Julius-Maximilians-Universität Würzburg
Institut für Informatik
Lehrstuhl für Kommunikationsnetze
Prof. Dr.-Ing. P. Tran-Gia

Performance Evaluation and Optimization of Hierarchical Content Delivery Networks

Valentin Burger

Würzburger Beiträge zur
Leistungsbewertung Verteilter Systeme

Bericht 1/16

Würzburger Beiträge zur

Leistungsbewertung Verteilter Systeme

Herausgeber

Prof. Dr.-Ing. P. Tran-Gia
Universität Würzburg
Institut für Informatik
Lehrstuhl für Kommunikationsnetze
Am Hubland
D-97074 Würzburg

Tel.: +49-931-31-86630
Fax.: +49-931-31-86632
email: trangia@informatik.uni-wuerzburg.de

Satz

Reproduktionsfähige Vorlage des Autors.
Gesetzt in LaTeX Linux Libertine 10pt.

ISSN 1432-8801

Performance Evaluation and Optimization of Hierarchical Content Delivery Networks

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Julius–Maximilians–Universität Würzburg

vorgelegt von

Valentin Burger

aus

Kronach

Würzburg 2016

Eingereicht am: 22.01.2016

bei der Fakultät für Mathematik und Informatik

1. Gutachter: Prof. Dr.-Ing. Phuoc Tran-Gia

2. Gutachter: Prof. Dr. Ulf Körner

Tag der mündlichen Prüfung: 20.04.2016

Contents

1 Introduction

http://www.laneas.com/sites/default/files/publications/1/PachosEtAl2016Misconceptions.pdf In the 90s, the traffic in the web exploded, leading its inventor Sir Tim Berners-Lee to declare the network congestion as one of the main challenges for the Internet of the future. The congestion was caused by the dotcom boom and specifically due to the client-server model of connectivity, whereby a webpage was downloaded from the same network server by every Internet user in the world. The challenge was ultimately resolved by the invention of Content Delivery Networks (CDNs), and the exploitation of web caching. The latter replicates popular content in many geographical areas and saves bandwidth by avoiding unnecessary multihop retransmissions. As a byproduct, it also decreases access time (latency) by decreasing the distance between the two communicating entities. Today's Internet

fog computing

One example of such a scenario are This results in a clash of interests which negatively impacts the network performance for all participants.

The goal of this monograph is To this end, we study If such an optimisation is not possible or attempts at it might lead to adverse effects, we discuss the reasons.

In the remainder of this chapter we provide an overview over the scientific contributions of this monograph in Section 1.1 and provides an outline of this monograph in Section 1.2.

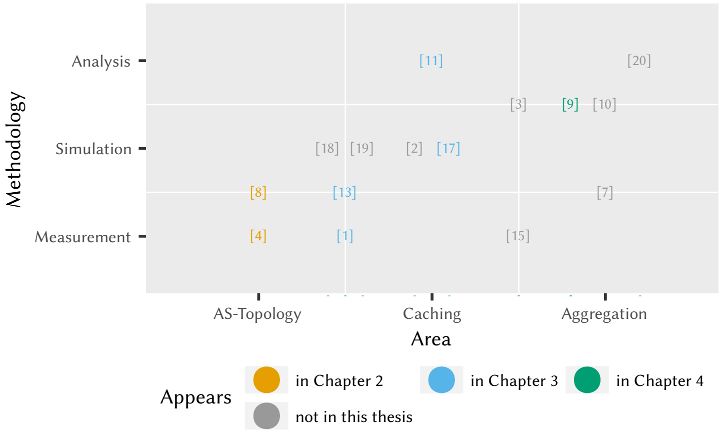


Figure 1.1: Contribution of this work as a classification of the research studies conducted by the author.

1.1 Scientific Contribution

This monograph studies the interactions between different stakeholders in three, partially overlapping, scenarios in order to provide an overview of today's interlocking network and application ecosystem.

In Figure 1.1 we classify the areas of research as well as scientific methods used in relation to the chapters of this monograph. The x-axis shows the impacted areas of research, i.e. topics related to the mobile network, the application domain or cloud technologies. The y-axis details the applied scientific method. In the theoretical area methods from queueing theory, mean value analysis and the analysis of random variables are used. Measurements were performed using testbeds and custom software tools. Simulation studies, performed using Discrete Event Simulation (DES), and created analysis tools are summarised in the practical area.

Annotations are used to highlight scientific publication whose content contributes to the respective chapters.

The first contribution of this monograph is a discussion of the impact of We study the impact of , and investigate the potential of network parameter optimisation as a means to .

As a second contribution, we provide models for We show that the streaming mechanism allows However, further study shows that in fact Furthermore, we provide

As a third contribution, we discuss the To this end, we study the performance of We derive guidelines for Furthermore, we discuss a mechanism to Finally, we present a mechanism to

1.2 Outline of Thesis

In Chapter 2 we study First, Then, Finally,

Chapter 3 focuses on For Video Streaming, we study the impact To address the second scenario, we discuss

In Chapter 4 we study First, Then, We analyse traffic characteristics and use them as input for a simulation model Combining these results, we evaluate the impact of different virtual server configurations and scaling strategies. Finally, we consider resource allocation

In Chapter 5 we provide a summary of the major contributions of this work and suggest future potential research directions.

2 Characterization of Content Delivery Networks on Autonomous System Level

While in the past a large fraction of the total Internet traffic was carried by peer-to-peer (P2P) networks [23], the largest fraction of Internet traffic is now carried by content delivery networks (CDNs). In 2015 61% of all Internet video was carried by CDNs [24]. The traffic from CDNs is highly asymmetric and produces a large amount of costly inter-domain traffic [25]. Inter-domain traffic is routed among different autonomous systems (ASs) that are individual parts of the Internet operated by an Internet Service Providers (ISP). Especially ISPs providing access to many end users have problems to deal with the huge amount of traffic originating from CDNs. Reducing the traffic carried by CDNs and the load put on ISP networks has high potential to reduce energy consumption and cost for content delivery. In order to develop traffic management mechanisms, aiming to reduce inter-domain traffic and to optimize content delivery, it is crucial to understand the current situation of CDNs and the number and distribution of available resources.

In this chapter, we show the principles and recent developments of content delivery networks and characterize their resources on AS level by performing and evaluating distributed measurements. The models derived are used to characterize the traffic in P2P networks and to develop optimization strategies to reduce inter-domain traffic and transit costs.

In order to characterize P2P networks on AS level, we analyze a measurement study and identify the number of peers per swarm available in each AS and investigate the performance of different peer selection strategies. As key performance indicators we determine the traffic volume and the traffic costs of ISPs.

Due to the location based server assignment in CDNs, a distributed measurement architecture is necessary to identify the server resources. Typically distributed measurement platforms such as PlanetLab are used for that purpose. The problem is that these measurement platforms may not reflect the perspective as consumer, since they are hosted in National Research and Education Networks (NRENs) and not in ISP networks. We compare the capabilities of a crowdsourcing platform and a PlanetLab testbed for distributed active measurements.

Recent approaches [26] suggest to use resources on customer premise equipment (CPE) such as home routers or network attached storage to support content delivery while saving energy. The potential of such peer assisted content delivery approaches depends on the number of subscribers in the ISPs network. To get a global view on the number of resources available in each AS we evaluate the Internet Census Dataset, which contains a complete scan of the IP address range.

The models derived from the measurement results of the CDN YouTube and P2P networks can be used for performance analysis and optimization of content delivery mechanisms. In particular the models serve as input for the optimization strategies for peer selection and the performance evaluation of hierarchical CDNs in realistic parameter studies in Chapter 3.

The content from this chapter has been published in [4, 8, 11]. Section 2.1 describes the evolution and structure of content delivery networks. We present measurements and models of CDNs in Section 2.2. In Section 2.3 two approaches for distributed active measurements are compared towards their capability to probe servers in CDNs and the global CDN of YouTube is characterized. The applied methodology to characterize P2P traffic and to estimate its transit costs is described in Section 2.4. We discuss lessons learned in Section 2.5.

2.1 Evolution and Structure of Content Delivery Networks

This section describes the background of this chapter and presents related work. We start with an introduction in ISP relations and charging models. We provide the basic ideas of content delivery network structures and their evolution. We briefly describe the structure of the YouTube video CDN. Finally we evaluate the Internet Census Dataset to assess the potential of peer assisted CDNs.

2.1.1 ISP Relations and Charging Models

On a technical level, the traffic exchange between ASs is controlled by the Border Gateway Protocol (BGP) [27]. Commercial relations between ISPs determine the routing policies configured via BGP.

An ISP must buy transit services to access parts of the Internet it neither owns nor can access by its customers. Hence, to route traffic between ASs, ISPs engage in business relationships. These business relationships are usually not open for public but they can be abstracted into three common types [28]. The relationship between two ASs can be transit, peering or sibling. A transit link is present if the customer AS pays the provider AS for transit service, i.e., the provider forwards the traffic of the customer and its customers. In peering relations the ASs have an agreement that they exchange each others traffic and the traffic of their customers, without paying each other. Sibling links between ASs of the same organization. These relations are defined in business agreements and kept secret, but they can be inferred by analyzing the routing between autonomous systems.

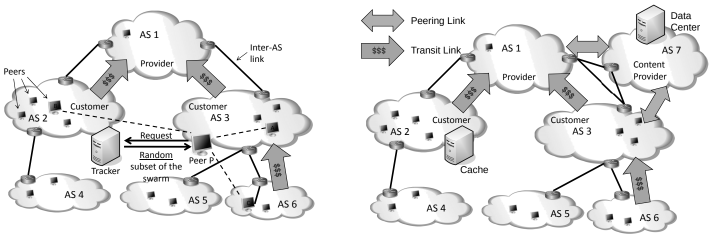
As peering links and sibling links are the same in terms of money flow, we consider sibling links as peering links in this manuscript. For the same reason most datasets available do not contain sibling links. Hence, we only consider transit and peering inter-domain links from now on.

2.1.2 Evolution of Content Delivery Networks

The two most popular solutions to this problem are content delivery concepts that work according to the P2P or CDN principle. To deliver content, it has to be transported from storage resources to clients that consume the content. Figure 2.1 shows the AS topology for a P2P network, and a CDN. In case of P2P networks, c.f., Figure 2.1a, peers contribute storage resources by serving chunks of data. To enable exchange of data, a tracker keeps track of the peers sharing chunks of a file. In the past P2P networks were responsible for a large fraction of Internet traffic. If the exchange of data among peers is uncoordinated, ignoring inter-AS links, P2P networks can produce high costs for ISPs.

Most P2P traffic today is transported using the BitTorrent P2P file-sharing protocol. BitTorrent is based on multi-source downloads between the users. All the users, i.e., *peers*, sharing the same file belong to a *swarm*, c.f., Figure 2.1a. To join the swarm, a peer requests addresses of other peers at an index server called *tracker*. In the standard BitTorrent algorithm the tracker uses random peer selection to select a subset of peers that are in the swarm. Then, the joining peer tries to establish a neighbor relation to the peers it got from the tracker and collects all peers which accepted the request in his *neighbor* set. The peer signals interest to all neighbors which have parts of the file it still needs to download. To which neighbor a peer is willing to upload data is decided by the choking algorithm, which is explained in [29].

While the amount of traffic transported over P2P networks remained about the same, the traffic transported by CDNs has increased exponentially [24]. The number of users watching videos on demand has massively increased and the bandwidth to access videos is much higher. Furthermore, the increased bandwidth enables web services to be interactive by using dynamic server- or client-side scripts. The appearance of dynamic services and the increasing quality of multimedia content raises user expectations and the demand on the servers. To bring content in high quality to end-users with low latency and to deal with increasing demand, content providers have to replicate and distribute the content on caches to get it close to end-users. Hence, in CDNs, c.f., Figure 2.1b, the



(a) P2P network

(b) Content delivery network

Figure 2.1: Autonomous system topology with a P2P network (left) and a CDN (right).

storage resources are managed data centers or caches.

The global expansion of the CDNs also changes the structure of the Internet. Google has set up a global backbone which interconnects Google's data centers to important edge points of presence. Since these points of presence are distributed across the globe, Google can offer direct peering links to access networks with many end users, c.f. Figure 2.1b. Such, access network providers save transit costs, while Google is able to offer services with low latency. To bring content even closer to users, ISPs can deploy Google cache servers inside their own network to serve popular content, including YouTube videos [30].

To select the closest server for a content request and to implement load balancing, CDNs use the Domain Name System (DNS). Typically a user watches a YouTube video by visiting a YouTube video URL with a web browser. The browser then contacts the local DNS server to resolve the hostname. Thereafter, the HTTP request is directed to a front end web server that returns an HTML page including URLs for default and fallback video servers. These URLs are again resolved by DNS servers to physical video servers, which stream the content. The last DNS resolution can happen repeatedly until a server with enough capacity is found to serve the request. Thus, load balancing between the servers is achieved [31].

Current developments try to bring content even closer to users by using resources in home networks or mobile networks cells for caching. A recent approach [26] proposes to augment spare capacities on customer premise equipment (CPE) such as home routers or nano data-centers (NaDas) to assist content delivery, showing that there is a high potential to save energy, although the capacity of home routers is small and the uplink is limited. The content is transported locally in a P2P manner keeping the traffic within the AS. Another example are femto caching architectures [32], where content is cached on femto-basestations with small capacity but with considerable storage space. The potential of these approaches highly depends on the number of caches available and their capacity for content delivery.

2.1.3 Characterization of Internet Subscriptions on AS Level

The performance of systems using CPE or resources provided by end-users depend on the capacity and number of devices available. To assess the potential of a hierarchical cache system in an ISPs network, the number of active subscribers in an AS has to be known. Assuming that the number of active IP-addresses is correlated to the number of subscribers in an AS, we use the Internet Census Dataset to determine the distribution of active IP-addresses on ASs.

We use the Internet Census Dataset [33] to determine the number of active IP-addresses for each AS in the Internet. The Internet Census Dataset provides a scan on the active IP-addresses in the Internet based on a full probing of the entire IPv4 Internet. The scan was conducted from June to October 2012 by infecting several hundred thousand unprotected devices on the Internet to form the so called *carna* botnet. The botnet functioned as distributed port scanner that transfered the results to a central server.

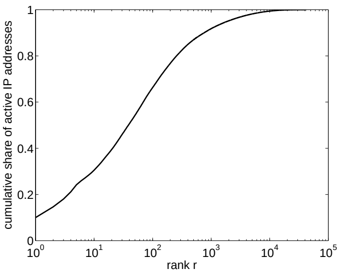
In the ICMP ping scan more than 420 million replied to requests more than once. The service probe data reveal open ports on devices which is used to infer the type of device. The Internet Census Dataset was validated forensically in [34] by aligning the probes of the botnet with the traffic captured at the UCSD Network Telescope[1], which is a large darknet, i.e., IP addresses that are inactive, thus not accepting connections. The raw logs of the carna botnet erroneously reported that a large number of IPs in the darknet were active, likely due to the presence of HTTP proxies. However, according to [34], only about 3% of the host probe and port scan logs are potentially affected by this problem. Unaffected by this issue are the logs based on ICMP pings and actual responses from the target hosts, which are used in our study. In [35] the scope of the dataset is taken into perspective and show that, although there are some qualitative problems, the measurement data seems to be authentic.

We use an IP to ASN mapping to derive the autonomous system number for each IP-address. There are different services, that provide an IP to ASN mapping. The whois-service can be used to get the current ASN for an IP-address. To enable an efficient evaluation we used the MAXMIND GeoLite ASN database [36], which is updated every month and can be downloaded and used as a local database. The results of the MAXMIND GeoLite ASN database were cross checked with results obtained from whois, which showed no differences.

The ICMP ping scan discovered a total of 598,180,914 IP-addresses. The service probe scan discovered 244,000 IP-addresses that listen to port 9100 and are identified as print servers, and 70.84 million IP-addresses of web-servers that listen to port 80. Assuming that most network functions do not reply to ICMP ping requests and neglecting different network functions, this results in 88.1% of IP-addresses assigned to end-user devices. Since the Internet Census the number of Internet users increased, which also has to be considered. According to [37] there is a 7% annual increase in fixed-broadband subscriptions in the past three years.

Figure 2.2 shows the cumulative share of active IP-addresses in the autonomous systems ranked in descending order. The 100 largest autonomous systems make up 2/3 of active IPs and more than 85% of the IPs are active in only 1%

[1] https://www.caida.org/projects/network_telescope/



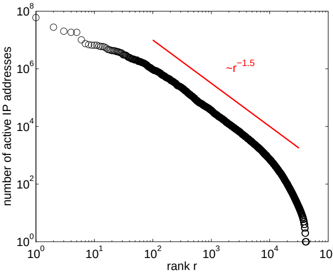


Figure 2.2: Cumulative share of active IPs in autonmous systems ranked in descending order.

Figure 2.3: Rank of Internet providers with number of active IPs per AS.

of the autonomous systems. The 10 largest autonomous systems already contain 30% of the active IPs.

Figure 2.3 shows the number of active IP-addresses per AS ranked in descending order. The top 5 ASs are shown in table 2.1. The AS with most active IP-addresses is ChinaTelecom with almost 60 million active IPs, followed by another Chinese provider. The largest AS in the US is Comcast on rank three. The largest Korean and German providers are ranked 4 and 5 with more than 18 million active IPs. The number of active IP addresses can be approximated with a power law with slope 1.5 that drops a little for low ranks. This shows that the distribution of active IP addresses on ASs is highly heterogeneous. That means the potential of approaches leveraging spare resources on home routers depends on the AS.

2.2 Measurements and Models of Content Delivery Networks

The most popular P2P overlay network today is BitTorrent. BitTorrent is still responsible for a large portion of Internet traffic [24, 38]. In Section 2.2.1 we give

Table 2.1: Rank of top 5 provider with most active IP-addresses.

rank r	ASN	provider	# active IPs
1	4134	ChinaTelecom	59,824,824
2	4837	China-Network-Communication-Group	27,776,643
3	7922	Comcast	20,227,918
4	4766	KoreaTelecom	18,502,963
5	3320	DeutscheTelekomAG	18,476,519

an overview on measurement studies of live BitTorrent networks and show different approaches to reduce costly traffic discussed in the ALTO working group of the IETF.

We summarize related work in the field of distributed active measurements of CDNs and give a short introduction in the principles of crowdsourcing in Section 2.2.2.

2.2.1 Measurements and Models of BitTorrent Networks

As basis of our methodology for modeling inter-ISP BitTorrent traffic in Section 2.4, the results in [39] are revisited. The authors provide measurements of a large number of live BitTorrent swarms taken from popular index servers such as *The Pirate Bay*, *Mininova*, and *Demonoid*. Using the IP addresses of the peers, the authors associate every peer with its AS and estimate the potential of ALTO mechanisms based on the differentiation between local peers (peers in the same AS) and remote peers located in other ASs. In contrast, we consider the actual Internet topology in this work, i.e., the inter-ISP relations, the ISP classification in the Internet hierarchy, and the AS paths between the peers in order to estimate the optimization potential of ALTO mechanisms.

The authors of [40] use the peer exchange protocol (PEX) in order to measure the neighbor set of all peers participating in a number of live BitTorrent swarms. Based on this information, they model the graph topology of the swarms and compare the structure to random graphs. They also investigate clustering of peers within ASs and countries, but do not focus on inter-AS relations and AS paths between peers as we do in this chapter.

In addition, there are measurement studies that examine and model distinct features of BitTorrent networks. In [41], a single swarm was measured for five months with a focus on the download times of the peers. Additional parameters such as the peer inter-arrival times in the swarm, their upload capacity and their online time are considered in [42]. The authors of [43] investigate these parameters also in multi-swarm scenarios. Finally, [44] measures 4.6 million torrents to provide an overview of the entire BitTorrent ecosystem with its different communities and index servers. While the distribution of peers in the Internet is also studied in this chapter, none of these works focuses on the location of the peers in the Internet and the AS paths between the peers, which is a major aspect of this chapter.

