Abdelkader Hameurlain, Josef Küng, and Roland Wagner, editors, *Transactions on large-scale data- and knowledge-centered systems III*, pages 87–116. Springer-Verlag, Berlin, Heidelberg, 2011.

[37] Fady Draidi, Esther Pacitti, Didier Parigot, and Guillaume Verger. P2Prec: a social-based P2P recommendation system. In *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, page 2593, New York, New York, USA, 2011. ACM Press.

[38] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational IP networks: methodology and experience. *IEEE/ACM Transactions on Networking*, 9(3):265–279, June 2001.

[39] Flavio Figueiredo, Fabrício Benevenuto, and J.M. Almeida. The tube over time: characterizing popularity growth of youtube videos. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 745–754. ACM, 2011.

[40] Agata Fronczak, Piotr Fronczak, and Janusz HoÅ‚yst. Mean-field theory for clustering coefficients in Barabási-Albert networks. *Physical Review E*, 68(4):046126, October 2003.

[41] Wojciech Galuba, Karl Aberer, Z. Despotovic, and W. Kellerer. Leveraging social networks for increased BitTorrent robustness. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1–5. IEEE, 2010.

[42] P. Garbacki, A. Iosup, D. Epema, and M. van Steen. 2Fast : Collaborative Downloads in P2P Networks. In *Sixth IEEE International Conference on Peer-to-Peer Computing (P2P'06)*, pages 23–30, Cambridge, UK, 2006. IEEE.

[43] Pawel Garbacki, D.H.J. Epema, Johan Pouwelse, and M. Van Steen. Offloading servers with collaborative video on demand. In *Proceedings of the 7th international conference on Peer-to-peer systems*, pages 6–6. USENIX Association, 2008.

[44] Natasha Gilani. What Are Hybrid Networks?, 2012.

[45] Christos Gkantsidis, Thomas Karagiannis, and Milan VojnoviC. Planet scale software updates. *ACM SIGCOMM Computer Communication Review*, 36(4):423, August 2006.

[46] Google. YouTube Facts and stats, 2010.

[47] Michel Grabisch and Agnieszka Rusinowska. A model of influence in a social network. *Theory and Decision*, 69(1):69–96, April 2008.

[48] C. (Rutgers University) Hedrick. RFC1058 - Routing Information Protocol, 1988.

[49] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99*, pages 230–237, New York, New York, USA, 1999. ACM Press.

[50] Günther A. (Humboldt University of Berlin) Hoffmann. *Failure prediction in complex computer systems: A probabilistic approach.* PhD thesis, Humboldt University of Berlin, 2006.

[51] Thomas Hofmann and Wolfgang Nejdl. Robust Collaborative Filtering -. *Computer*, 2007.

[52] Alexander Höhfeld, Patrick Gratz, Angelo Beck, Jean Botev, Hermann Schloss, and Ingo Scholtes. Self-organizing collaborative filtering in global-scale massive multi-user virtual environments. In *Proceedings of the 2009 ACM symposium on Applied Computing - SAC '09*, page 1719, New York, New York, USA, 2009. ACM Press.

[53] Ian Horrocks. Ontologies and the semantic web. *Communications of the ACM*, 51(12):58, December 2008.

[54] Daniel Hughes, G. Coulson, and J. Walkerdine. Free Riding on Gnutella Revisited: The Bell Tolls? *IEEE Distributed Systems Online*, 6(6):1–1, June 2005.

[55] Pan Hui and Sonja Buchegger. Groupthink and Peer Pressure: Social Influence in Online Social Network Groups. In *2009 International Conference on Advances in Social Network Analysis and Mining*, pages 53–59. IEEE, July 2009.

[56] C Huitema. *Routing in the Internet.* CCIE Resource Library. Prentice Hall PTR, 2nd edition, 2000.

[57] The Interactive Advertising Bureau (IAB). IAB Internet Advertising Revenue Report - Full Year 2011. Technical Report April, The Interactive Advertising Bureau (IAB), 2012.

[58] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-molina. Incentives for Combatting Freeriding on P2P Networks . In *In Proc. EURO-PAR*, 2003.

[59] Thomas Karagiannis, Pablo Rodriguez, and Konstantina Papagiannaki. Should internet service providers fear peer-assisted content distribution? In *Proceedings of the 5th ACM SIGCOMM conference on Internet measurement - IMC '05*, page 1, New York, New York, USA, 2005. ACM Press.

[60] Murat Karakaya, Ibrahim Korpeoglu, and Özgür Ulusoy. Free Riding in Peer-to-Peer Networks. *IEEE Internet Computing*, 13(2):92–98, March 2009.