Various mechanisms to reduce the inter-ISP traffic generated by BitTorrent and other P2P applications are currently being investigated. Besides caching of BitTorrent traffic [45–47], which might involve legal issues, changing standard BitTorrent algorithms is a promising approach. The authors of [48] propose to use an oracle service provided by the ISP guiding the peers in their peer selection process. The evaluation uses a Gnutella network and shows that intra-AS traffic is increased significantly without a negative impact on the overlay graph. Similar approaches are proposed for BitTorrent. Bindal et al. [49] reduce the inter-ISP traffic by modifying the neighbor set of the BitTorrent peers, which can be done at the tracker or enforced by the ISPs using deep packet inspection. Their simulations use a uniform peer distribution over ASs and show a high optimization potential of this approach. The authors of [50] propose to use *iTrackers* to guide the peers and formulates an optimization problem to find the best neighbor sets. Finally, Oechsner et al. [51] propose to change the choke algorithm of BitTorrent to further reduce inter-ISP traffic and evaluate it via simulations in homogeneous scenarios. The BitTorrent plugin *Ono* [52] uses the servers of CDNs as landmarks and estimates the proximity of two peers by the similarity of the CDN re-direction behavior.

The authors of [53] investigate analytically the capabilities of a P2P-based content distribution network and the impact of locality. In contrast to our work, they use traffic characteristics which arise from software updates and do not consider AS relationships. A set of evaluations of ALTO mechanisms uses scenarios inspired by measurements of live BitTorrent swarms [54–56]. The studied scenarios consider heterogeneous peer distributions where some ASs contain more peers of a specific swarm than others. Nevertheless, they do not take into account inter-AS relations and the AS paths between two peers. This is different in our study. Using the AS affiliation of peers and the data obtained from Caida.org, we infer the actual paths of the BitTorrent connections in the Internet. In addition, we focus on the inter-ISP relations and investigate to which degree selfish ISPs profit from recommending their peers to preferentially use connections to peers located in lower tier ASs.

2.2.2 Measurements of Content Delivery Networks

There already exist a number of publications which study the structure of the YouTube CDN and its selection of video servers. A distributed active measurement platform is necessary for these evaluations, because the CDN mechanisms consider the client locations, both geographical as well as in terms of the connected access network. Recently, the physical server distribution of NetFlix's CDN was mapped in [57], using a DNS crawler and exploiting the naming scheme of the cache servers. In [58], two university campus networks and three ISP networks were used to investigate the YouTube CDN from vantage points in three different countries.

While the view of five different ISPs on a global CDN is still narrow, the authors of [59] used PlanetLab to investigate the YouTube server selection strategies and load-balancing. They find that YouTube massively deploys caches in many different locations worldwide, placing them at the edge of the Google autonomous system or even at ISP networks. The work is enhanced in [31], where they uncover a detailed architecture of the YouTube CDN, showing a 3-tier physical video server hierarchy. Furthermore, they identify a layered logical structure in the video server namespace, allowing YouTube to leverage the existing DNS system and the HTTP protocol.

However, to assess the expansion of the whole YouTube CDN and its cache locations in access networks, the PlanetLab platform, which is located solely in NRENs, is not suitable, since it does not reflect the perspective of end users in ISP access networks. Therefore, a different distributed measurement platform is used in [60] which runs on end user equipment and thus implies a higher diversity of nodes and reflects the perspective of end user in access networks. However, the number of nodes that was available for the measurement is too small to obtain a global coverage of vantage points

To achieve both, the view of access networks and a high global coverage with a large number of measurement points, the participation of a large number of end users in the measurement is necessary. Bischof et al. [61] implemented an approach to gather data from P2P networks to globally characterize the service quality of ISPs using volunteers.

In contrast to this we propose using a commercial crowdsourcing platform to recruit users running a specially designed measurement software, and therewith, act as measurement probes. Crowdsourcing is an emerging service in the Internet that enables outsourcing jobs to a large, anonymous crowd of users [62]. So called *Crowdsourcing platforms* act as mediator between the users submitting the tasks, the *employers*, and the users willing to complete these tasks, the *workers*. All interactions between workers and employers are usually managed through these platforms and no direct communication exists, resulting in a very loose worker-employer relationship. The complexity of crowdsourcing tasks varies between simple transcriptions of single words [63] and even research and development tasks [64]. Usually, the task descriptions are much more fine granular than in comparable forms in traditional work organizations [65]. This small task granularity holds in particular for *micro-tasks*, which can be completed within a few seconds to a few minutes. These tasks are usually highly repetitive, e.g., adding textual descriptions to pictures, and are grouped in larger units, so called *campaigns*. In comparison to other approaches using volunteers, this approach offers better scalability and controllability, because the number and origin of the participants can be adjusted using the recruiting mechanism of the crowdsourcing platform.

2.3 Content Delivery Network Characterization by Distributed Active Measurements

To understand and monitor the impact of YouTube traffic on ISPs and the topology of CDNs appropriate measurements are acquired. Due to YouTube's load-balancing and caching mechanisms the YouTube video server selection is highly dependent on the location of the measurement points. Hence, we need a globally distributed measurement platform to perform active measurements to uncover the location of YouTube servers. The problem is that probes disseminated from PlanetLab nodes origin solely from National Research and Education Networks (NRENs). This may not reflect the perspective of access ISPs which have a different connection to the YouTube CDN with different peering or transit agreements. To achieve a better view on the YouTube CDN from the perspective of end users in access networks we use a commercial crowdsourcing platform to recruit regular Internet users as measurement probes. This complementary view can help to gain a better understanding of the characteristics of Video CDNs. To evaluate the impact of the measurement platform and the coverage of their vantage point, we perform the same measurements using PlanetLab nodes and crowdsourcing users and compare the obtained results.

The measurements conducted in the PlanetLab and via crowdsourcing are described in Section 2.3.1. In Section 2.3.2 we provide details on the measurement results and their importance for the design of distributed network measurements.

2.3.1 Distributed Active Measurement Setup

To assess the capability of crowdsourcing for distributed active measurements we conduct measurements with both PlanetLab and the commercial Crowdsourcing platform Microworkers [66]. We measure the global expansion of the YouTube CDN by resolving physical server IP-addresses for clients in different locations.

Description of the PlanetLab Measurement

PlanetLab is a publicly available test bed, which currently consists of 1173 nodes at 561 sites. The sites are usually located at universities or research institutes. Hence, they are connected to the Internet via NRENs. To conduct a measurement in PlanetLab a slice has to be set up which consists of a set of virtual machines running on different nodes in the PlanetLab test bed. Researchers can then access these slides to install measurement scripts. In our case the measurement script implemented in Java extracted the server hostnames of the page of three predetermined YouTube videos and resolved the IP addresses of the physical video servers. The IP addresses of the PlanetLab clients and the resolved IP addresses of the physical video servers were stored in a database. To be able to investigate locality in the YouTube CDN, the geo-location of servers and clients is necessary. For that purpose the IP addresses were mapped to geographic coordinates with MaxMinds GeoIP database [67]. The measurement was conducted on 220 randomly chosen PlanetLab nodes in March 2012.

Description of the Crowdsourcing Measurement

To measure the topology of the YouTube CDN from an end users point of view who is connected by an ISP network we used the crowdsourcing platform Microworker [66]. The workers were asked to access a web page with an embedded Java application, which automatically conducts client side measurements. These include, among others, the extraction of the default and fallback server URLs from three predetermined YouTube video pages. The extracted URLs were resolved to the physical IP address of the video servers locally on the clients. The IP addresses of video servers and of the workers client were sent to a server which collected all measurements and stored them in a database.

In a first measurement run, in December 2011, 60 different users of Microworkers participated in the measurements. Previous evaluation have shown, that the majority of the platform users is located in Asia [68], and accordingly most of the participants of there first campaign were from Bangladesh. In order to obtain wide measurement coverage the number of Asian workers participating in a second measurement campaign, conducted in March 2012, was restricted. In total, 247 workers from 32 different countries, finished the measurements successfully identifying 1592 unique physical YouTube server IP addresses.

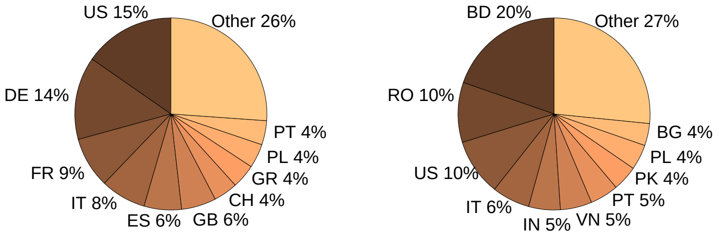
2.3.2 Measurement Results and Their Implications

In this section we show the results of the distributed measurement of the global CDN. The obtained results show the distribution of clients and servers over different countries. Furthermore, the mapping on autonomous systems gives insights to the coverage of the Internet.

Distribution of Vantage Points on Countries

To investigate the coverage of measurement points we study the distribution of the PlanetLab nodes and Crowdsourcing workers. Figure 2.4a shows the distribution of PlanetLab nodes on countries over the world. The pie chart is denoted with the country codes and the percentage of PlanetLab nodes in the respective country. Most of the 220 clients are located in the US with 15% of all clients. However, more than 50% of the clients are located in West-Europe. Only few clients are located in different parts of the world. The tailored distribution towards Western countries is caused by the fact, that the majority of the PlanetLab nodes are located in the US or in western Europe.

Figure 2.4b shows the geo-location of workers on the crowdsourcing platform.



(a) PlanetLab

(b) Crowdsourcing

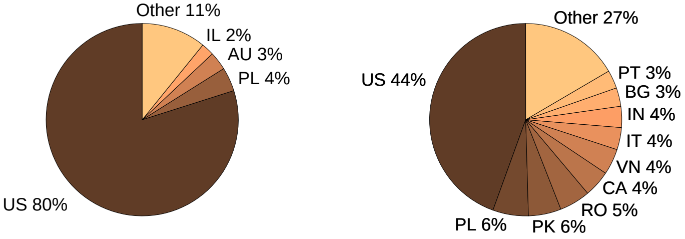
Figure 2.4: Distribution of measurement points on countries in a) PlanetLab and b) Crowdsourcing platform.

In contrast to PlanetLab, most of the 247 measurement points are located in Asia-Pacific and East-Europe. The majority of the participating workers 20% are from Bangladesh followed by Romania and the US with 10%. This bias is caused by the overall worker distribution on the platform [68]. However, this can be influences to a certain extend by limiting the access to the tasks to certain geographical regions.

Distribution of Identified YouTube Servers on Countries

To investigate the expansion of the YouTube CDN we study the distribution of YouTube servers over the world. Figure 2.5a shows the location of the servers identified by the PlanetLab nodes. The requests are mainly directed to servers in the US. Only 20% of the requests were directed to servers not located in the US.

The servers identified by the crowdsourcing measurement are shown in Figure 2.5b. The amount of requests being directed to servers located in the US is still high. 44% of clients were directed to the US. However, in this case the amount of requests resolved to servers outside the US is higher. In contrast to



(a) PlanetLab

(b) Crowdsourcing

Figure 2.5: Distribution of physical YouTube servers on countries accessed from a) PlanetLab nodes and b) workers of a crowdsourcing platform.

the PlanetLab measurement many requests are served locally in the countries of clients. Furthermore, the decrease of 80% to 44% of request being directed to the US shows a huge difference.

Hence, network probes being overrepresented in the US and Europe leads to a limited view of the content delivery network and the Internet. This shows the impact of different locations of measurement points on the view of the CDN. It also demands a careful choice of vantage points for a proper design of experiments in distributed network measurements. Although both sets of measurement points are globally distributed the fraction of the CDN which is discovered by the probes has very different characteristics.

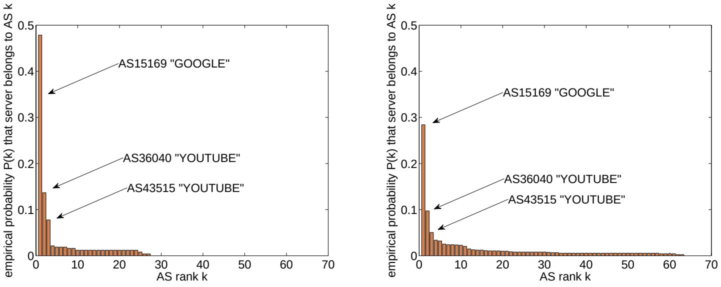
The amount of servers which is located in the US almost doubles for the PlanetLab measurement. While 44% of the requests are resolved to US servers in the Crowdsourcing measurement, nearly all requests of PlanetLab nodes are served by YouTube servers located in the US. Although less than 15% of clients are in US, requests are frequently directed to servers in the US. That means that there is still potential to further distribute the content in the CDN.

Coverage of Autonomous Systems with YouTube Servers

To identify the distribution of clients on ISPs and to investigate the expansion of CDNs on autonomous systems we map the measurement points to the corresponding autonomous systems.

Figure 2.6a shows the autonomous systems of YouTube servers accessed by PlanetLab nodes. The autonomous systems were ranked by the number of YouTube servers located in the AS. The empirical probability $P(k)$ that a server belongs to AS with rank k is depicted against the AS rank. The number of autonomous systems hosting YouTube servers that are accessed by PlanetLab nodes is limited to less than 30. The top three ranked ASs are AS15169, AS36040 and AS43515. AS15169 is the Google autonomous system which includes the Google backbone. The Google backbone is a global network that reaches to worldwide points of presence to offer peering agreements at peering points. AS36040 is the YouTube network connecting the main datacenter in Mountain-View which is also managed by Google. AS43515 belongs to the YouTube site in Europe which is administrated in Ireland. Hence, two thirds of the servers are located in an autonomous systems which is managed by Google. Only few requests are served from datacenters not being located in a Google AS. The reason that request from PlanetLab are most frequently served by ASs owned by Google might be a good interconnection of the NRENs to the Google ASs.

Figure 2.6b depicts the autonomous systems where requests to YouTube videos from the crowdsourcing workers were directed. The empirical probability that a server belongs to an AS has been plotted dependent on the AS rank. The YouTube servers identified by the crowdsourcing probes are located in more than 60 autonomous systems. Hence, the YouTube CDN is expanded on a higher range of ASs from the crowdsourcing perspective compared to PlanetLab. Again the three autonomous systems serving most requests are the ASs managed by Google, respectively YouTube. But the total number of requests served by a Google managed AS is only 41%. Hence, in contrary to the PlanetLab measurement, requests are served most frequently from ASs not owned by Google. Here, caches at local ISPs managed by YouTube could be used to bring



(a) PlanetLab

(b) Crowdsourcing

Figure 2.6: Distribution of YouTube servers on autonomous systems from a) PlanetLab and b) Crowdsourcing perspective.

the content close to users without providing own infrastructure. This would also explain the large number of identified ASs providing a YouTube server. The results show that the PlanetLab platform is not capable to measure the structure of a global CDN, since large parts of the CDN are not accessed by clients in NRENs.

2.4 Traffic Characterization of Peer-to-Peer Networks

To model the BitTorrent traffic flow across ISPs in the Internet, we use measurements of live BitTorrent swarms and the actual AS topology of the Internet provided by Caida.org. Thus, we estimate BitTorrent traffic characteristics and the emerging transit costs. In addition, we define peer selection strategies that decide which peer in a swarm is connected to which other peer and we estimate the transit costs for these strategies.

The applied methodology to characterize BitTorrent traffic and to estimate transit costs is described in Section 2.4.1. In Section 2.4.2 we describe the numerical examples of this study and their importance for ISPs.

2.4.1 Method for Modeling BitTorrent Traffic Flow and Revenue of ISPs

In this section we describe the methodology to estimate transit costs of ASs. First, we show where we obtain the AS affiliation of peers. Second, we explain how AS paths are inferred from AS relations and how to classify the ASs. Further on we describe different BitTorrent peer selection strategies determining the connection among peers in the Internet. Finally we introduce our transit cost model.

AS Affiliation of Peers

In order to know where peers are located and where BitTorrent swarms generate costs for ISPs, we need to know how the swarms are distributed over the Internet and in which ASs the peers are located. For that purpose, we use the dataset of BitTorrent movie torrents “Mov.” provided by the authors of [39]. A snapshot of all available movie torrents on Mininova.org was taken. The swarm sizes and peer distributions were recorded by distributed measurements. The data set consists of files with AS number and number of peers pairs for each BitTorrent swarm. Hence, they provide information for each swarm on how many peers are located in which AS. The measurement took part in April 2009 and recorded 126 050 swarms. Peers of 8 492 ASs are present in the swarms.

AS Relations, Paths and Classification

To be able to estimate the transit costs produced by peers exchanging data in BitTorrent swarms, we need to know the AS paths that connect the peers. Datasets with complete AS paths would be very large and are not available to our knowledge. Hence, we infer AS paths from AS relationships. We use the AS relationship dataset from Caida.org [69]. The dataset contains AS links annotated with

Table 2.2: Classification of autonomous systems.

Type	**Classification**	**#ASs**
Tier–1	AS has no providers	11
Large ISP	AS customer tree ≥ 50	337
Small ISP	AS customer tree < 50 and ≥ 5	1770
Stub	AS customer tree < 5	36289

AS relations. Each file contains a full AS graph derived from RouteViews BGP table snapshots. For our estimations we use the dataset from January 2011. The dataset consists of transit and peering relations.

We implement the algorithm described in [70] in Java to infer the AS paths between any two peers based on the AS relationship dataset. The authors developed a breadth first search algorithm which infers shortest paths conforming to the AS path constraints. The algorithm has runtime $O(N \cdot M)$ for finding all pair valid shortest AS paths of the graph, where N is the number of AS relations and M is the number of ASs. The algorithm's input parameter is the source AS α. For every destination AS β the algorithm returns a set of paths $\mathbb{P}(\alpha, \beta)$, which connect α and β.

Further on, we want to obtain results dependent on the AS size and type of business. Therefore we classify the ASs into *stub*, *small ISP*, *large ISP* and *tier–1*. For that purpose, we use a dataset from [71], which provides information about the number of customers and providers for each AS number. This dataset is from November 2011 and is used to classify the ASs according to the size of their customer tree. Table 2.2 lists the different AS types and their classification. Tier−1 ASs are the largest ASs building the core of the Internet. Tier−1 ASs do not have providers. In their dataset 11 tier−1 ASs are identified. If an AS has a customer tree that contains at least 50 nodes, it is classified as a large ISP. An AS is classified as small ISP if its customer tree has less than 50 but at least 5 nodes. Most ASs are stub ASs, which have a customer tree that is smaller than 5.

BitTorrent Neighbor Set Creation

The BitTorrent neighbor set of a peer defines the data exchange with other peers in BitTorrent swarms. Neighbors are the peers in the swarm which are connected to a peer. It has to be noted that the measurements in [39] do not reveal the real composition of the neighbor sets of the swarms. Further on, neighbor sets are randomly generated and differ for every peer, which makes them hard to capture. Hence, we estimate the composition of the neighbor sets in three simple ways, *random*, *locality* and *selfish-ISP*. The number of peers in the swarm is the swarm size S. The number of neighbors is denoted as N with $N \leq S-1$. In the standard BitTorrent implementation a client can connect to up to 40 peers, so we set the maximum size of the neighbor set to $N_{max} = 40$. Hence, the neighbor set for each peer in a swarm with size S has size

$$N = min(N_{max}, S-1)\,. \tag{2.1}$$

We add peers to the neighbor set until it contains N neighbors according to the following algorithms.

random In the random selection strategy we add random N peers of the swarm to the neighbor set. In the standard BitTorrent algorithm the selection of neighbors is also random. Hence, with this selection strategy we try to estimate the traffic and costs produced by the standard BitTorrent algorithm.

locality In the locality algorithm we sort the AS paths connecting two peers by the number of AS hops. Then we add the peers according to the sorted set of increasing AS paths until N peers are in the neighbor set. Note, that first the peers located in the same AS, i.e., zero AS hops, are added. This selection algorithm is used to optimize the swarm by minimizing AS hops between peers and thereby potentially reducing latencies. Hence, the motivation for this algorithm is to optimize the swarm from the overlay's point of view. In practice, such a selection could be realized e.g. with an *iTracker* [50], a database which maps

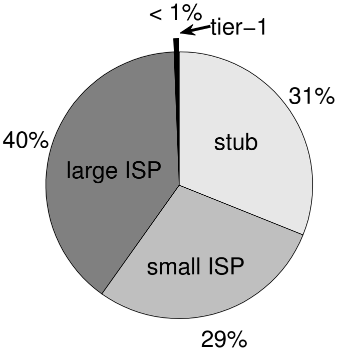


Figure 2.7: Ratio of peers per ISP class aggregated over all swarms in the measurement set.

IP-addresses to autonomous system numbers, or other ALTO mechanisms.

selfish-ISP The selfish-ISP selection algorithm tries to select as many peers from customer ASs as possible. Until the neighbor set contains N peers it first adds peers from paths starting with provider-to-customer links, then peers of the same AS, then paths starting with peering links and finally customer-to-provider links. This selection algorithm is used to maximize the revenue of ISPs. This is achieved by the selfish-ISP strategy by selecting preferentially peers that are connected by customers and avoiding peers at providers. In practice an ISP must be able to control the neighbor set. Hence, ALTO mechanisms for selfish-ISP selection must be controllable by the ISP. One approach is that the ISP provides an information service to guide the peer selection, such as an oracle [48] or an information service [72].

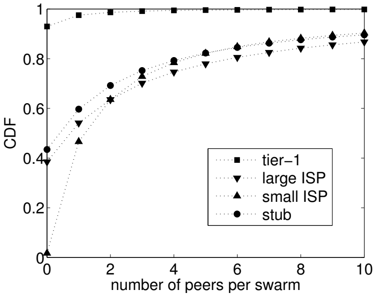


Figure 2.8: CDF of the number of peers per swarm depending on the different ISP classes.

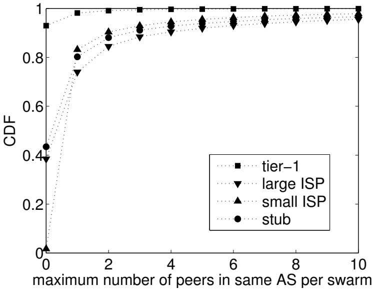


Figure 2.9: CDF of the maximum number of peers in same AS per swarm depending on the tier.

Cost Model

To be able to estimate the costs for ASs arising from transit services, we need to know how much traffic is generated and how much providers charge customers for forwarding the traffic. We consider a snapshot and assume instantaneous traffic rates, i.e., the file-size of the download can be neglected. For simplicity we make assumptions on how much traffic is generated in each swarm, depending on the the number and location of peers.

Assumption 1. *The traffic generated by a peer is equally shared among its neighbors.*

Assumption 2. *All peers generate traffic at the same rate.*

Assumption 3. *The traffic between ASs is equally shared among the paths that connect them.*

In practice, traffic rates are allocated by BitTorrent's choke algorithm, which takes into account the upload and download speed of the other peers. Further on, traffic is generally not shared among different AS paths. But, since we consider the aggregated traffic of a large number of swarms, we argue that these assumptions are reasonable and the results do not change significantly.

Traffic Amount We use the above assumptions to estimate the traffic generated by the BitTorrent swarms. Assumption 1 implies, that the traffic sent by a given peer p_1 is equally distributed among its N neighbors. Hence, the traffic p_1 sends to a neighbor p_2 is

$$T(p_1, p_2) = \frac{1}{N}\,. \tag{2.2}$$

Assumption 2 implies that the traffic originating in a given AS α is proportional to the number of peers located in this AS. Let $\mathcal{S}$ be the set of all swarms, then the traffic of all swarms that is sent from AS α to AS β can be calculated by

$$T(\alpha, \beta) = \sum_{s \in \mathcal{S}} \sum_{\substack{\alpha \in s, \\ p_1 \in \alpha}} \sum_{\substack{\beta \in s, \\ p_2 \in \beta}} T(p_1, p_2)\,. \tag{2.3}$$

The set of AS paths connecting AS α with β obtained by the AS inference algorithm is given by $\mathbb{P}(\alpha, \beta)$. Assumption 3 implies that the traffic between α and β and later the costs are shared equally among the paths in $\mathbb{P}(\alpha, \beta)$. Hence, we can calculate the traffic on a path $P \in \mathbb{P}(\alpha, \beta)$.

$$T(P) = \frac{1}{|\mathbb{P}(\alpha, \beta)|} \cdot T(\alpha, \beta), \quad P \in \mathbb{P}(\alpha, \beta)\,. \tag{2.4}$$

Next we can calculate the link load $L(\alpha, \beta)$ on the link between two directly connected ASs α and β. We use $\alpha \leftrightarrow \beta \in P$ as notation for a direct link between α and β on the path P. The link load is the sum of the load on all paths sharing the link $\alpha \leftrightarrow \beta$.

$$L(\alpha, \beta) = \sum_{P | \alpha \leftrightarrow \beta \in P} T(P)\,. \tag{2.5}$$

As we consider each AS in a swarm as source AS, the outgoing AS traffic equals the incoming AS traffic. Therefore, we only consider the outgoing AS traffic as inter-AS traffic. The in- and outgoing traffic for AS α is the sum of all loads on links connecting α.

$$in(\alpha) = out(\alpha) = \sum_{\beta | \exists P, \alpha \leftrightarrow \beta \in P} L(\alpha, \beta)\,. \tag{2.6}$$

In the following we estimate the transit costs. The transit costs are weighted by the link loads defined in this section.

Transit Costs

The business relationships between ISPs define the exact transit costs, but they are part of the private contracts between the ISPs. Hence, we develop a simple model for the arising transit costs. It is common that peering ASs exchange their traffic and the traffic of their customers without charging. Hence, we assume no costs for peering links. The amount a customer pays a provider for transit for a specific volume of traffic is unclear, so we set it to one cost unit, i.e., 1. That is not the case in practice, but as we have a large number of ASs and swarms, we get a qualitatively good estimation.

The *costs* of an AS α are increased, if it acts as customer of an AS β. The costs are increased by one unit weighted by the amount of traffic on the link connecting α and β, i.e., $L(\alpha, \beta)$ from Equation 2.5. Let $\mathcal{P}(\alpha)$ be the set of providers of α, and let $\mathcal{C}(\alpha)$ be the set of customers of α. Then we can calculate the costs of AS α emerging in all swarms as follows.

$$costs(\alpha) = \sum_{\beta \in \mathcal{P}(\alpha)} L(\alpha, \beta)\,. \tag{2.7}$$

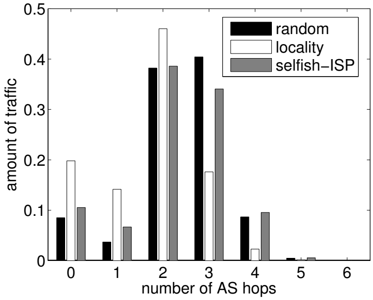


Figure 2.10: Distribution of AS path lengths weighted by the amount of traffic.

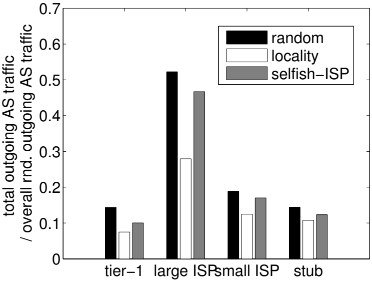


Figure 2.11: Total outgoing AS traffic for different peer selection strategies.

In the same way we can calculate the *revenues* for all AS links and swarms, where α acts as provider.

$$revenues(\alpha) = \sum_{\beta \in \mathcal{C}(\alpha)} L(\alpha, \beta)\,. \tag{2.8}$$

The *balance* is the difference between revenues and costs.

$$balance(\alpha) = revenues(\alpha) - costs(\alpha)\,. \tag{2.9}$$

2.4.2 Numerical Results and Their Implications

In this section we present the numerical results obtained by applying our methodology to the measurement data and describe their importance for ISPs. First we show how the peers are distributed over the different ASs. Then we characterize the traffic emerged by BitTorrent swarms and investigate the impact of locality and selfish-ISP peer selection algorithm. Finally, we estimate the transit costs arising by the BitTorrent swarms and investigate the potential of

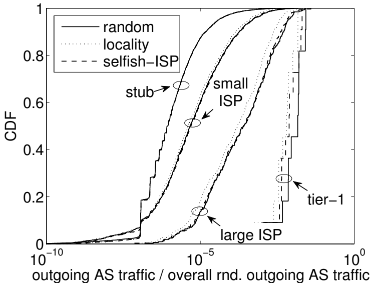


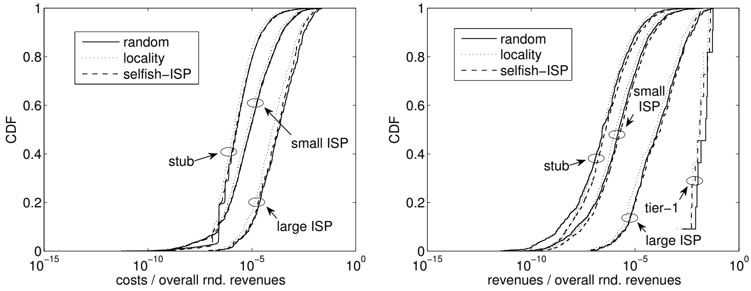
Figure 2.12: CDF of the outgoing AS traffic grouped by AS size.

ISPs to maximize their balance by the peer selection algorithms.

Distribution of Peers in the Internet Hierarchy

In the following we describe how peers of BitTorrent swarms are distributed over the Internet. Figure 2.7 shows the distribution of peers over the different tiers. The peers of all torrents are considered. Most of the peers are in large ISP ASs, where 40 % of all peers are located. In small ISP and stub ASs a similar amount of 29 % and 31 % of the peers is located, respectively. Only very few peers are located in tier−1 ASs, which is less than 1 % of all peers in all swarms. Hence, the access to peers by tier−1 ASs is negligible. That means that tier−1 ASs barely have an impact on ALTO mechanisms that control only the peers of the own AS.

Figure 2.8 shows the cumulative distribution function (CDF) of peers per swarm. We summed up the number of peers being in the same tier for each swarm and calculated the CDF. The probability that at least one peer in a small ISP is existing in a swarm is highest. Only about 2 % of the swarms do not contain any small ISP peer. About 57 % of the swarms contain stub AS peers and more than 60 % contain peers from large ISPs. The probability to find more than one peer of non tier−1 ASs is about 45 % for small ISPs and a bit higher for large



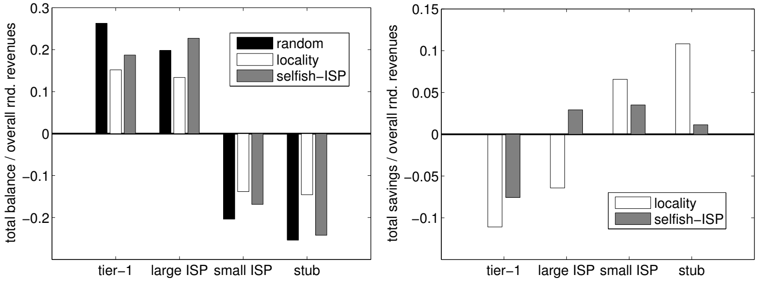
(a) Costs

(b) Revenues

Figure 2.13: Cumulative probability of transit costs (left) and revenues (right) normalized by the overall revenue for random peer selection grouped by AS size.

ISPs and stub ASs. There are less than 10 % of the swarms which contain a peer from tier–1. Finding more than one peer of tier–1 ASs in one swarm is very unlikely. Few swarms have a very large number of peers, with the maximum of 9 467 peers of one distributed over all large ISPs.

Peers can exchange data locally in the same AS as soon as at least two of them are in it. This cannot be derived from Figure 2.8, because peers can be located in the same tier, but not in the same AS. In Figure 2.9 we calculated the cumulative probability of the maximum number of peers in a swarm which are located in the same AS. As soon as the maximum number of peers in one AS is at least 2, data can be exchanged by the peers locally. Figure 2.9 shows that the probability to exchange traffic locally is low and that large ISPs have the greatest potential. In about 15 % of the large ISPs, peers find neighbors being located in the same AS. For small ISP and stub ASs the chance to find peers of the same swarm in the same AS is about 10 % and 12 % respectively. Hence, considering all ASs the potential for local neighbor selection is relatively small, intra-AS traffic is only generated in 15 % and less of non tier–1 ASs. But there are a few swarms with



(a) Balance

(b) Savings

Figure 2.14: Total balance of transit costs (left) and total savings over random selection (right) normalized by the overall revenue for random peer selection.

many peers generating a lot of traffic which have a very high potential for traffic optimization. The AS with the most peers in one swarm is a large ISP containing 3 372 peers of one swarm. The dataset contains 42 swarms with more than 1 000 peers in a single AS. In tier−1 ASs there is barely no chance to connect to a local neighbor.

Traffic Characteristics for P2P Guidance Strategies

In this subsection we characterize the traffic produced by BitTorrent swarms. Further on we investigate the potential of ALTO techniques to optimize the swarm in terms of load on the network and AS path length. First we look at the traffic characteristics of the standard BitTorrent algorithm and in the following we compare the different selection strategies. The number of AS hops is the number of ASs on the AS path connecting two peers without regarding the source AS. The number of hops are weighted by $L(\alpha, \beta)$, i.e. the amount of traffic and the number of concurring AS paths, see Equation 2.5. Figure 2.10 shows the amount of traffic on AS paths with length in AS hops for the different selection strategies. The median is about 2 AS hops if peers are selected randomly.

Most traffic is on paths with two or three AS hops without selection strategy. Paths are up to 10 AS hops in the investigated swarms.

If we use the local selection strategy, the probability for shorter AS paths is higher, compared to random and selfish selection. If local peer selection is used, about 20 % of the traffic can be exchanged in the same AS, i.e. with no AS hop, which is twice as much as for the other strategies. Random and selfish selection have a median of two AS hops, whereas paths have two or less AS hops in about 80 % with local selection strategy. Selfish selection has no considerable potential to reduce the AS path length.

Figure 2.11 shows the amount of inter-AS traffic produced by BitTorrent swarms. We estimate the outgoing traffic of each AS with $out(\alpha)$ in Equation 2.6, i.e. the load produced by all peer-to-peer connections on the links connecting α normalized by the number of neighbors and the number of paths sharing the links. The outgoing traffic of each BitTorrent swarm and each AS is calculated and summed up for the different AS types. For each AS type Figure 2.11 depicts the sum of outgoing traffic normalized by the overall total outgoing traffic produced by random selection of all AS types. The peer selection strategy is coded in the different levels of grey and later line styles. Independent of the selection strategy, most of the traffic is at large ISPs. Less than half of large ISP traffic is at small ISPs. The traffic going out of all the stub ASs is in total a similar amount as the traffic going out of the 11 tier−1 ASs. Hence, most traffic is going out of tier−1 ASs on a per AS basis.

We use the outgoing traffic as a measure for the load on the network. Figure 2.11 depicts the outgoing traffic for the different selection strategies dependent on the AS type. Locality selection reduces the amount of emerging inter-AS traffic in every AS type. Especially large ISPs have a high potential to take load of inter-AS links by selecting local peers. Selfish peer selection reduces the traffic going out of tier−1 ASs, probably because less customers use them as transit providers and route their traffic to customers or keep it local. Apart from that selfish selection does not reduce the load on the network significantly.

Figure 2.12 shows the cumulative distribution function of the outgoing AS traffic grouped by the AS type. The outgoing traffic is normalized by the overall outgoing AS traffic of the random peer selection strategy. AS mention before tier−1 ASs have most outgoing traffic on a per AS basis. Further on we observe that the outgoing traffic decreASs with size of the AS. Also noticeable is that with the locality peer selection algorithm we get less outgoing traffic, especially for large ISPs. The difference is not very big for a single AS, but the large number of ASs makes a big difference in the total outgoing AS traffic.

Transit Costs

Now we estimate the transit costs emerged by BitTorrent traffic for the different ISPs and show the potential to save costs and maximize revenues of the peer selection algorithms. We use the overall revenues for random selection, i.e., the sum of total revenues of all AS types, to normalize the values derived in this section. As the overall total balance is zero, the overall total revenues equal the overall total costs. As described in Section 2.4.1 every customer/provider AS α on an AS path connecting peers is charged by $\pm L(\alpha, \beta)$.

Figure 2.13a shows the cumulative distribution function of transit costs, as calculated in Equation 2.7, for the ASs grouped by AS types. Hence, the amount ASs pay providers for transit services. The costs are normalized by the overall revenues of random selection. tier−1 ASs do not have providers and therefore no transit costs. Local peer selection reduces the transit costs, regarding the overall distribution of costs, for all non tier−1 AS types. Costs of large ASs, i.e., ASs that have many customers and forward a lot of traffic, tend be higher.

Figure 2.13b shows the cumulative probability of revenues, see Equation 2.8, of the ASs grouped by AS type. Tier−1 ASs achieve highest revenues. They have the largest customer tree which pay for transit services. ASs with a smaller customer tree get less revenues. The difference between the selection strategies is small for every single AS, but the large number of ASs makes a big difference in the total revenues and further total balance, as we explain in the next paragraph. However, we observe that stub ASs, small and large ISPs tend to have lower revenues using locality selection compared to random selection. In contrast revenues increase with higher probability for selfish-ISP selection in large intervals, in particular from 10^{-8} to 10^{-4} for stub and small ISPs. This was the aim of the selfish-ISP selection strategy. Tier−1 ISPs are loosing revenues if selection strategies are used. Hence, peer-to-peer guidance and selfish-ISP selection are not beneficial for tier−1.

The total balance over all measured BitTorrent swarms is calculated by subtracting costs from revenues of each AS. Figure 2.14a depicts the total balance depending on the AS size. The total balance is normalized by the overall revenues of random selection. The balance is calculated for the standard BitTorrent peer selection, the locality-aware and selfish strategy. For all three strategies, tier−1 and large ISPs have a positive balance and small ISP and stub ASs have a negative balance. This corresponds to the expectation, because tier−1 and large ISPs have many customers whereas small ISPs and stub ASs have many providers. Hence, small ASs have to pay for the transit provided by large ASs.

To highlight the effect of the peer selection strategies on the balance of the ASs, we investigate the savings over random selection. Comparing the local strategy with the standard strategy, we notice that small ASs save costs by selecting local neighbors, resulting in less revenues by the large ASs. Figure 2.14b shows the savings over random selection achieved by using locality and selfish-ISP selection. The savings are calculated by subtracting the total balance with selection strategy from the total balance of the random selection strategy. The savings are normalized by the overall revenues of random selection. tier−1 ASs loose most revenue when local selection is used, which is 10 % of the overall total revenue. The traffic is kept locally and less traffic is forwarded by tier−1 ASs to reach remote destinations. Hence, the transit services of tier−1 ASs are avoided which results in less revenues. Large ISPs also gain less when local peer selection is used. Small ISPs and stub ASs gain from local peer selection because they save costs for transit services by avoiding long AS paths. 10 % of the overall total costs are saved by stub ASs, hence they have the highest potential to profit from selecting peers by locality.

The only way to increase the prospect on higher profit for large ISPs is using the selfish strategy. But also small ISPs have a high potential to maximize their revenues being selfish. Thus, large and small ISPs are in a win-win situation, because they can connect to their plenty customers and do not have to pay for transit services by avoiding connections to providers. This is where tier−1 ASs loose, because less of the ISPs use them as provider in the selfish strategy. Thus, a tier−1 AS cannot be more selfish than in the random selection strategy. Having only few or no customers, stub ASs have poor capabilities to be selfish but avoiding providers also gives them a small advantage over random selection.

2.5 Lessons Learned

In this chapter we characterized content delivery networks on autonomous systems level. For that purpose we used measurements conducted on the distributed platform PlanetLab and a crowdsourcing platform. To assess the potential of peer assisted content delivery approaches, we determined the number of active IP-addresses from the Internet Census dataset.

First, we have investigated where in the Internet BitTorrent traffic is located and which ISPs benefit from its optimization. To this end, we used measurements of live BitTorrent swarms to derive the location of BitTorrent peers and data provided by Caida.org in order to calculate the actual AS path between any two peers. Our results show that the traffic optimization potential depends heavily on the type of ISP. Different ISPs will pursue different strategies to increase revenues. Our results confirm that selecting peers based on their locality has a high potential to shorten AS paths between peers and to optimize the overlay network. In the observed BitTorrent swarms twice as much traffic can be kept intra-AS using locality peer selection. Thus, the inter-AS traffic is almost reduced by 50 % in tier−1 and in large ISPs.