[61] Murat Karakaya, Ä°brahim KörpeoÄŸlu, and Özgür Ulusoy. Counteracting free riding in Peer-to-Peer networks. *Computer Networks*, 52(3):675–694, February 2008.

[62] Henry Kautz, Bart Selman, and Mehul Shah. Referral Web: combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, March 1997.

[63] Bernd Klasen. Efficient Content Distribution in Social-Aware Hybrid Networks. *Journal of Computational Science*, 2011.

[64] Bernd Klasen. Social, fast, efficient: Content distribution in hybrid networks. In *2011 IEEE Symposium on Computers and Communications (ISCC)*, pages 61–67, Kerkyra, Greece, June 2011. IEEE.

[65] Bernd Klasen. SatTorrent - Satellite-Aided P2P Content Distribution for Virtual Environments. In *Proceedings of the Fifth International Conference on Simulation Tools and Techniques*, Desenzano, Italy, 2012. ACM.

[66] Bernd Klasen. Content Delivery in Hybrid Networks Using SatTorrent. In *5th International Conference on Personal Satellite Services (PSATS 2013) (accepted for publication)*, 2013.

[67] Bernd Klasen. Efficient Integration of Satellites in Collaborative Network Structures. In Prashant Pillai, Rajeev Shorey, and Erina Ferro, editors, *Proceedings of 4th International ICST Conference on Personal Satellite Services*, pages 130–137. Springer Berlin Heidelberg, Bradford, UK, 2013.

[68] Bernd Klasen. SocioAware Content Distribution using P2P solutions in Hybrid Networks. In *The 18th IEEE Symposium on Computers and Communications (ISCC 2013) (accepted for publication)*, Split, Croatia, 2013.

[69] Bernd Klasen, Alexander Vinzl, and Takeshi Martinez. Shopping Assistant. In Petros Daras and Oscar Mayora Ibarra, editors, *User Centric Media First International Conference, UCMedia 2009, Venice, Italy, December 9-11, 2009, Revised Selected Papers*, pages 214–217. Springer Berlin Heidelberg, 2010.

[70] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M Jose. On social networks and collaborative recommendation. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '09*, page 195, New York, New York, USA, 2009. ACM Press.

[71] Dmitri Krioukov, k c Claffy, Kevin Fall, and Arthur Brady. On compact routing for the internet. *ACM SIGCOMM Computer Communication Review*, 37(3):41, July 2007.

[72] Gijs Kruitbosch and Frank Nack. Broadcast Yourself on YouTube - Really ? In *Proceeding of the 3rd ACM international workshop on Human-centered computing*, pages 7–10. ACM, 2008.

[73] James F Kurose and Keith W Ross. *Computer Networking: A Top-Down Approach*. Addison-Wesley Publishing Company, USA, 5th edition, 2009.

[74] Arnaud Legout, G Urvoy-Keller, and P Michiardi. Rarest first and choke algorithms are enough. In *Proceedings of the 6th ACM SIGCOMM on Internet measurement - IMC '06*, page 203, New York, New York, USA, 2006. ACM Press.

[75] Gregor Maier, Anja Feldmann, Vern Paxson, and Mark Allman. On dominant characteristics of residential broadband internet traffic. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference - IMC '09*, volume 9, page 90, New York, New York, USA, 2009. ACM Press.

[76] Gary Scott Malkin. RFC2453 - RIP Version 2, 1998.

[77] MJ Miller, WD List, and NH Vaidya. A hybrid network implementation to extend infrastructure reach. Technical Report September 2002, 2003.

[78] Bradley Mitchell. Do Wireless Routers Support Hybrid Networks?, 2012.

[79] Melanie Mitchell. *Complexity: A Guided Tour*. Oxford University Press, New York, NY, 2009.

[80] John Moy. RFC2328 - OSPF Version 2, 1998.

[81] Nielsen. State of the Media - 2010 - US Audiences and Devices.pdf. Technical report, The Nielsen Company, 2011.

[82] Nielsen. The U . S . Media Universe. Technical report, The Nielsen Company, 2012.

[83] David Oran. RFC1142 - OSI IS-IS Intra-domain Routing Protocol, 1990.

[84] Athanasios D. Panagopoulos, Pantelis-Daniel M. Arapoglou, and Panayotis G. Cottis. Satellite communications at KU, KA, and V bands: Propagation impairments and mitigation techniques. *IEEE Communications Surveys & Tutorials*, 6(3):2–14, 2004.

[85] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Physics*, page 13, October 2000.

[86] Vern Paxson. End-to-end routing behavior in the internet. *ACM SIGCOMM Computer Communication Review*, 36(5):41, October 2006.