Second, we proposed the usage of crowdsourcing platforms for distributed network measurements to increase the coverage of vantage points. We evaluated the capability to discover global networks by comparing the coverage of video server detected using a crowdsourcing platform as opposed to using the PlanetLab platform. To this end, we used exemplary measurements of the global video CDN YouTube, conducted in both the PlanetLab platform as well as the crowdsourcing platform Microworkers. Our results show that the vantage points of the concurring measurement platforms have very different characteristics. We could show that the distribution of vantage points has high impact on the capability of measuring a global content distribution network. The capability of PlanetLab to measure a global CDNs is rather low, since 80% of requests are directed to the United States. Our results confirm that the coverage of vantage points is increased by crowdsourcing. Using the crowdsourcing platform we obtain a diverse set of vantage points that reveals more than twice as many autonomous systems deploying video servers than the widely used PlanetLab platform.

Finally, we analyzed the Internet Census Dataset to derive the distribution of IP-addresses on autonomous systems. To this end, we used a mapping of IP-addresses to autonomous system numbers. we find that the distribution of IP-addresses is highly heterogeneous showing that 30% of the active IPs belong to the 10 largest autonomous systems. This means that the potential of approaches that use resources of home routers highly depend on the ISP network.

Based on the results obtained in this chapter, we develop models that describe the characteristics of CDNs and the number of active subscribers in ISP networks. The models allow us to analyze the performance of traffic management mechanisms in realistic scenarios.

3 Analysis and Optimization of Hierarchical Caching Systems

CDNs do not only carry a lot of traffic among the data-centers but also put huge loads on Internet Service Provider (ISP) networks that provide access to a high number of end-users consuming the content. Reducing the traffic carried by CDNs and the load put on ISP networks has high potential to reduce energy consumption and cost for content delivery. A common approach is to cache frequently requested content in or close to access networks to serve the content with low latency and few hops to save resources on the path. The content centric networking architecture proposes content caches on routers on the network path. Caches have a limited capacity to store content, which means that content items stored on the cache need to be replaced if a newly requested item has to be stored in the cache.

A recent approach [26] proposes to augment spare capacities on customer premise equipment (CPE) such as home routers or nano data-centers (NaDas) to assist content delivery, showing that there is a high potential to save energy, although the capacity of home gateways is small and the uplink is limited. The content is transported in a peer-to-peer manner keeping the traffic within the AS. Requests are first directed to the overlay of home gateways or NaDas, and is only forwarded to a cache of the content delivery network, if the target object is not found in the overlay. In this way a hierarchy of caching systems is formed, in which the requests that cannot be served in one tier, i.e. the miss stream, is forwarded to the next tier in the hierarchy. Another example for a hierarchical caching system with bandwidth constraints are femto caching architectures

[32], where content is cached on femto-basestations with small capacity but with considerable storage space. The potential of these approaches highly depends on the number of caches available and their capacity for content delivery. Our goal is to evaluate the performance of hierarchical content delivery networks using a high number of caches with limited capacity and to assess their potential to reduce inter-domain traffic.

The performance of hierarchical cache networks can be accurately determined by analytic models developed in recent work [73, 74]. The models do not consider constraints that limit the capability of caches to upload content such as the bandwidth of the uplink. To consider the upload bandwidth the system is modeled as loss network consisting of a server for each of the caches. The exact stationary distribution of the loss network is too complex to evaluate. In [75] the system is analyzed under large system asymptotic where simplifications occur. We use a different approach by approximating the arrival rate of requests at the caches. This allows us to effectively assess the loss probability by using a simple form of the Erlang formula for a loss network. Even for traffic with highly heterogeneous request rates the approximation reflects the system performance. To determine the number home gateways available to assist content delivery we rely on the insights gained from the characterization of Internet subscriptions on AS level in Chapter 2. In order to assess the potential of hierarchical caching systems to reduce inter-domain traffic, we use the model for transit traffic described in Chapter 2.

Our contribution is three-fold. First, we provide a versatile simulation framework for evaluating hierarchical caching systems allowing to consider different features including the home router sharing probability, bandwidth constraints and the AS topology. Second, we use the inferred AS paths to calculate the real AS paths and assess the transit cost savings by hierarchical caching systems. Finally, we develop a method to accurately assess the system performance of tiered caching architectures with bandwidth constraints analytically.

The content of this chapter is published in [2, 11, 14, 18]. Section 3.1 gives an overview on related work on the performance evaluation of caching systems and describes traffic models and analytic methods, which are relevant for the evaluation of hierarchical caching systems. We describe the simulation model in Section 3.2 and discuss the results derived to assess the potential to save inter-domain traffic in Section 3.2.2. In Section 3.3 we describe our method to evaluate the performance of hierarchical caching systems with bandwidth constraints and give analytic results. Numerical examples derived by analysis and simulation are given in Section 3.3.3. Finally, Section 3.4 summarizes this chapter and presents the lessons learned.

3.1 Background and System Description

In this section we describe content popularity and traffic models used to evaluate the performance of contents delivery networks. We describe the systems model for hierarchical caching systems. We present a representative set of cache eviction policies and provide an overview of recent efforts in modeling the performance of isolated and interconnected caches.

3.1.1 System Model for Hierarchical Caching Systems

Modern content delivery networks use a tree like structure of content caches to delivery content efficiently. Since the tree like structure of caches spreads on the way to the end user, the content replication scales with the content demand.

To model tree like content delivery networks, we consider a set of caches Γ that is organized in a tiered caching architecture as depicted in Figure 3.1. Each cache in Γ has a certain cache capacity C, which specifies the number of content items it can store.

If complete files are considered as transport unit, file size can have an impact due to bin-backing problems. However, content delivery networks and coding schemes for video streams are segmenting data into small chunks in the kB range. Therefore, we simply assume objects of fixed size corresponding to data chunks. Consider that the methods can also be applied to content with varying

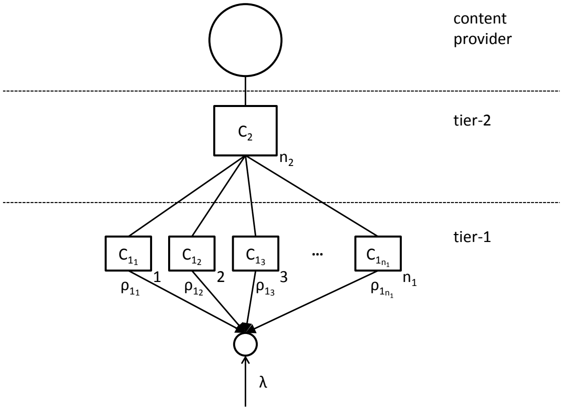


Figure 3.1: System model.

file size, if the sums are weighted accordingly.

Tier-1 caches have capacity $C_{1_i}, i \in \{1, ..., n_1\}$ and tier-2 caches have capacity $C_{2_i}, i \in \{1, ..., n_2\}$. Here we assume, that the cache capacity of all tier-1 caches is equal and use the convention $C_1 = C_{1_i}, \forall i$. This is for example the case if the caches are deployed by a provider on customer premise equipment. If the caches are set up by end-users the cache capacities may vary. Each tier-1 cache i has a specific average upload throughput ρ_{1_i}. Content items are requested from a catalog with size N. Each item $m \in \{1, 2, \dots, N\}$ is requested with rate λ_m. The total arrival rate of requests is $\lambda = \sum_{m=1}^{N} \lambda_m$.

The arrival rate of requests for an item can then also be expressed with the probability p_m that item m is requested:

$$\lambda_m = p_m \lambda \, , \text{where } \sum_{m=1}^{N} p_m = 1 \, . \tag{3.1}$$

Table 3.1: Notation of the paper.

param.	description	default
N	catalogue size	1e6
n_1	number of tier-1 caches	1e4
n_2	number of tier-2 caches	1
C_1	tier-1 cache capacity	8
C_2	tier-2 cache capacity	1e4
ρ_1	tier-1 cache upload bandwidth	0.8Mbps
λ_m	arrival rate of requests for object m	
b_m	bit rate of object m	
d_m	duration of object m	

3.1.2 Content Popularity and Traffic Model

In order to evaluate the performance of content delivery networks, the arrival process of objects needs to be specified. The standard approach to characterize the pattern of object requests arriving at a cache is the Independent Reference Model (IRM) [76]. The IRM makes the following assumptions:

Assumption 1. *Users request objects from a catalogue with fixed size N.*

Assumption 2. *The object popularity does not vary over time, i.e., the probability p_m that an item $m, 1 \leq m \leq N$ is requested, is constant.*

Assumption 3. *The probability p_m that an item is requested, is independent of all past requests, generating an i.i.d request process.*

The IRM ignores temporal correlations in the request process. In practice the request rate of an object increases in a short period of time. This effect is referred to as *temporal locality* and can have a strong positive impact on the efficiency of caching.

To account for temporal locality the request process can be modeled as renewal process [74]. In the renewal traffic model the request process for every content m is described by an independent process with assigned inter-request time distribution. In this case the request process for each content is stationary.

Stationary request processes also mean a static content popularity, which is a very strong assumption. In practice the popularity of contents to be cached can be extremely dynamic over time. In modern content delivery networks, such as YouTube, a high number of new contents is uploaded every single day. While some contents are active only a few days after publishing, e.g. news, other contents, such as songs, remain popular over a long period of time. Hence, the content popularity is highly dynamic and can have a high impact on the efficiency of caching, since the variation of the request rates happens on time scales which are comparable or even smaller than the churn time of caches.

To cover non-stationary macroscopic effects related to dynamic content popularity the Shot Noise Model (SNM) is proposed in [77]. It represents the overall request process as the superposition of a potentially infinite population of independent inhomogeneous Poisson processes, which are referred to as shots. Analytic models for LRU caches under SNM traffic can be developed using the Che approximation.

The analysis of non-LRU caches is very difficult under SNM traffic. In order to analyze the impact of dynamic content popularity on non-LRU cache, [78] propose a traffic model based on a Markov modulated Poisson Process. The Markov modulated Poisson process describes an ON-OFF process for a given content m. The ON and OFF periods are exponentially distributed. During an ON period the request rate for an item λ_m is constant. The model allows simple analysis if the OFF period is set much larger than the cache eviction time. This makes the probability negligible that an item m is still in the cache at the end of its OFF period. An ON period in the ON-OFF traffic model plays exactly the same role as a (rectangular) shot in the SNM traffic model. The authors show in [78] that the cache efficiency under the SNM traffic model can be predicted with high accuracy by adopting a fixed-size content catalogue, and modeling the arrival process of each content by a renewal process with a specific inter-request time distribution.

The popularity of objects is modeled by a Zipf-like law, which is frequently observed for different types of content distributed in the Internet, including video [79, 80]. The Zipf law states that the probability to request the object with rank r, i.e., the r-th most popular object, is proportional to $r^{-\alpha}$. The exponent α has a high impact on the cache performance and ranges between 0.65 and 1 depending on the system and the type of object [81].

3.1.3 Caching Strategies

In the following we give a representative set of caching strategies. The caching strategy decides which object in the cache is evicted if a newly requested item has to be stored in the cache.

- RANDOM: The simplest way to choose an item to make room for a new object is by random.
- LFU: The Least Frequently Used policy evicts the least frequently used item. It stores the most popular items in the cache. LFU performs optimal under IRM traffic.
- LRU: If a newly requested item is not in the cache, it is stored in the cache. The Least Recently Used item is evicted if the cache is full. A well known problem of LRU caches is cache pollution, which occurs if objects are replaced by less frequently requested items or items that are requested only once.
- q-LRU: If a newly requested item is not in the cache, it is stored in the cache with probability q. The Least Recently Used item is evicted if the cache is full. The probability ob being stored in the caches is higher for frequently requested objects, which prevents cache pollution. [74]
- k-LRU: In the k-LRU policy $k-1$ virtual caches storing only object hashes precede the actual cache k. An object is only stored in cache i, if it is found its preceding cache $i-1$. The eviction policy of the caches is LRU. The virtual caches function as filters to prevent cache pollution. [74]
- SG-LRU: Score Gated LRU caching strategies attributes a store to each object. A newly requested object is only stored in the cache if it has a

higher score that the bottom object. The score functions can be based on statistics of past requests approaching the LFU policy if the memory is large. [82]

- LRL: If the capacity of caches is limited to serve requests due to bandwidth or processing constraints, requests for an item are blocked although the item is stored in the cache. The Least Recently Lost strategy evicts the object for which a request was least recently or never blocked. [83]

In a system of interconnected caches, such as the hierarchical caching system described in Section 3.1.1, requests that cannot be served at one cache or in one tier produce a miss and are forwarded to the next tier. In this way the requests traverse a route towards the repository which stores all objects, until they finally hit the target. The following replication strategies for cache networks decide how the object is replicated on the route traversed by the request [74, 84]:

- LCE (leave-copy-everywhere): the object is sent to all caches on the backward path.
- LCD (leave-copy-down): the object is sent only to the cache preceding the one in which the object is found.
- LCP (leave-copy-probabilistically): the object is sent with probability q to each cache on the backward path.

3.1.4 Performance Models for Hierarchical Caching Systems

A vast amount of studies on performance models for hierarchical caching systems have been conducted recently. Table 3.2 gives an overview on the literature on performance evaluation of caching systems. The table shows different categories, considering the caching strategy, the cache topology, the request processes, caching constraints and the methods used for performance evaluation and optimization. In the following we describe the performance models into more detail.

Analytic Performance Models for Caching Systems

To analyze the performance of isolated and interconnected caches, many works leverage the Che approximation [73], which provides a decoupling technique for LRU caches. The requests are generated according to the IRM, which assumes identically and independently distributed requests of a set of objects. It is shown in [81, 87] that the model also applies in more general conditions. The model also provides accurate results for a high number of objects with varying file sizes. In [74] the model is extended, to analyze advanced caching strategies like k-LRU, where objects have to pass a certain number of $k-1$ virtual caches to be stored in the actual cache. The virtual caches replace objects according to LRU and store only meta information. The IRM assumptions are generalized in order to apply to effects of temporal locality in the request process. The model for LRU can be further extended to evaluate the performance of general cache networks [74, 86]. Further caching strategies with limited memory, like W-LFU and Geometrical Fading, are investigated in [82]. The Sliding Window strategy applies the LFU approach to the frequency of requests in a limited time frame. Geometrical Fading scores recent requests with a factor that decreases according to a geometric sequences with the number of intermediate request.

Evaluation of Caching Systems with Dynamic Content Popularity

Only few studies have investigated the effects of dynamic content popularity. The renewal traffic model, which generalizes the IRM, allows capturing temporal locality in the traffic [74]. However, the request process generated by the renewal traffic model is still stationary.

To cover non-stationary macroscopic effects related to dynamic content popularity, a Shot Noise Model (SNM) is proposed in [77]. As already mentioned, for LRU caches analytic models based on the Che approximation can be developed

Table 3.2: Overview on literature on performance evaluation of caching systems.

	LRU policy	q-LRU / k-LRU policy	score based policy (SG-LRU)	loss / bw based policy (LRL)	optimal placement (HWC)	cache hierarchy	general cache network	temporal locality	popularity dynamics	bandwidth constraints	inter-domain traffic	trace driven simulation	simulation (synthetic traffic)	analysis	optimization
Che et al. [73]	●	–	–	–	–	●	–	–	–	–	–	●	–	●	–
Applegate et al. [85]	●	–	–	–	–	●	–	–	–	●	–	–	●	–	●
Rosensweig et al. [86]	●	–	–	–	–	●	●	–	–	–	–	–	–	●	–
Fricker et al. [81, 87]	●	–	–	–	–	●	–	●	–	–	–	●	●	●	–
Hasslinger et al. [82]	●	–	●	–	–	–	–	–	–	–	–	–	●	–	–
Martina et al. [74]	●	●	–	–	–	●	●	●	–	–	–	–	●	●	–
Valancius et al. [26]	●	–	–	–	●	●	–	–	–	●	–	●	–	–	●
Tan et al. [75]	●	–	–	–	●	●	–	–	–	●	–	–	–	●	●
Leconte et al. [83]	–	–	–	●	–	–	–	–	–	●	–	–	●	●	●
Zhou et al. [88]	–	–	–	●	–	–	–	–	–	●	–	–	●	●	●
Traverso et al. [77]	●	–	–	–	–	–	–	●	●	–	–	●	●	●	–
Garetto et al. [78]	●	●	–	–	–	●	●	●	●	–	–	●	●	●	–
Leconte et al. [89]	●	–	–	–	–	●	–	●	●	–	–	●	●	●	–
Lareida et al. [14]	●	–	–	–	–	●	–	–	–	–	●	●	●	–	–
Burger et al. [11]	●	–	–	–	●	●	–	–	–	●	–	–	●	●	–

under SNM traffic. To allow analysis of non-LRU policies the ON-OFF model is proposed in [78] using an on-off-modulated Poisson process. In [89] the analysis under SNM traffic is further extended for caches with small population, such as base-stations, home routers or set-top boxes.

Evaluation of Caching Systems with Limited Capacity

The analytic models described above do not consider the service time of requests, which is limited by the bandwidth of the uplink of the cache. In this monograph we consider a tiered caching architecture, where the upload bandwidth of the caches is highly limited. The paper closest to this works is [26], which proposes the NaDa approach and develops an optimal content placement on NaDas. The performance of the approach is evaluated with traces only. To evaluate the performance analytically considering bandwidth constraints the system can be modeled as loss network consisting of a server for each of the caches. The exact stationary distribution of the loss network is too complex to evaluate due to the high number of feasible content placements. In [75] the system is analyzed under large system asymptotics where the number of NaDas goes to infinity and simplifications occur.

In order to evaluate the system for a finite number of NaDas, we use a different approach by approximating the arrival rate of requests. This allows us to effectively assess the loss probability by using a simple form of the Erlang formula for a loss network.

To optimize the content placement adaptively [83] propose Least-Recently-Lost (LRL) replacement, which tries to optimize the loss rate in the first tier. A different approach is proposed by [88] which allocates bandwidth resources instead of content copies proportionally to the content popularity.

3.2 Simulative Evaluation of Hierarchical Caching Systems

We develop an event-based simulation framework to evaluate the performance of content delivery networks. The results derived from the simulative evaluation are used to validate the analytic models. The simulation framework further allows considering complex system characteristics in the performance evaluation that are not covered by the analytic models, such as the transit costs charged on inter-domain links.

In the following we describe the simulation model and investigate the benefit of overlays networks in hierarchical caching systems. We use the model for transit traffic, c.f. Section 2.4.1, to assess the potential to save costs produced by inter-domain traffic.

3.2.1 Simulation Model

The parameters considered in the content delivery simulation framework are

a) the resource distribution,

b) the caching and content placement strategy,

c) the resource selection strategy,

d) the content demand,

e) the AS-Topology.

Other considered parameters that are not relevant for this monograph, are

a) the social network of users,

b) the video bitrate and chunk-size distribution,

c) and the application and QoE.

In the following we briefly describe each of the parameter sets and provide models.

Resource Distribution

The resource distribution determines how video streaming sources are distributed among autonomous systems. The number and size of autonomous systems is specified. The size of an autonomous system is given by the number of end-users located in it. In literature, the distribution of end-users on ASs is characterized as heterogeneous [39]. We use a geometric distribution as a basic model for number of end-users in the ASs. A more detailed model is developed using the Internet Census dataset, c.f. Section 2.1.3. Video streaming sources can be a) data centers of the content provider, b) edge caches of the content provider, c) caches hosted by the ISP, d) home router / NaDas, or e) end-user devices. For each video streaming source the AS-location and its capacity is specified. The capacity is given by the number of items that can be cached. The size of the item catalogue is also specified in this parameter set.

The cache resources can have bandwidth constraints specified by the mean and the standard deviation of the upload bandwidth. If the upload bandwidth of a cache is limited, the service time of an object is calculated according to the available bandwidth and the object size. The service times of the objects served by a cache are updated if the upload bandwidth changes or if an object request arrives or is completed. Requests are blocked by a cache if the available bandwidth is below a certain threshold, or if the cache is busy serving a request.

Caching and Content Placement Strategy

The caching and content placement strategy determines in which video streaming source which video item is placed and when. The content placement strategy is defined by the caching strategies of the individual caches. In a distributed approach each cache decides based on the information it has, which items to cache. Thus, the availability of items in ASs might for example be increased. If global knowledge of the item demand is assumed, optimized content placement strategies such as hot warm cold can be used. Each caching strategy is further defined by its specific parameters according to Section 3.1.3.

Resource Selection Strategy

The resource selection strategy determines from which cache instance an item is streamed when requested. The simplest resource selection just selects a random resource. In hierarchical content delivery networks, resources in tier-1 are selected first, by default. Other resource selection strategies that try to optimize different metrics were implemented. E.g., local resource selection tries to save inter-domain traffic by prioritizing caches in the order: home router / NaDa in the same AS, ISP managed cache in the same AS, edge cache of content provider, data center of content provider.

Social Network of Users

The social network of users determines the friendship relationships between users. A basic model only defines the number of users in the system. The number of friends of the user can be modeled by a power-law or geometric distribution. A more detailed model specifies the friendship graph which consists of a node for each user and edges between users with friend relationships. Friendship graphs have typical properties, such as a heavy-tailed in and out degree distribution. In literature are different models for generating graphs with these properties. A model used to generate social network graphs with varying size and density is the forest fire model [90]. We further specify the feed size as parameter that represents the news feed of social network platforms. The news feed is updated in sharing events. Videos which are on the news feed of a user are watched with high probability. Categories are defined by specifying the probability that a user is interested in a particular category.

Traffic and Popularity Model

The content demand determines the request rates of the video items. Different demand models are implemented in the simulation, that reach from basic models that only consider the popularity distribution of the items, to detailed models that consider temporal, spatial and social dynamics, c.f. Section 3.1.2.

The arrival process of video requests is specified by the inter-arrival time of video requests. The request rate depends on the time of day and is generally lower at night. The day is divided in short time slots, where the arrival rate does not change significantly, so that the arrival process can be assumed as quasi stationary. In these time slots the arrival process is modeled as Poisson-process. The parameter lambda of the arrival process depends on the popularity of the item and the time of day. The probability of sharing a watched video is given by the sharing probability.

Autonomous System Topology

In order to estimate the amount of inter-domain traffic and transit costs produced, and AS topology with AS paths can be specified. The AS paths connect the caches and data centers providing the content with the users consuming the content. For that purpose the AS paths are inferred from AS relationships as in Section 2.4.1. Assuming that the number of users is proportional to the number of IP addresses in an AS, we use the results of the Internet Census Dataset evaluation in Section 2.1.3, to determine the distribution of users on ASs.

Finally, Simulation parameters are specified that define the random number seed, the simulation time and the parameter study.

Performance metrics

To assess the performance of content delivery networks several metrics are considered:

a) Cache hit rate: The ratio of requests to a cache that find the object in the cache (cache hit).

b) Cache serve rate: The ratio of requests to a cache that find the object in the cache and that are not blocked due to bandwidth constraints.

c) Cache contribution: The share of all requests that is served by a cache.

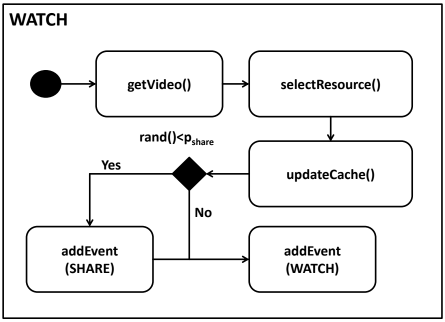


Figure 3.2: Proccess diagram of a WATCH event.

d) Inter-domain traffic: The share of requests by users in an AS that cannot be served by a cache in the same AS.

The content delivery simulation framework is implemented in MatLab. The simulation is event-based including two major events. First, the WATCH event, which is processed when a user watches or consumes a video item. Second, the SHARE event, which simulates a sharing action of a user, where the video is posted on the news feeds in the social network. Figure 3.2 shows the process diagram of a WATCH event. The process of a WATCH event starts by selecting a video according to the specified demand model. A video identifier v_{id} is returned. In the next step a cache or data center is selected according to the resource selection strategy that holds the item with v_{id}. The download of the item from the selected resource is recorded in the statistics. The cache identifier c_{id} is returned and the cached items are updated according to the caching strategy specified in the parameters. The user then decides to share v_{id} with probability p_{share}. In this case a SHARE event is queued. Finally, the next WATCH event is queued according to the traffic and popularity model.

A SHARE event puts a given v_{id}, or a random video according to the user's interest on top of the news feed of the user's friends. The user's friends are determined by the social graph. The simulation is initialized with a WATCH event for each user.

The simulation framework is open source and available on github[1].

3.2.2 Numerical Examples and Impact on Transit Costs

To evaluate the performance of a CDN supported by home routers, two scenarios are simulated. The first scenario simulates requests to a CDN with caches organized in a tree structure and compares isolated caches to cooperating caches to assess the benefit of the overlay. The second scenario adds an AS topology with peering and transit links to evaluate the inter-domain traffic saving potential. As described in Chapter 2 a transit link exists between a customer ISP and its transit provider, if the customer ISP pays the transit provider to forward its traffic destined to parts of the Internet that the customer ISP does not own or cannot reach.

Within tier-1, the caches are placed on shared home routers. These caches are referred to in the following as home routers (HRs). The cache capacity of HRs is specified by C_1 and their caching strategy is LRU. In this study C_1 is set to four (4) content items. We evaluate the performance dependent on the autonomous system size n_{user}, in terms of the number of end-users in the autonomous system. The probability that an end-user enables the HR to shares contents is given by p_{share}. The probability that a user requests certain content items depends on the content's popularity distribution, which is specified by the Zipf exponent α.

Benefits of an Overlay

To evaluate the performance of the overlay, two cases are considered (a) the tree case and (b) the overlay case. In the tree case (a), each user is assigned to one shared HR in its AS. If a user shares its HR, it is assigned only to its HR. A requested item is looked up in the assigned HR initially, i.e. in the tier-3 cache.

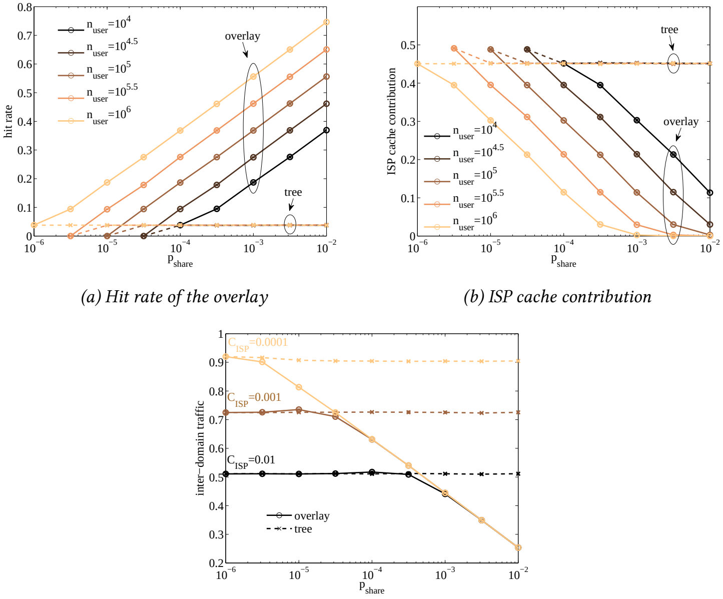
[1] `https://github.com/pettitor/content_delivery`

If the requested item is not found, the request is forwarded to the next tier. The hierarchic caching strategy is leave-copy-everywhere, which means that the video is cached in each cache on the look up path. In the overlay case (b), a requested item is looked up in the HR of the user, if it is not found, it is looked up in shared HRs in the same autonomous system using the overlay. If no tier-3 cache in the AS contains the item it is looked up in tier-2 caches and finally in the data center of the content provider. The hierarchic caching strategy is leave-copy-everywhere, too, with the constraint, that the item is cached in the tier-3 cache only, which was looked up first.

As the goal of this evaluation is to assess the potential of the overlay and to identify success scenarios, the simulation model assumes that the upload rate of caches is unlimited. However, in practice the upload rate limits the number of requests that can be served by a cache, especially for smaller devices like HRs. The evaluation uses a static and global popularity distribution. In practice the item request process is dynamic and dependent on personal and regional preferences. The simulation uses a catalog size of $N = 10^6$. The results obtained show the average of ten simulation runs with 10^6 requests and their respective 95% confidence intervals.

Figure 3.3a shows the hit rate of the overlay dependent on the sharing probability for a constant ISP cache capacity of $C_{\text{ISP}} = 0.01$. In the tree case, where each user is assigned to a HR as a tier-3 cache, the hit rate is independent of the sharing probability. The hit rate is limited by the cache capacity of the HR. If HRs are organized in an overlay, their hit rate increases with the sharing probability, since requested content items are looked up in all HRs belonging to the overlay. This shows that an overlay highly increases the performance of a caching system with a high number of small caches. Hence, the overlay highly benefits providers and end-users. The hit rate increases with the size of AS n_{user}, because a higher total cache capacity is available.

Figure 3.3b shows the ISP cache contribution dependent on the sharing probability for a constant ISP cache capacity of $C_{\text{ISP}} = 0.01$. In a tree structure the sharing probability has no significant impact on the ISP cache contribution. This



(a) Hit rate of the overlay

(b) ISP cache contribution

(c) Inter-domain traffic

Figure 3.3: Overlay and ISP cache contribution and inter-domain traffic dependent on home router sharing probability.

depends on the fact that the hit rate of tier-3 caches is low and independent of the sharing probability. All remaining requests are forwarded to the ISP cache and in case of a hit the ISP cache contributes. If the HRs are organized in an overlay, the ISP cache contribution decreases because more requests can be served from the overlay. In this case the ISP cache also gets less efficient because it is only requested for rare items that are not cached in the overlay. For large ASs with a high number of end-users the ISP cache contribution approaches zero, if at least every thousandth user shares its HR. In this case the ISP cache can be shut down, which saves operating costs and energy. This shows that especially large ASs can benefit from an overlay.

Figure 3.3c shows the inter-domain traffic dependent on the sharing probability for an AS with $n_{\text{user}} = 10^6$ end-users. If no overlay is present the sharing probability has close to no impact on the amount of requests served locally. In this case the inter-domain traffic can only be reduced by increasing the ISP cache capacity. In the overlay case the number of requests served locally increases with the sharing probability, which decreases the inter-domain traffic. Dependent on the ISP cache capacity a higher fraction of shared HRs is necessary to reduce inter-domain traffic.

Inter-Domain Traffic

The overlay is not only used to access content from HRs in the same AS, but also from HRs in neighboring ASs. If the neighboring AS is a peering or customer ISP, no transit costs are incurred. To assess the inter-domain traffic saved, an AS topology is added to the simulation. The AS relationship dataset provided by caida.org[91] of January 2015 is used and it specifies peering and customer-to-provider links of each AS. The data set consists of 46,172 ASs and 177,000 links. To be able to process the simulation the topology is limited to RIPE NCC EU ASs. The remaining subset still consists of 31,256 ASs and 77,382 links. The number of users per AS is determined by evaluating the Internet Census Dataset[33], which provides a scan on active IP addresses in the Internet. Assuming that the number of users in an AS is proportional to the number of active IP addresses and the probability of a user being in an AS is set accordingly. To save costly inter-domain traffic and to mitigate load on ISP caches, the following resource selection policy is applied:

If an item is not found on enabled HRs in the same AS, it is requested from other resources in the order:

a) HRs in peering ISP ASs

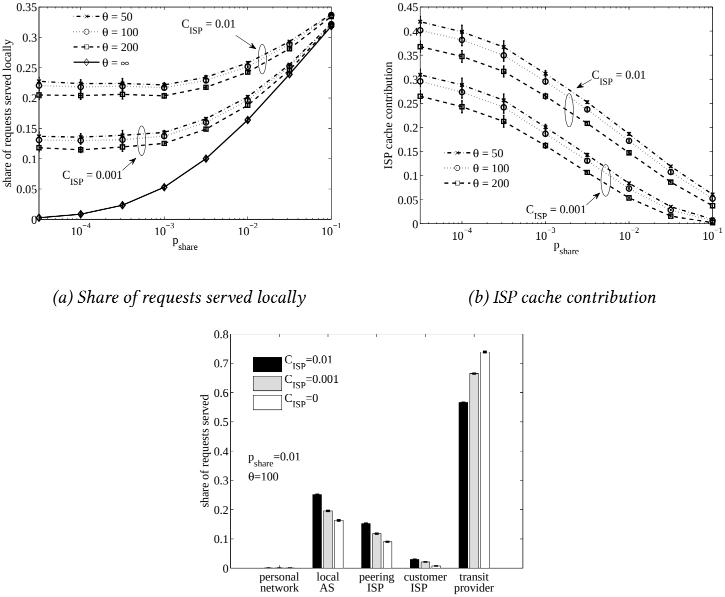
b) HRs in customer ISP ASs

c) ISP cache in local AS

d) ISP cache in peering ISP ASs

e) ISP cache in customer ISP ASs

f) content provider



(a) Share of requests served locally

(b) ISP cache contribution

(c) Share of requests served per domain

Figure 3.4: Share of requests served locally and ISP cache contribution dependent on sharing probability and share of requests served per domain.

A policy designed to prioritize ISP caches to remote HRs did not have a significant impact on traffic savings. The threshold θ specifies the minimum number of users an AS must have to host an ISP cache. If $\theta = \infty$ no AS hosts an ISP cache and content delivery is solely supported by HRs. To investigate the performance of our approach, the impact of the HR sharing probability p_{share} on the inter-domain traffic and on the ISP cache contribution is studied. The share of traffic within the local AS, peering and customer-to-provider links is evaluated. For the generation of content item requests a Zipf popularity distribution with slope $\alpha = 0.99$ was applied.

Figure 3.4a shows the share of requests served locally dependent on the HR sharing probability. More than 20% of requests can be served locally, if the ISP cache can store 1% of the catalog size. With an increasing threshold θ the number of ASs hosting an ISP cache decreases and, thus, the share of requests being served locally. If the number of shared HRs increases, more traffic can be kept locally. This effect is stronger for a lower ISP cache capacity. In case of $\theta = \infty$ where no ISP caches are available, the sharing probability has the strongest impact on inter-domain traffic. For a high sharing probability the ISP cache size has only little impact on the inter-domain traffic.

Figure 3.4b shows the ISP cache contribution dependent on the HR sharing probability. The number of requests an ISP cache can serve increases with its capacity. As for the inter-domain traffic, the sharing probability has a high impact on the ISP cache contribution. For high sharing probabilities the ISP cache contribution approaches zero. This means that ISP caches can be shut down, if a sufficient amount of users would share their HRs. For a lower threshold θ more ISP caches are deployed and the ISP cache contribution increases.

To study the requests served per domain, the HR sharing probability is set to 1% and the threshold θ to 100 users. Figure 3.4c shows the share of requests served per domain. Almost none of these requests can be served by the personal HR. This might depend on the fact that items are requested according to a global popularity distribution. If personal interests are considered in the demand model, higher hit rates and contributions from personal caches are expected. Dependent on the ISP cache capacity, 20 to 25% of requests can be served locally and 15 to 20% from neighboring ASs. Still about 2 out of 3 requests are served by the content provider. This depends on the fact that with Zipf slope of $\alpha = 0.99$

content item requests are highly heterogeneous. In practice, temporal and social dynamics of users' interests will lead to temporal and local correlations in requests, which improve the performance of local and personal caches.

3.3 Analysis of Caching Systems with Bandwidth Constraints

In order to evaluate content delivery networks based on the number of available home routers and their limited capacity, we define a system model for a tiered caching architecture. Tier-1 caches are on leaf nodes such as home routers, caches of the content delivery network are in tier-2 and ultimately tier-3 is the content provider. We use analytic models to calculate the efficiency of the tiered caching architecture.

3.3.1 Analytic Performance Models for Caching Systems

In the following we provide analytic models to evaluate the performance of a tiered caching architecture. We use existing models for systems without bandwidth constraints in order to determine baselines and upper bounds for comparison. We then show our approach to determine the hit rate of hierarchical cache networks with bandwidth constraints.

The Che Approximation

We first consider the Che approximation [73] for the simple case of a single cache with LRU policy. Let C be the capacity of the single cache. $T_C(m)$ is the cache eviction time of object m, i.e., the time needed before C objects, not including m, are requested at the cache. Object m is in the cache if the last request for object m is less than T_C in the past. For Poisson arrivals, the probability that an object m is in the cache equals the probability that the inter-request time for object m is smaller than T_C. Let A_m be a random variable for the inter-arrival time for requests of object m. The probability that A_m is smaller than

T_C is given by the cumulative distribution function, which is exponentially distributed:

$$p_{\text{hit}}(m) = p_{\text{in}}(m) = P(A_m \leq T_C) = 1 - e^{-\lambda_m T_C} \tag{3.2}$$

Due to the memoryless property of the Poisson process the hit probability equals the stationary probability that an item m is in the cache.

We define the indicator function χ_m to determine if item m is in the cache.

$$\chi_m = \begin{cases} 1, & m \text{ in cache} \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

Following [74], we obtain for the cache capacity C:

$$C = \sum_m \chi_m \;, \tag{3.4}$$

and

$$C = \mathbb{E}\left[\sum_m \chi_m\right] = \sum_m \mathbb{E}\left[\chi_m\right] = \sum_m p_{\text{in}}(m) \tag{3.5}$$

after averaging both sides.

T_C is the only unknown in the above equation and can be determined by a fixed point iteration. Thus, the interaction among the contents is summarized by the cache eviction time T_C, which allows decoupling the dynamics of the different contents.

The overall hit probability is calculated by considering the probability p_m of requesting item m.

$$p_{\text{hit}} = \sum_m p_m p_{\text{hit}}(m) \tag{3.6}$$

No bandwidth constraints

We use the Che approximation for the LRU cache hit rate to calculate the baseline given by the cache hit rate of the tier-2 cache without tier-1 cache support $p'_{\text{hit}}(2)$. The characteristic time T_{C_2} depends on the capacity of the tier-2 cache C_2 and is determined by a fixed point approximation.

$$p_{\text{hit}}(2, m) = p_{\text{in}}(2, m) = 1 - e^{-\lambda_m T_{C_2}} \tag{3.7}$$

The overall hit probability in tier-2 is calculated by considering the probability p_m of requesting item m.

$$p'_{\text{hit}}(2) = \sum_m p_m p_{\text{hit}}(2, m) \tag{3.8}$$

To calculate the maximum hit rate for LRU, we assume that tier-1 caches are completely organized with the tier-2 cache, and the capacity of tier-1 caches is added to the tier-2 cache capacity.

$$\hat{p}_{\text{hit}}(m) = \hat{p}_{\text{in}}(m) = 1 - e^{-\lambda_m T_{(C_2 + n_1 \cdot C_1)}} \tag{3.9}$$

It is practically not feasible to control the capacity of all tier-1 caches and coordinate them with the tier-2 cache. To still bundle the capacity of the tier-2 caches, the caches can form an overlay.

If an overlay is used, the requests that cannot be served by the personal tier-1 cache are forwarded to other tier-1 caches in the overlay, before they are forwarded to the tier-2 cache.