[87] J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. R. van Steen, and H. J. Sips. TRIBLER: A social-based peer-to-peer system. *Concurrency and Computation: Practice & Experience - Recent Advances in Peer-to-Peer Systems and Security*, 20(2):127–138, 2008.

[88] Lakshmish Ramaswamy. Free riding: a new challenge to peer-to-peer file sharing systems. In *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, page 10 pp. IEEE, 2003.

[89] Yakov Rekhter, Tony Li, and Susan Hares. RFC4271 - A Border Gateway Protocol 4 (BGP-4), 2006.

[90] Dave Roos. How Hybrid Networks Work, 2012.

[91] Stefan Saroiu, P Krishna Gummadi, and Steven D Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. In *Proceedings of the Multimedia Computing and Networking (MMCN)*, 2002.

[92] Badrul Sarwar, George Karypis, Joseph Konstan, and John Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the tenth international conference on World Wide Web - WWW '01*, pages 285–295, New York, New York, USA, 2001. ACM Press.

[93] Salvatore Scellato, Renaud Lambiotte, Cecilia Mascolo, and A Noulas. Socio-spatial properties of online location-based social networks. In *Proceedings of Fifth International AAAI Conference on Weblogs and Social Media (ICWSM 2011)*, Barcelona, Catalonia, Spain, 2011. The AAAI Press.

[94] Salvatore Scellato, Cecilia Mascolo, Mirco Musolesi, and Jon Crowcroft. Track Globally , Deliver Locally : Improving Content Delivery Networks by Tracking Geographic Social Cascades Categories and Subject Descriptors. In *Proceedings of WWW '11 Proceedings of the 20th international conference on World wide web*, pages 457–466, New York, NY, USA, 2011. ACM.

[95] Upendra Shardanand and Pattie Maes. Social information filtering. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95*, pages 210–217, New York, New York, USA, 1995. ACM Press.

[96] Hans-Peter Siebenhaar. Online-Werbung uebertrifft erstmals TV-Reklame, 2012.

[97] Betsy Sparrow, Jenny Liu, and Daniel M Wegner. Google effects on memory: cognitive consequences of having information at our fingertips. *Science (New York, N.Y.)*, 333(6043):776–8, August 2011.

[98] Ao-jan Su, David R Choffnes, Aleksandar Kuzmanovic, and E Bustamante. Drafting Behind Akamai âˆ— (Travelocity-Based Detouring) Categories and Subject Descriptors. *Measurement Techniques*, 2006.

[99] Gabor Szabo and Bernardo a. Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80, August 2010.

[100] Andrew S. Tanenbaum. *Computer networks*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 3rd edition, 1996.

[101] Teledesic. Fast Facts about Teledesic, 2001.

[102] Theory.org Wiki contributors. Bittorrent Protocol Specification v1.0. http://wiki.theory.org/BitTorrentSpecification, 2013.

[103] R. Toivonen, J. P. Onnela, J. Saramäki, J. Hyvönen, and K. Kaski. A Model for Social Networks. *Physica A: Statistical and Theoretical Physics*, 371(2):851–860, November 2006.

[104] Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The Anatomy of the Facebook Social Graph. *The Computing Research Repository*, page 17, November 2011.

[105] Princeton University. Sidereal Time, 2012.

[106] Wernher von Braun. TV Broadcast Satellite. *Popular Science*, pages 64–66, 1970.

[107] Zhi Wang, Lifeng Sun, Shiqiang Yang, and Wenwu Zhu. Prefetching strategy in peer-assisted social video streaming. In *MM '11 Proceedings of the 19th ACM international conference on Multimedia*, pages 1233–1236, New York, New York, USA, 2011. ACM Press.

[108] D J Watts and S H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–2, June 1998.

[109] Duncan J. Watts. *Six Degrees: The Science of a Connected Age*. W. W. Norton & Company, 2003.

[110] Hao Yin, Xuening Liu, Tongyu Zhan, Vyas Sekar, Feng Qiu, Chuang Lin, Hui Zhang, and Bo Li. Design and deployment of a hybrid CDN-P2P system for live video streaming. In *Proceedings of the seventeen ACM international conference on Multimedia - MM '09*, page 25, New York, New York, USA, 2009. ACM Press.

[111] YouTube-Team. At five years, two billion views per day and counting, 2010.

[112] Yipeng Zhou, Dah-ming Chiu, John C. S. Lui, and Abstract Peer-to-peer Pp. A Simple Model for Chunk-Scheduling Strategies in P2P Streaming. *IEEE/ACM Transactions on Networking*, 19(1):42–54, February 2011.

Index