We approximate the hit rate of the overlay by calculating the hit rate of a tandem network with two caches according to [74]. The tandem network consists of the tier-2 cache and a cache that has the sum of capacities of tier-1 caches. The miss stream of the consolidated tier-1 cache is forwarded to the tier-2 cache. The replacement strategy in the network is leave-copy-down, that means that an item is only placed in a cache if it is found in a higher tier cache. Thus only frequently requested items are propagated to lower tier-caches, which makes them more efficient. The Che approximation can be applied to tandem networks by determining $p_{\text{in}}(1,m)$ for the tier-1 caches.

$$p_{\text{in}}(1,m) = 1 - e^{\lambda(1,m)T_{n_1 C_1}} \tag{3.10}$$

The hit probability is no longer equal to the probability that an item is in the cache, as the cache miss stream arriving at the tier-2 cache is no longer Markov. According to [74] the hit probability can then be determined by:

$$\begin{aligned} p_{\text{hit}}(1,m) = ((1 - p_{\text{in}}(1,m))p_{\text{hit}}(2,m) + p_{\text{in}}(1,m)) \\ \cdot (1 - e^{\lambda(1,m)T_{n_1 C_1}}) \end{aligned} \tag{3.11}$$

The rate of the miss stream $\lambda(2,m)$ arriving at the tier-2 cache can then be determined as

$$\lambda(2,m) = (1 - p_{\text{in}}(1,m))\lambda(1,m)\,. \tag{3.12}$$

The probability that an item is in the tier-2 cache is approximated assuming exponentially distributed inter request times.

$$p_{\text{hit}}(2,m) = p_{\text{in}}(2,m) = 1 - e^{\lambda(2,m)T_{C_2}} \tag{3.13}$$

The total hit rate in tier-i is then calculated by considering the probability of an item m being requested at tier-i cache $p(i,m)$.

$$p_{\text{hit}}(i) = \sum_m p(i,m)p_{\text{hit}}(i,m), i \in \{1,2\} \tag{3.14}$$

The overall hit rate of the hierarchical caching system is then calculated by

$$\bar{p}_{\text{hit}} = p_{\text{hit}}(1) + (1 - p_{\text{hit}}(1))p_{\text{hit}}(2)\,. \tag{3.15}$$

3.3.2 Analytic Model with Bandwidth Constraints

The above hit rates only apply if tier-1 and tier-2 caches have unlimited bandwidth, which is practically not feasible. Considering the home router scenario, the bandwidth of tier-1 caches is limited depending on the subscription and the availability of DSL. We use the throughput of the tier-1 caches ρ_1 as parameter to specify the upload bandwidth available on home routers.

The offered traffic of item m at tier-1 cache k can be calculated by the quotient of the arrival rate per cache $\frac{\lambda_m}{n_1}$ and the mean service rate, which is determined by the bitrate b_m and the duration d_m and the link throughput ρ_1 in case of video contents.

$$a(m,k) = \frac{\lambda_m \cdot b_m \cdot d_m}{n_1 \cdot \rho_1(k)} \tag{3.16}$$

The total offered traffic of item m is $a(m) = \sum_k a(m,k)$.

The content placement in tier-1 is specified by $X : N \times n_1 \mapsto 0, 1, X(m,k) = 1$, if content m is placed on cache k, else 0.

According to [26] an optimal placement of items in terms of minimum loss rate in the stationary case is achieved by the hot-warm-cold content placement. Hot content with $a(m) \geq n_1$ is placed on each cache. Warm content is placed on $\lfloor a(m) \rfloor$ caches. Cold content is not placed on any of the caches. The constraint $\forall k, \sum_m X(m,k) \leq C_1(k)$ has to be met, such that the cache capacities are not exceeded.

Since not every hit can be served by tier-1 caches because of their limited bandwidth, we consider the loss rate $p_b(1,m)$, which is defined as the share of requests of item m that is not hit in tier-1 or is blocked if none of the tier-1 caches storing the requested item has enough bandwidth left to serve the request.

Let ν_m be the number of tier-1 caches that hold item m.

$$\nu_m = \sum_{k=1}^{n_1} X(m,k) \tag{3.17}$$

An item is not hit, if it is not placed in any tier-1 cache, i.e., if $\nu_m = 0$. If an item is placed in at least one of the tier-1 caches, i.e., $\nu_m > 0$, we approximate the blocking probability by the Erlang formula for a loss system with ν_m servers with mean service rate $\mu_m = \frac{\rho_1}{b_m \cdot d_m}$ and arrival rate $C_1 \lambda_m$:

$$p_b(1, m) = \begin{cases} \frac{\frac{a_m^{\nu_m}}{m!}}{\sum_{k=0}^{\nu_m} \frac{a_m^k}{k!}}, & \nu_m > 0 \\ 1, & \text{otherwise}\,, \end{cases} \tag{3.18}$$

where we approximate the offer of item m with

$$a_m \approx \frac{\lambda_m \cdot C_1}{\mu_m}\,. \tag{3.19}$$

Note, that here we assume that the arrival rate of requests of the C_1 items stored in the cache is equal to the rate of item m. In order to calculate the exact stationary distribution of the blocking probability, the arrival rate of requests has to be conditioned on the feasibility of the content placement, which is too complex to evaluate. Refer to [75] for details.

The blocked requests are forwarded to the tier-2 cache and the arrival rate of requests of item m at the tier-2 cache can be determined as

$$\lambda_m(2) = \lambda_m \cdot p_{\text{b}}(1, m)\,. \tag{3.20}$$

We determine the hit rate of the tier-2 cache $p_{\text{hit}}(2)$ again by using the Che approximation for cache capacity C_2 and arrival rates $\lambda_m(2)$, assuming that the miss stream of the tier-1 caches follows a Poisson processes.

$$p_b(1) = \sum_m p_m p_b(1, m) \tag{3.21}$$

The total rate of requests hit and served by tier-1 and tier-2 caches is then determined by

$$p_{\text{hit}} = (1 - p_b(1)) + p_b(1) \cdot p_{\text{hit}}(2) \tag{3.22}$$

In order to assess the benefit of the tiered architecture compared to a single ISP cache, we define the cache hit rate gain ω as the normalized difference of the total hit rate p_{hit} and the cache hit rate of the tier-2 cache $p'_{\text{hit}}(2)$ without tier-1 cache support.

$$\omega = \frac{p_{\text{hit}} - p'_{\text{hit}}(2)}{p'_{\text{hit}}(2)} \tag{3.23}$$

3.3.3 Numerical Examples

We evaluate the performance of tiered caching systems with bandwidth constraints in parameter studies. We validate the model via simulations, which allows us to verify the accuracy of the analytical model and the validity of our conclusions based on the model for a wide range of system parameters. Before presenting numerical examples, we briefly describe the event-based simulation. The simulation framework used is implemented in Matlab and described in detail in [2]. The results presented show the mean values of 8 runs with 95% confidence intervals, each run simulating 10^5 requests in the stationary phase.

If not stated otherwise, in the remainder of this study, the catalogue size is $N = 1e6$. The number of tier-1 caches is $n_1 = 1e4$, each having a capacity of $C_1 = 8$. There is one tier-2 cache with default capacity $C_2 = 1e4$. The tier-1 cache upload bandwidth is limited to 0.8Mbps, whereas the tier-2 cache has unlimited upload bandwidth.

We first consider a scenario without tier-2 cache. In this case the caching architecture only consists of tier-1 caches. We evaluate the cache hit rate dependent on the number of tier-1 caches n_1. Figure 3.5 shows the results for different upload bandwidth of tier-1 caches ρ_1. The cache hit rate increases with the number of tier-1 caches and their upload bandwidth. The analytic model slightly overestimates the cache hit rate for finite upload bandwidth of tier-1 caches. In the case of infinite upload bandwidth, the results can be compared to

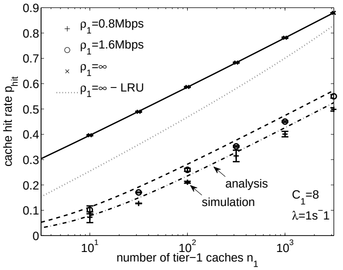


Figure 3.5: Comparison of cache hit rate for optimal placement with LRU policy.

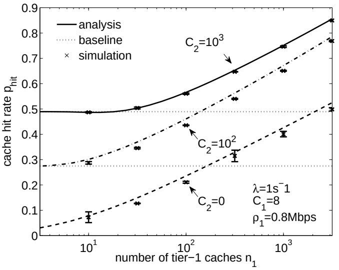


Figure 3.6: Cache hit rate dependent on the number of tier-1 caches for different tier-2 cache capacities.

an LRU cache with capacity n_1C_1. In the optimal placement, the items that are most frequently requested are placed on the caches, which explains the higher hit rate compared to LRU. Hence, there is a high potential to increase the hit rate of the system by using an optimal content placement used. Furthermore, the results show that it is important to consider bandwidth constraints in the analysis of caching systems. If the bandwidth is limited to 1.6Mbps, the cache hit is reduced by about 30% compared to the case of unlimited bandwidth.

To investigate the influence of the tier-2 cache on the cache hit rate, we vary the tier-2 cache capacity C_2. Figure 3.6 depicts the hit rate of the tiered caching architecture dependent on the number of tier-1 caches for different tier-2 cache capacities. As baselines the cache hit rate $p'_{\text{hit}}(2)$ of a single tier-2 cache with capacity C_2 without tier-1 support is depicted. The overall hit rate increases with the tier-2 cache capacity.

The performance of the caching architecture also highly depends on the overall request rate λ. Figure 3.7 shows the cache hit rate dependent on the number of tier-1 caches for varying request rates λ. Due to the limited upload bandwidth of tier-1 caches, more requests are blocked and forwarded to the tier-2

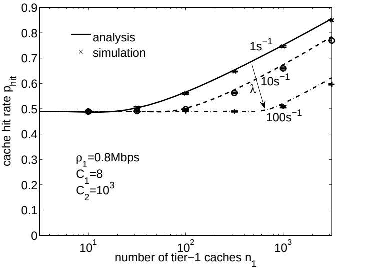


Figure 3.7: Cache hit rate dependent on the number of tier-1 caches for varying request rates λ.

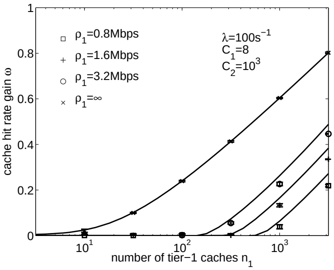


Figure 3.8: Impact of the upload bandwidth ρ_1 on the cache hit rate gain ω.

cache, which reduces the total rate of requests hit and served by tier-1 and tier-2 caches. If the request rate increases, more tier-1 caches are necessary to increase the overall hit rate.

In order to evaluate the performance of the tiered caching architecture for a medium sized ISP, we consider a high request rate $\lambda = 100s^{-1}$ and a high tier-2 cache capacity $C_2 = 10^3$. We study the impact of the upload bandwidth of tier-1 caches on the cache hit rate gain ω. If the upload bandwidth of the caches is low, a high number of tier-1 caches is necessary to improve the performance of the caching architecture. The number of tier-1 caches necessary to gain hit rate decreases with their upload bandwidth.

Hence, if no tier-2 cache or only a small tier-2 cache is available, the system benefit depends on the number and bandwidth of tier-1 caches available. In larger ISPs where a large tier-2 cache is available and where the request rate of items is high, the approach is only beneficial if the number or upload bandwidth of tier-1 caches is high enough.

3.4 Lessons Learned

To support content delivery networks, cache capacities on small data-centers with limited upload capacities, such as home routers, can be used to reduce the overall energy consumption and operation cost of the system. In this chapter we evaluate the performance of hierarchical content delivery networks that have a high number of caches with limited capacity in the lowest tier.

To this end, we first provide a comprehensive overview of literature on performance evaluation of caching systems and show the state of the art of current research. We describe the most relevant traffic models and caching strategies and introduce the Che approximation, which is a highly versatile method to determine the efficiency of caching systems.

We develop a simulation framework for hierarchical content delivery networks to evaluate the system characteristics that are not covered by the analytic models. This allows us to evaluated the approach in terms of caching efficiency and inter-domain traffic. The results show that an overlay is imperative for the success of such an approach, especially for a high number of small caches. Moreover, by investigating the share of locally served content requests, the impact for the network operator is quantified. The results indicate that such a mechanism significantly reduces the inter-domain traffic and the contribution of an operator-owned cache. The ISP owned cache can be discontinued if at least every thousandth user shares its home router for caching in large ISPs. The system was prototyped in [1] showing the practicability of the approach.

We develop an analytical model based on the Erlang formula for loss networks to evaluate the performance of hierarchical cache systems with small capacities and limited upload bandwidth. The results show that the efficiency of the overlay can be even further increased by more than 10% if an optimal content placement used. If the bandwidth is limited to 1.6Mbps the cache hit rate is reduced by about 30%, which shows the importance of considering bandwidth constraints in the analysis. There is a high potential to increase the efficiency of the content delivery network if only a small or no ISP cache is available. If a larger ISP cache is available the benefit of the approach highly depends on the number of caches available and their upload bandwidth. In the considered case at least 1000 tier-1 caches with more than 1.6Mbps upload bandwidth need to be available in a system with a large ISP cache so that the hit rate gain is more than 10%.

4 Bandwidth Aggregation Systems

In 2015, mobile networks carried more than 40 exabytes of traffic, which is expected to increase 8-fold towards 2020 [92]. To handle the growth and to reduce the load on mobile networks, offloading to WiFi has come to the center of industry thinking [93].

In contrast to strict offloading, in which the Internet access link is switched completely, e.g., from cellular to WiFi, current concepts such as BeWifi[1] also consider multiple connections to the Internet, thereby sharing and aggregating available backhaul access link capacities. The question is which sharing policy to apply for which system characteristics. In the case of BeWifi, which considers access link sharing among neighboring users, each user should only share its access link when having spare capacity in order to avoid negatively affecting his own Internet connections. Therefore, two thresholds were introduced, i) a support threshold until which utilization a user will offer bandwidth to other users, and ii) an offloading threshold indicating from which utilization a user can offload to supporting neighbors. It is hard and non-intuitive to determine the threshold settings for fair and effective operation of a bandwidth sharing system. In this work a partial bandwidth sharing environment with offloading policy is investigated using an analytic model. A direct application of the model is the aggregation of backhaul bandwidth by connecting neighboring access links.

We develop a Markov model to analyze the bandwidth aggregation potential of neighboring access links. The Markov model is limited to two access links only, which limits its applicability. It was shown in pilot studies that the technology's only limitation is the actual WiFi bandwidth available. In urban envi-

[1] `http://www.bewifi.es/`

ronments there are far more than two access links available. As shown in [94], an average of 25 WiFi access points are visible in every scan in densely-populated areas. In this case an assessment with the model previously proposed by the authors is not possible, since it is limited to two access links. An extension of the Markov model to m dimensions would require solving an equation system with n^m equations, which is computationally too complex. We extend the Markov model to be applicable for two and more links using a fixed point approximation. This allows us to reduce the n-dimensional Markov chain to evaluate the steady state probabilities efficiently.

The contribution of this chapter is three-fold. First, the approximation using fixed point iteration can be used to seamlessly evaluate the performance of systems between partitioning and complete sharing dependent on the threshold settings. Second, by considering an outer and an inner composite system we are able to apply the method to the case of heterogeneous load, which is crucial to assess the full potential of the approach. Bandwidth sharing systems are designed to increase the throughput of systems that are currently overloaded by using spare bandwidth of underutilized links. In such situations the load on the links is highly heterogeneous. Our results show that an overloaded system can highly benefit, by receiving multiples of its own capacity, from spare bandwidth of underutilized cooperating systems. Third, we evaluate the robustness of the mechanism against free riders by prioritizing links and find that altruistic users may only lose slightly more bandwidth than in normal operation. This is important, since a bandwidth sharing system that is running an inefficient offloading policy may be exploited by free riders that claim spare bandwidth by offloading traffic, but do not share any of their own bandwidth.

The content of this chapter is mainly taken from [9, 12]. Its remainder is structured as follows. Section 4.1 summarizes offloading and bandwidth sharing systems and technologies. In Section 4.2, the model of a bandwidth aggregation system is described in detail and results of the performance evaluation are reported. In Section 4.3 the model is extended to consider imbalanced systems loads, while this chapter is concluded with lessons learned in Section 4.4.

4.1 Background and System Description

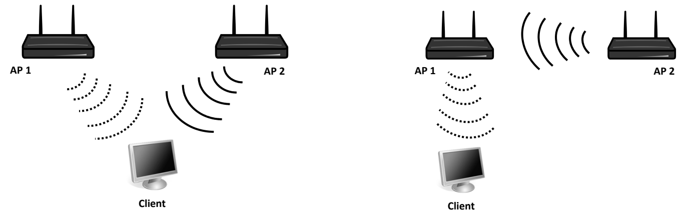
In this section we describe different bandwidth aggregation approaches in praxis and present related work on the performance evaluation of bandwidth aggregation systems. In addition, we describe our system model of bandwidth aggregation systems with offloading policy.

4.1.1 Bandwidth Aggregation Approaches

The principle of sharing or offloading between multiple Internet access links is already widely used by commercial services as well as research work. WiFi-sharing communities like Fon[2], Karma[3], WeFi[4], and Boingo[5] offer access to an alternative Internet link (WiFi instead of mobile), which provides a faster access bandwidth and reduces the load on stressed mobile networks. With respect to this so called WiFi offloading, the research community investigated incentives and algorithms for access sharing [95], and ubiquitous WiFi access architectures for deployment in metropolitan areas [96, 97]. Moreover, [98–100] describe systems for trust-based WiFi password sharing via an online social network (OSN) app. WiFi sharing is not a legal vacuum and a first exemplary overview on Swiss and French rights and obligations was given in [101] but must be treated with caution due to international differences and interim law revisions. The opposite concept to Wifi offloading, i.e., WiFi onloading, is presented in [102]. The idea is to utilize different peaks in mobile and fixed networks to onload data to the mobile network to support applications on short time scales (e.g., prebuffering of videos, asymmetric data uploads).

An access link sharing concept, which goes beyond pure offloading, is BeWifi, which was developed by Telefonica [103] and builds on previous works about backhaul capacity aggregation [104, 105]. BeWifi uses modified access points,

[2] http://www.fon.com
[3] https://yourkarma.com/
[4] http://wefi.com/
[5] http://www.boingo.com/



(a) Client based

(b) Access point based

Figure 4.1: Client based and access point based solution for bandwidth aggregation.

which act as normal access points until their clients saturate more than 80% of the backhaul capacity. Then, the access point will scan for close access points, which will provide additional bandwidth if their utilization is below 70%. Backhaul capacity and utilization are announced by each access point via beacon frames. Instead of introducing a secondary WiFi radio, BeWifi uses time-division multiple access (TDMA) and the 802.11 network allocation vector (NAV) to connect to neighboring access points for bandwidth aggregation in a round robin fashion with a weighted proportional fairness schedule.

Figure 4.1 shows a client based and an access point solution for the bandwidth aggregation proposed in [103]. The state of the art client-based system proposes to use a TDMA based access strategy for accessing selected access points in range in a round robin fashion, i.e., no concurrent data transmission via different frequencies is taking place. The system utilizes inband signalling, a switching frequency of 100ms and requires less than 1.5ms for switching. Using the standard 802.11 power saving feature, a client is able to notify its absence to the access points it is connected to, so that they buffer packets directed to it. A client performing aggregation appears to be sleeping in all access points but the one that is currently scheduled in the round robin cycle. The access point based solution can be mapped to the client based solution, if an access point acts as access point to its clients, and as a client to neighboring access points.

From a technical perspective, bandwidth sharing and offloading are enabled by implementing handovers and/or multipath connections, which are well covered in research. [106–108] show the feasibility of multipath TCP for handovers between mobile and WiFi networks in the current Internet and [109] describes available features for mobile traffic offloading. Futhermore, [110] gives an overview on approaches that enable mobility and multihoming.

4.1.2 Perfomance Models for Bandwidth Aggregation Systems

Theoretically, bandwidth sharing between WiFi access points can be considered as load sharing among systems. Generally load sharing systems can be classified in partitioning, partial sharing and complete sharing systems. Partitioning systems work completely independent from each other. Each system has its own queue and buffer space and processes only requests arriving at its queue. Complete sharing systems have a shared queue and buffer space. When processed, a request in the shared queue is assigned to the system which is currently least loaded.

Partial sharing systems have their own queues, but may offload requests to other systems if they are overloaded, or process requests from other overloaded systems. Different partial sharing or complete sharing models have been investigated in literature. In [111] the bandwidth usage by different services in a broadband system in complete sharing and partial sharing mode with trunk reservation is investigated. Multidimensional Markov chains are used in [112–114] to evaluate the performance of cellular network systems with different service categories. The blocking probability of a complete sharing system has been approximated in [115]. This approximation is used in [116] to evaluate the performance of mobile networks with code division multiplexing supporting elastic services. However, none of the models can be used to seamlessly evaluate the performance of systems between partitioning and complete sharing.

Thus, we develop a model based on a two dimensional Markov chain with thresholds to study the transition of blocking probabilities of partitioned, partial sharing, and complete sharing systems. This limits its applicability, since the number of average WiFi access points visible to clients is much higher in densely-populated areas. In densely-populated areas bandwidth of a high number of WiFi access points is aggregated. In this case an assessment with the Markov model is not possible, since it is limited to two access links. An extension of the Markov model to m dimensions would require solving an equation system with n^m equations, which is computationally too complex. Therefore, we extend the model to be applicable to multiple access links by utilizing a fixed point approximation.

The fixed point approximation is used to reduce the m-dimensional Markov chain to one dimension similar to [117, 118], where the approach is used for analytic models for polling systems and the interference distribution in UMTS networks, respectively. The underlying Markov chain highly differs from existing fix-point approaches, since it considers support and offloading thresholds. To the best of our knowledge this is also the first work that considers an inner and an outer composite system to apply the fixed point analysis in heterogeneous load conditions.

4.1.3 System Model

For simplicity and mathematical tractability we make assumptions on the link capacities and the service rates of bandwidth fractions. This allows analytic performance evaluation of bandwidth aggregation systems with offloading policy and understanding its characteristics.

Assumption 1. *The switching time to another access link is zero.*

In practice, TDMA is used to aggregate the bandwidth of two access points operating on different channels. During the time in which the client is switching frequencies, it cannot send or transmit data. This time is called switching time and for state of the art systems it is 1.5ms [103]. This switching time slightly decreases the effective throughput of the system. Signaling among the cooperating access points is necessary to report the current load and the offloading state. The messages exchanged produce a signaling overhead, which can limit the performance of the system. In practice APs announce their backhaul link capacity through Beacon frames, as well as their available-for-aggregation throughput, i.e. the part of their capacity that is not utilized by their clients [103]. However, in [103] the aggregate throughput remains almost constant across the different experiments, indicating that the overhead of switching and signaling is fixed and only slightly impacts the overall throughput.

Assumption 2. *The wireless channels are clean.*

Interference can limit the capacity of the wireless links. The effect of the channel quality on the aggregation capacity is evaluated in [103]. To account for a bad channel quality in our model, the link capacity can be reduced accordingly.

Assumption 3. *The service time of bandwidth fractions follows a negative exponential distribution.*

We model the load on $m \geq 2$ access links as depicted in Figure 4.2. The throughput of each Internet connection is limited by a bottleneck (either on application side, on server side, or in the core Internet), such that single connections will utilize a certain share of the access link bandwidth. Therefore, the available capacity of a link c is divided into a number n of small atomic bandwidth fractions of equal size. This means, $c = n \cdot \xi$ with a global constant ξ denoting the granularity of bandwidth allocation. Thus, different capacities c_i are modeled by assigning different n_i to the links.

We consider the system in a short time frame, where the system load can be considered stationary. Each access link is modeled as a multi-server blocking system, in which each server represents an available bandwidth fraction of the link. Its utilization variations are modeled as a stationary process of singular and independent arrivals of traffic bursts, i.e., bandwidth fraction requests. This allows modeling an access link as M/M/n loss system [119]. We define X as the random variable of the number of occupied bandwidth fractions on each

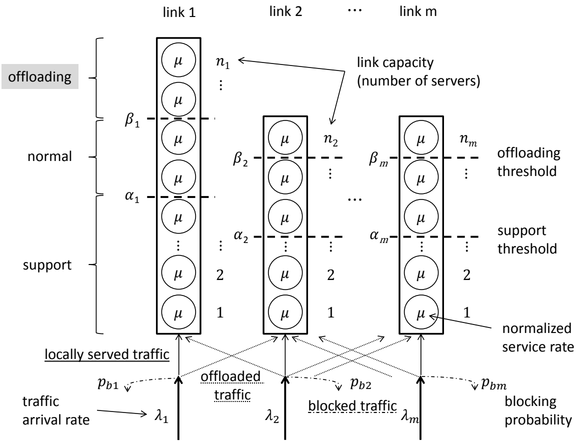


Figure 4.2: System model.

backhaul link. It is modeled by a birth-death-process, in which bandwidth fractions are requested with Poisson arrivals at rate λ and occupied for an negative-exponentially distributed service time with globally normalized rate $\mu = 1$. Consequently, the load on each link is given by $\rho = \frac{\lambda}{n \cdot \mu} = \frac{\lambda}{n}$. The probability that k bandwidth fractions are occupied in the considered M/M/n queue is $x(k) = P(X = k)$.

In the BeWifi approach (cf. Section 4.1.1), two thresholds are used, which define the bandwidth aggregation/offloading policy. The support threshold α indicates up to which percentage of utilization (i.e., number of own occupied bandwidth fractions) the system will offer bandwidth fractions to other systems. Furthermore, the offloading threshold β with $\alpha \leq \beta$ sets the percentage of utilization above which the system will try to use bandwidth of other systems. According to these thresholds, a system can be in one of the following three macro states:

a) *support* ($0 \leq X < \lfloor \alpha \cdot n \rfloor$):
low utilization and offering bandwidth

b) *normal* ($\lfloor \alpha \cdot n \rfloor \leq X < \lfloor \beta \cdot n \rfloor$):
normal operation

c) *offloading* ($\lfloor \beta \cdot n \rfloor \leq X \leq n$):
high utilization and offloading to other systems

By applying the offloading policies, different Internet access links will collaborate and share traffic. More details on the investigated scenarios are presented in the following section.

Two bandwidth aggregation systems, i.e., systems offloading between m access links, will be analyzed. First, we consider a bandwidth aggregation system with equal load on each access link. Moreover, a system in which one access link has a different load than the other $m - 1$ links is modeled. As reference system we considered partitioned systems without offloading.

4.1.4 Analysis of Reference Systems

We compare the bandwidth aggregation gain of multiple collaborating Internet access links to a partitioned system without offloading. The received bandwidth of each access link $E[X_i]$ and the blocking probability p_{b_i} of each system i are evaluated. The blocking probability gives the probability that the link is fully utilized and a bandwidth request of an application cannot be entirely satisfied. In practice, if TCP is used on the access link, the Internet connections throttle themselves and share the link equally. Depending on the used application and its characteristics, the application performance can then suffer, which can result in user dissatisfaction.

Partitioned and Complete Sharing Systems

For completely partitioned systems, i.e., m different M/M/n_i loss systems with arrival rates $\lambda_i, \ i \in \{1, \dots, m\}$, the received bandwidths $E_0[X_i]$ can be computed individually for each access link by Little's Theorem as

$$E_0[X_i] = \frac{\lambda_i}{\mu} \cdot (1 - p_{b_i}), \tag{4.1}$$

in which we use the rate of accepted arrivals $\lambda_i \cdot (1 - p_{b_i})$ and the globally normalized service rate $\mu = 1$.

The blocking probability of partitioned systems p_{b_i} follows from the Erlang-B formula [119]

$$p_{b_i} = \frac{\frac{(\frac{\lambda_i}{\mu})^{n_i}}{n_i!}}{\sum_{k=0}^{n_i} \frac{(\frac{\lambda_i}{\mu})^k}{k!}} \,. \tag{4.2}$$

The performance $E_s[X]$ of a complete sharing system, i.e., a single M/M/n loss system with $n = \sum_{i=1}^{m} n_i$ servers and an arrival rate of $\lambda = \sum i = 1^m \lambda_i$, can be computed by the same formulae.

Approximations and Performance Metrics

An approximation $\tilde{p}_b$ of the blocking probability p_b can be calculated by the joint probability of a single system being fully occupied, while a separate single system is above the support threshold α, i.e. could not help. If X_1 and X_2 are random variables for the number of jobs in system 1 and system 2, the joint probability is

$$\tilde{p}_b = P(X_1 = n_1, X_2 \geq \alpha \cdot n_2) = P(X_1 = n_1) \cdot P(X_2 \geq \alpha \cdot n_2) \,. \tag{4.3}$$

Moreover, we analyze the mean total number of occupied bandwidth fractions $E[X]$, which corresponds to the mean of total aggregated bandwidth. Following the same argumentation as above, $E[X]$ can be computed by Little's Theorem as

$$E[X] = \frac{\lambda_1 + \lambda_2}{\mu} \cdot (1 - p_b) = \frac{\lambda_1}{\mu} \cdot (1 - p_{b_1}) + \frac{\lambda_2}{\mu} \cdot (1 - p_{b_2}) \,. \tag{4.4}$$

Finally, we take a look at the received bandwidth at each access link $E[X_{A_i}]$. Thereby, X_{A_i} is a random variable for the number of bandwidth fractions (in all systems), which are occupied by arrivals from system i. It is obvious that $E[X_{A_i}] = E[X_i] = E_0[X_i]$ for the partitioned system. In case of offloading, $E[X_{A_i}]$ can be calculated from the mean total number of occupied bandwidth fractions by taking into account the share of accepted requests from each system.

$$E[X_{A_i}] = \frac{\lambda_i(1 - p_{b_i})}{\lambda_1(1 - p_{b_1}) + \lambda_2(1 - p_{b_2})} \cdot E[X] = \frac{\lambda_i}{\mu} \cdot (1 - p_{b_i}) \tag{4.5}$$

Nevertheless, it is the goal of bandwidth aggregation to cooperate in order to use spare capacity on access links to increase the received bandwidth where needed. Therefore, we can quantify the percentage of bandwidth gain for each system as

$$\omega_i = \frac{E[X_{A_i}] - E_0[X_i]}{E_0[X_i]} \,. \tag{4.6}$$

4.1.5 Simulation Description

A discrete-event based simulation using arrival and departure events is implemented to validate the analytic model and to assess the system performance in more general cases. Each of the m systems has a Poisson arrival process with rate according to its load. The service time of bandwidth fractions is exponentially distributed with mean 1. Offloading decisions are made according to the the number of occupied bandwidth fractions in the systems with respect to the support and offloading threshold. Therefore, the simulation state holds the requests being processed and the number of occupied bandwidth fractions for each system.

4.2 Potential of Backhaul Bandwidth Aggregation

In order to investigate the system dynamics of bandwidth aggregation systems with offloading policy, we develop analytical models.

In Section 4.2.1 we introduce a Markov model for two links. The Markov model is extended in Section 4.2.2 to be applicable for two and more links using a fixed point approximation, allowing us to reduce the n-dimensional Markov chain to evaluate the steady state probabilities efficiently.

The models are used in Section 4.2.3 to study the characteristics of bandwidth aggregation systems and to determine optimal threshold settings for the offloading policy.

4.2.1 The Case of Two Systems

We first consider a scenario with two different Internet access links. In the case of two links, the actual system state can be described by two random variables X_1 and X_2, which represent the number of occupied bandwidth fractions in the respective access link. As the model components comprise the memoryless property, a two-dimensional Markov process can be analyzed using standard techniques of queueing theory.

With the state probabilities

$$x(i,j) = P(X_1 = i, X_2 = j),\ 0 \leq i \leq n_1, 0 \leq j \leq n_2, \tag{4.7}$$

i.e., the probability that i bandwidth fractions are occupied in system 1 and j bandwidth fractions are occupied in system 2, the two-dimensional state transition diagram, presented in Figure 4.3, can be arranged. Two major areas are visible. In the upper left part and the lower right part (white background), each system operates independently in such way that all arriving requests are served locally by this system. In the top-right and bottom-left parts (shaded in gray), one of the links is in offloading state and the other link is in support state. In these cases, all traffic arriving at the offloading link will be served by the sup-

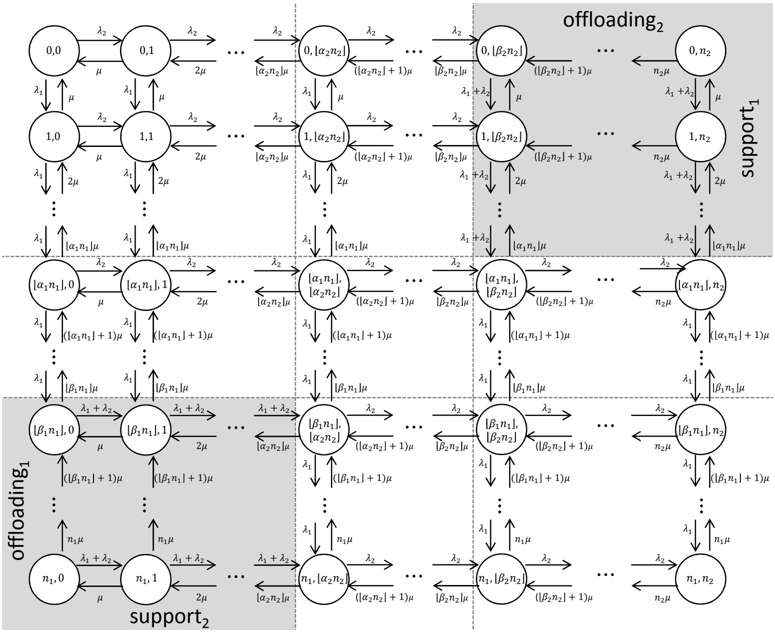


Figure 4.3: The state transition diagram.

porting link. Thus, blocking only occurs when the other link cannot help, i.e., in states $\{(n_1, j) : \lfloor \alpha_2 n_2 \rfloor \leq j \leq n_2\}$ and $\{(i, n_2) : \lfloor \alpha_1 n_1 \rfloor \leq i \leq n_1\}$.

Having the state probabilities, we calculate the blocking probability p_{b_i} of each system i and the total blocking probability p_b, which is the sum of blocking probabilities of each system weighted by the probability that a request arrives at each respective system.

$$p_{b_1} = \sum_{k=\lfloor \alpha_2 \cdot n_2 \rfloor}^{n_2} x(n_1, k), \quad p_{b_2} = \sum_{k=\lfloor \alpha_1 \cdot n_1 \rfloor}^{n_1} x(k, n_2) \tag{4.8}$$

$$p_b = \frac{\lambda_1}{\lambda_1 + \lambda_2} \cdot p_{b_1} + \frac{\lambda_2}{\lambda_1 + \lambda_2} \cdot p_{b_2} \tag{4.9}$$

Since requests can be offloaded from system 1 to system 2 in states (n_1, k) for $k < \lfloor \alpha_2 \cdot n_2 \rfloor$, the requests are not blocked and the state probabilites are not added to the blocking probability p_{b_1}. The same holds for states (k, n_2) with $k < \lfloor \alpha_1 \cdot n_1 \rfloor$ and p_{b_2}.

4.2.2 The Case of Multiple Systems

The case, in which m Internet access links offload traffic according to the policy defined via the support and offloading thresholds, is more interesting, since much more than two access links can be available in densely populated neighborhoods and since the potential of the approach increases with the number of links available for bandwidth sharing. We start with assuming that all access links are equal ($n = n_i, \forall i \in \{1, \dots, m\}$) and face equal loads ($\lambda = \lambda_i, \forall i \in \{1, \dots, m\}$) and policies ($\alpha = \alpha_i, \beta = \beta_i, \forall i \in \{1, \dots, m\}$). First, we distinguish one access link, and merge the remaining $m - 1$ cooperating access links into a composite system. This reduces the problem of m systems to two systems. Still, the complexity of the composite system prohibits creating and analyzing the two-dimensional state transition diagram as it was done in [9]. Thus, we apply a fixed point approach to analyze this system.Therefore, we model an observed system, which will take into account offloading to and supporting the abstract composite system. For simplifying the notation, we define the macro state probabilities p_1 (support), p_2 (normal), and p_3 (offload):

$$\begin{aligned} p_1 &= \sum x(i), 0 \leq i < \lfloor \alpha \cdot n \rfloor \\ p_2 &= \sum x(i), \lfloor \alpha \cdot n \rfloor \leq i < \lfloor \beta \cdot n \rfloor \\ p_3 &= \sum x(i), \lfloor \beta \cdot n \rfloor \leq i \leq n \end{aligned} \tag{4.10}$$

In the support macro state, the arrival rate will be increased by λ_s, i.e., the arrivals that are offloaded by the composite system. λ_s can be computed as shown in Equation 4.11 from the multinomial probability that j of the $m-1$ links in the composite system are in offloading state, and k links in the composite system can support.

$$\lambda_s = \sum_{j=1}^{m-1} \sum_{k=0}^{m-1-j} \binom{m-1}{j} \binom{m-j-1}{k} p_3^j p_1^k p_2^{m-j-k-1} \frac{j\lambda}{k+1} \tag{4.11}$$

The arrival rate is decreased by λ_o in the offloading macro state when the composite system can support the observed system, i.e., at least one of the $m-1$ systems is in support macro state.

$$\lambda_o = (1-(1-p_1)^{m-1})\lambda \tag{4.12}$$

This gives new steady state equations for the observed system as described in Equation 4.13. As all access links have equal load, and thus, show a homogeneous behavior, not only the state probabilities of the observed system, but also of the $m-1$ systems in the composite system are influenced. Thus, the state probabilities of all m links can be obtained by computing the state probabilities of the observed system. Therefore, we initialize the observed system with equal state probabilities. Then, we iterate the offloading and support and normalize the state probabilities until a fixed point is reached.

$$x(i) = \begin{cases} \frac{x(i-1)(\lambda+\lambda_s)}{i\cdot\mu}, & 0 \le i < \lfloor \alpha \cdot n \rfloor \\ \frac{x(i-1)\lambda}{i\cdot\mu}, & \lfloor \alpha \cdot n \rfloor \le i < \lfloor \beta \cdot n \rfloor \\ \frac{x(i-1)(\lambda-\lambda_o)}{i\cdot\mu}, & \lfloor \beta \cdot n \rfloor \le i \le n \end{cases} \tag{4.13}$$

$$\sum_{i=0}^{n} x(i) = 1$$

For our modeled bandwidth aggregation system with m Internet access links, we consider the blocking probability $p_b = x(n)\cdot(1-p_1)^{m-1}$ of a link, which is calculated by the probability that a request arrives when the link is fully loaded (i.e., in state n) and none of the $m-1$ other links can support.

Moreover, we take a look at the received bandwidth at each access link $E[X_{A_i}]$. Thereby, X_{A_i} is a random variable for the number of bandwidth fractions (in all systems), which are occupied by arrivals from system i. It is obvious that $E[X_{A_i}] = E_0[X_i]$ for the partitioned system. In case of offloading between m equal links, $E[X_{A_i}] = \frac{\lambda}{\mu} \cdot (1-p_b)$ is equal for all links and can be calculated from the mean total number of occupied bandwidth fractions by taking into account the share of accepted requests. Finally, we also quantify the percentage of bandwidth gain for each system as

$$\omega_i = \frac{E[X_{A_i}] - E_0[X_i]}{E_0[X_i]} \; . \tag{4.14}$$

4.2.3 Numerical Examples and Threshold Setting

Using the model we aim to calculate numerical examples to evaluate the performance of the system in different scenarios. As parameters we study the load on the reference system ρ_1 and the load on the cooperating system ρ_2. We consider the blocking probability of the reference system p_{b_1} and the normalized received bandwidth of the reference system $E[X_{A_1}]/n_1$. To validate our model and to get a first assessment, we analyse the performance of systems with equal thresholds and compare the analytic results with the results obtained from simulation and those of simple reference systems. We consider the symmetric case with even load $\rho_1 = \rho_2$ to investigate the impact of the offloading thresholds and to optimize them. We then consider the asymmetric case to analyse the performance of systems with imbalanced load. We conduct parameter studies to find system configurations where one of the systems can highly benefit from offloading, e.g. by being prioritized. Finally we run simulations with different service time distributions to assess the system performance in more general cases.

Figure 4.4 shows the blocking probability dependent on the system load of two server groups with equal arrival processes. In this case the blocking probability

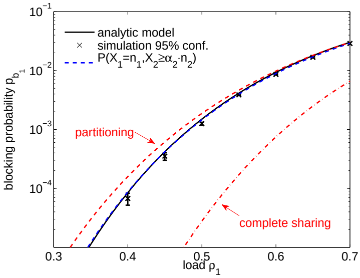
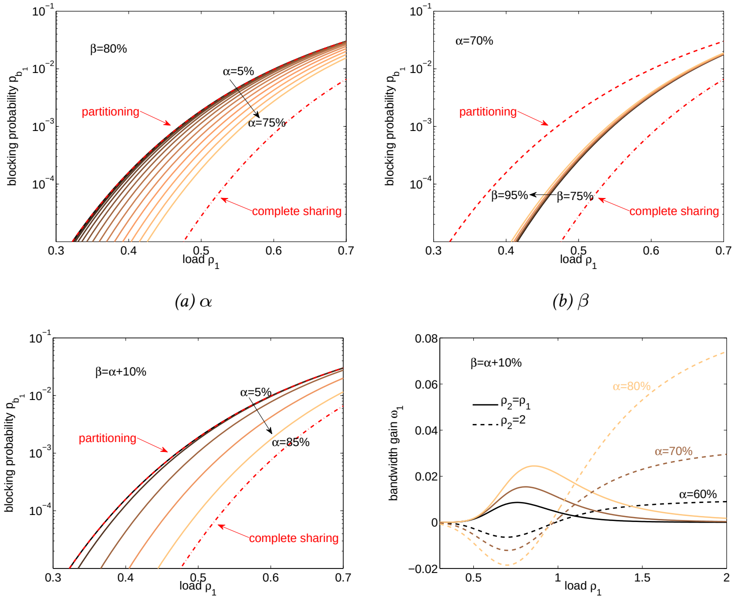


Figure 4.4: Blocking probability of two systems with equal load.

is equal for both systems. Both systems have $n = 20$ bandwidth fractions, and the thresholds are set to $\alpha = 40\%$ and $\beta = 80\%$. The black line shows the result based on the analytic model for a composite system as described in Section 4.2.1. The markers show the mean of 8 simulation runs with 95% confidence intervals. The blocking probability increases with the load on the system as expected. The results of the simulation match the analytical model with high confidence.

For comparison the analytic result for the approximation $\tilde{p}_b$, for partitioning and for complete sharing, i.e., with combined arrival process and bandwidth fractions, is plotted. The latter equals a system with a single server group, double arrival rate and double number of bandwidth fractions. Compared to partitioning the composite system performs slightly better for low loads. For low system loads the probability is high, that one of the two systems has less than $\alpha \cdot n$ active jobs and can help if the other system is in an offloading state. The load is taken from the highly loaded system and the blocking probability is decreased. This effect is negated for higher loads on the system, since the probability to be in a support state, with less than $\alpha \cdot n$ jobs, diminishes. If the systems cannot help each other, their performance equals partitioning the systems.

To investigate the potential of the system, it is compared to a complete sharing system. The red dash dotted line shows the result of a system with double arrival rate and $n' = 2 \cdot n = 40$ combined bandwidth fractions. The blocking



(a) α

(b) β

(c) α and β

(d) ω_1

Figure 4.5: Blocking probability p_{b_1} dependent on thresholds (a) α, (b) β and (c) α and β, and (d) bandwidth gain ω_1.

probability is reduced by a magnitude. This effect is also known as the economy of scale.

Offloading Thresholds

In the following we investigate the setting of the thresholds α and β to optimize the performance of the system. Therefore we analyse the symmetric case with $\rho_1 = \rho_2$ and vary the thresholds α and β. The number of bandwidth fractions per system is again set to $n = 20$.

Figure 4.5a shows the blocking probability of the reference system p_{b_1} dependent on the load ρ_1 for different support thresholds α. The offloading threshold β is constant at 80% of the system capacity. For $\alpha = 5\%$ a system only helps if it is empty and is not processing jobs. The systems work almost isolated from each other and thus the performance is equal to the performance of a single system. By increasing the support threshold α the systems can offer more help when one of the systems is overloaded and decrease the blocking probability. The support threshold α determines the amount of jobs that can be offloaded.

Figure 4.5b shows the blocking probability of the reference system p_{b_1} dependent on the load ρ_1 for different offloading thresholds β. The support threshold α is constant at 70% of the system capacity. The offloading threshold β is increased from 75% to 95%. Increasing the offloading threshold has almost no impact on the blocking probability. The effect on the blocking probability is small, since the threshold β just shifts the point of time at which the system starts offloading. The amount of jobs that can be offloaded is not dependent on β. The reason for the slight increase of the blocking probability with β is that there are less chances to find the cooperating system in support state when β is high.

We have seen that the performance of the system depends on the amount of jobs that can be offloaded, so the support threshold α needs to be set as high as possible. Theoretically, the support threshold could be set to the offloading threshold $\alpha = \beta$, so that a system would switch directly from support to offloading mode. However, in practice this may lead to problems, since the systems could switch unnecessarily frequently among the modes. This is especially the case if mode switches result in a high signalling overhead or imply expensive context switches. Therefore, a gap is left among the thresholds. Hence, in order to prevent frequent mode switches, we set $\beta - \alpha$ to 10%. In order to maximize the available bandwidth we can increase the support threshold α. Figure 4.5c shows the blocking probability of the reference system p_{b_1} dependent on the load ρ_1 with fixed gap $\beta - \alpha$ for increasing support thresholds α from 5% to 85%. The blocking probability decreases with increasing α, since more bandwidth fractions are shared among the systems. However, the performance of the system can also drop if the support threshold α is to high, which can be seen in Figure 4.5d. Figure 4.5d shows the bandwidth gain ω_1, c.f. Equ. 4.5, of the reference system for an equally loaded cooperating system with $\rho_2 = \rho_1$ and an overloaded cooperating system with $\rho_2 = 2$. If the cooperating system is equally loaded the bandwidth gain is always positive. If the cooperating system is overloaded, the bandwidth gain is negative, if the reference system is underutilized. In this case an increasing α has a negative effect on the bandwidth gain, because less bandwidth fractions are left for arrivals in the own system. To prevent the system from being overloaded, we leave 30% of the capacity as buffer for peak periods and set the support threshold α to 70%. Hence, we set the support threshold α to 70% and the offloading threshold β to 80% in the following.

4.3 Backhaul Bandwidth Aggregation in Imbalanced Load

To assess the full potential of the approach, we consider an inner and an outer composite system. Thus, we are able to apply the method to the case of heterogeneous load in Section 4.3.1, which allows evaluating the gain in situations where an overloaded system can use spare bandwidth of underutilized links Section 4.3.2. This further allows us to evaluate the fairness of the system and its robustness against free riders that try to exploit the system by receiving available bandwidth without contributing spare bandwidth to neighboring systems.

4.3.1 System Dynamics in Imbalanced Operation

Now, we consider the case of m systems, in which one link is different from the other $m-1$ links. Thus, we have the observed system with n_1 servers, arrival rate λ_1, and thresholds α_1, β_1, and a composite system of $m-1$ homogeneous links with $n' = n_i, \lambda' = \lambda_i, \alpha' = \alpha_i, \beta' = \beta_i, \forall i \in \{2, \ldots, m\}$. This gives two different macro state probabilities p_1, p_2, p_3 for the observed system and

p_1', p_2', p_3' for the systems in the composite system, respectively, which can be computed analogously to Equation 4.10. The corresponding support rate λ_{1s} and offloading rate λ_{1o} of the observed system can then be computed as follows:

$$\lambda_{1_s} = \sum_{j=1}^{m-1} \sum_{k=0}^{m-1-j} \binom{m-1}{j} \binom{m-j-1}{k} p_3'^{j} p_1'^{k} p_2'^{m-j-k-1} \frac{j\lambda'}{k+1} \tag{4.15}$$

$$\lambda_{1_o} = (1-(1-p_1')^{m-1})\lambda_1 \tag{4.16}$$

These rates of supported and offloaded traffic cannot be easily integrated into the fixed point iteration of Equation 4.13 as they depend on the state probabilities $x'(i)$ of links in the composite system, which are in this case different from the state probabilities $x(i)$ of the observed system. To obtain the $x'(i)$ values, we introduce an inner model. This means, we again distinguish one of the $m-1$ links of the outer composite system, and merge the remaining $m-2$ links to an inner composite system. Although this inner model resembles the case described above in Section 4.2.2, the equations for the inner observed system cannot be easily transferred, as the impact of the outer observed system cannot be neglected. Therefore, depending on the macro state of the outer observed system, the following support rate λ_s' and offloading rate λ_o' can be derived for the inner observed system:

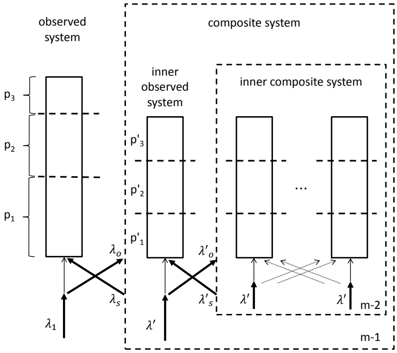


Figure 4.6: Fixed point approach for system with imbalanced load.

$$
\begin{aligned}
&\lambda'_s = \\
&p_1 \sum_{j=0}^{m-2} \sum_{k=0}^{m-2-j} \binom{m-2}{j} \binom{m-2-j}{k} p_3'^j p_1'^k p_2'^{m-2-j-k} \frac{j\lambda'}{k+2} + \\
&p_2 \sum_{j=0}^{m-2} \sum_{k=0}^{m-2-j} \binom{m-2}{j} \binom{m-2-j}{k} p_3'^j p_1'^k p_2'^{m-2-j-k} \frac{j\lambda'}{k+1} + \\
&p_3 \sum_{j=0}^{m-2} \sum_{k=0}^{m-2-j} \binom{m-2}{j} \binom{m-2-j}{k} p_3'^j p_1'^k p_2'^{m-2-j-k} \frac{j\lambda' + \lambda_1}{k+1}
\end{aligned}
\tag{4.17}
$$

$$
\lambda'_o = (1 - (1 - p_1)(1 - p'_1)^{m-2})\lambda' \tag{4.18}
$$

In Equation 4.17, the support rate λ'_s is computed for the case that the inner observed system can support. The summations consider the cases that j links want to offload and k links can support in the inner composite system. With probability p_1, the outer observed system is also in support macro state, thus, in total $k + 2$ systems can support (including the k systems from the inner composite system and both the inner and outer observed systems) and share the offloaded traffic $j\lambda'$. With probability p_2, the outer observed system is in normal macro state and will not interact. However, it is in offloading macro state with probability p_3, which means that the offloaded traffic is increased to $j\lambda' + \lambda_1$ and shared by $k + 1$ links. In contrast, the inner observed system can offload if the outer observed system is in support macro state, or at least one of the $m - 2$ links of the inner composite model can help, which is reflected by Equation 4.18.

Solving this system by a joint fixed point iteration for the outer and the inner system, i.e., iterating and normalizing in turns over both systems according to Equation 4.13, will give the state probabilities $x(i)$ and $x'(i)$.

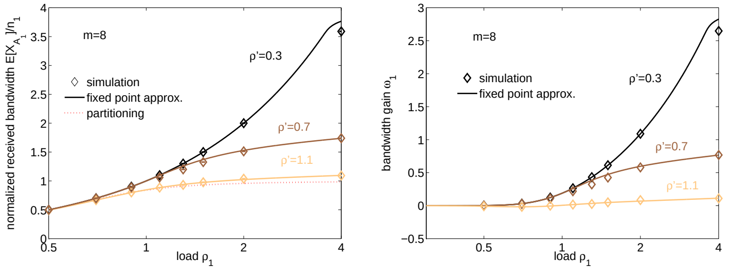
For the evaluation of this system, we will focus on the results for the outer observed link, i.e., the link that is not equal to the other $m - 1$ links. For this link we will investigate the blocking probability $p_{b_1} = x(n) \cdot (1 - p_1')^{m-1}$, the received bandwidth $E[X_{A_1}] = \frac{\lambda_1}{\mu} \cdot (1 - p_{b_1})$, and the bandwidth gain $\omega_1 = \frac{E[X_{A_1}] - E_0[X_1]}{E_0[X_1]}$.

4.3.2 Numerical Examples and Implications on Fairness

The cooperating system can benefit if the load is heterogeneously distributed among the systems, such that a system which is currently busy can offload to an idle system.

In order to assess the potential of bandwidth aggregation of m systems in heterogeneous load conditions, we study the load of the observed system ρ_1 and set the load of the other $m - 1$ systems to the same value ρ', i.e., $\rho_i = \rho', \forall i \in \{2, \ldots, m\}$.

In the following we investigate how the load on the links in the composite system ρ' affects the throughput of the observed system for $m = 8$ cooperating systems. Figure 4.7a shows the normalized received bandwidth of the observed



(a) Equal load

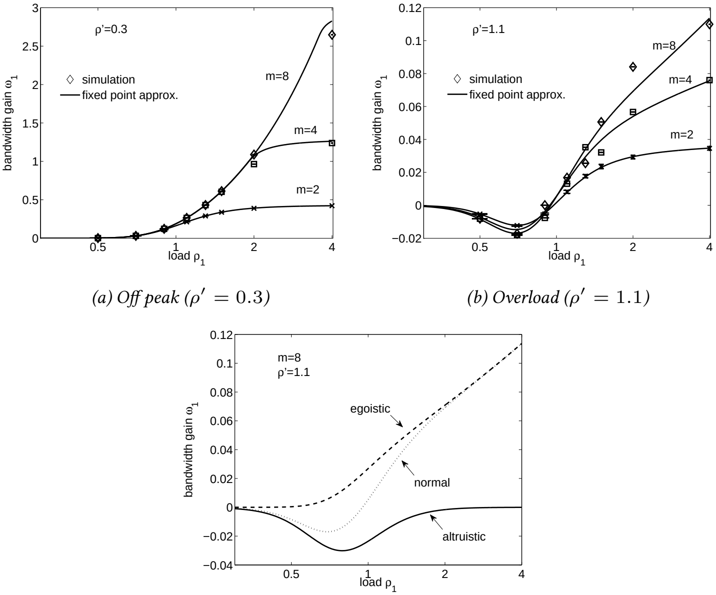
(b) bandwidth gain ω_1

Figure 4.7: Received bandwidth $E[X_{A_1}]/n_1$ dependent on load of the other links ρ' for $m = 8$

system dependent on the throughput of the links in the composite system ρ'. In case of $\rho' = 0.3$ a lot of spare bandwidth is available for offloading. If the observed system is overloaded it can use the spare bandwidth and receives almost 400% of its capacity if its load is 400%. If the load ρ' on the other links is higher, less bandwidth is available, which limits the received bandwidth. Still, the received bandwidth is above partitioning, although the links in the composite systems are overloaded with $\rho' = 1.1$ if the observed system is even more overloaded.

Figure 4.8a shows the bandwidth gain of the observed system ω_1 dependent on the number of cooperating systems m for $\rho' = 0.3$. Hence, in this case there is a high potential to obtain spare bandwidth from the cooperating systems. Depending on the number of cooperating systems the bandwidth gain of the observed system is limited.

Figure 4.8b shows the bandwidth gain of the observed system ω_1 dependent on the number of cooperating systems m for $\rho' = 1.1$. In this case the links in the composite system are overloaded. This leads to a loss of up to 2% bandwidth, if the observed system is not overloaded itself. If the load on the observed system is high, but low enough that it supports other systems, a traffic burst is more



(a) Off peak ($\rho' = 0.3$)

(b) Overload ($\rho' = 1.1$)

(c) Unfair: ($\alpha_1 \neq \alpha'$)

Figure 4.8: Bandwidth gain dependent on load of the observed system in off peak, overload and unfair operation.

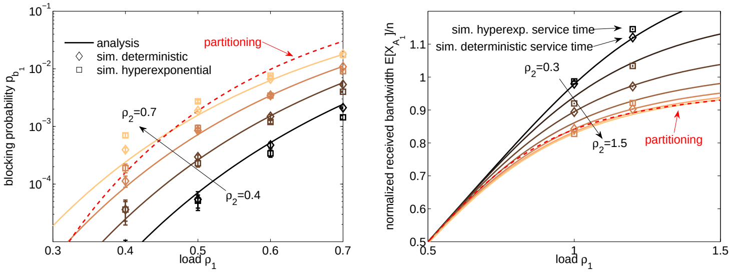
likely to block the system, since the overall load is higher than in the partitioning case.

To conclude, if the load on the other systems is low, an overloaded system can highly profit from their spare bandwidth by gaining multiples of its own bandwidth. The maximum bandwidth gain is limited by the number of cooperating systems m. If the cooperating systems are overloaded, the received bandwidth might be up to 2% lower in some cases, but this is compensated with multiples of the base level bandwidth in high peak periods.

To prevent a system from being congested from an overloaded cooperating system, it can be prioritized. One possibility of prioritizing is to decrease the support threshold α, so that it still can offload to other systems, but shares less bandwidth fractions to support. Figure 4.8c shows the bandwidth gain of the observed system for three cases. The dotted line shows the blocking probability if observed and other systems have equal support threshold $\alpha_1 = \alpha' = 70\%$. The solid line shows the case where the observed system is altruistic and keeps its threshold at $\alpha_1 = 70\%$, but interacts with egoistic cooperating systems with support threshold $\alpha' = 0\%$. The dashed line shows the egoistic case where the observed system limits its support threshold to $\alpha_1 = 0\%$, while the cooperating systems support up to $\alpha' = 70\%$. The altruistic system suffers from egoistic cooperating systems by losing up to about 3% bandwidth while not being able to gain bandwidth in high loads. Compared to that, the bandwidth gain in the egoistic case is never negative. Hence, if a system is egoistic it always gains more bandwidth. However, the gain compared to normal operation is not high, and if each system would be egoistic no bandwidth can be shared. This would mean completely partitioned systems which would not change the current situation without bandwidth sharing. On the other hand, if a system is the only one sharing among only free riders, which corresponds to the altruistic case, the situation is not worse, since only about 3% of the bandwidth are lost. Thus it is a win-win situation if everybody contributes to the system and shares spare bandwidth. This provides incentives for systems to contribute.

4.3.3 Simulation with General Service Times

To assess the system performance in more general cases we run simulations with different service time distributions. Figure 4.9a shows the blocking probability of the reference system dependent on the load of the systems. The mean values with 95% confidence intervals of 8 simulation runs are plotted for the service time distributions Deterministic and Hyper-exponential. The service times in the Deterministic process are constant. In the Hyper-exponential process we



(a) Blocking probability

(b) Received bandwidth

Figure 4.9: Blocking probability and received bandwidth dependent on the load of reference and cooperating system. Simulation with different service time distributions.

use two branches with probabilities 10% and 90%. For constant service times the blocking probability does not differ from the analytic model for high system loads. The blocking probability differs slightly from the analytic model for Deterministic service times in low system loads, showing higher blocking probabilities if the load on the cooperating system is high. The reason for this has to be investigated and is part of future work. In case of the Hyper-exponential distribution the service times are highly variant. Here the system which is highly loaded benefits from lower blocking probabilities compared to the analytic model.

In Figure 4.9b, which shows the available bandwidth of the reference system dependent on the load, simulation results are plotted for Deterministic distributed and highly variant Hyper-exponential distributed service times. For Deterministic service times the analytic model fits the simulation results. If the service times are highly variant the reference system receives only slightly more bandwidth than in the model if it is overloaded. Hence, considering the available bandwidth the analytic model can be used to assess the system performance with general service time distributions.

4.4 Lessons Learned

In this chapter we investigate the potential of bandwidth aggregation approaches with offloading policy. A direct application is the aggregation of backhaul link bandwidth to increase the overall capacity of the system to cope with the increasing demand of traffic.

To this end, we develop a simple Markov model that consists of an M/M/n loss system for each link. The offloading policy is modeled by introducing a support and an offloading threshold. In parameter studies we investigate the impact of thresholds that decide when a system offloads to a helping system or share bandwidth to support depending on its load. The threshold settings used in the BeWiFi system using a support threshold of 70% provides a good trade-off between providing spare bandwidth and leaving capacity as buffer for peak periods. Our results show that even if the cooperating system is overloaded, only up to 1% of the bandwidth is lost in off peak periods. The received bandwidth of a system can exceed its capacity significantly if the cooperating system is underutilized.

In practice the only limitation of bandwidth aggregation is the actual bandwidth available. The available bandwidth increases with the number of access links. In order to evaluate the performance of a system with multiple access links that share their bandwidth, we approximate the steady state probabilities of a multi dimensional Markov chain using a fixed point iterative procedure.

The full potential of the bandwidth aggregation approach is reached when an overloaded link can use the spare bandwidth of an underutilized link by exceeding its capacity significantly. A joint fixed point iteration of an outer and an inner composite system is used to derive the state probabilities in heterogeneous load conditions. In parameter studies we investigate the potential of the mechanism depending on the number of access links and find that in case of underutilized cooperating links, the bandwidth gain grows faster than linear with the number of contributing access links. By prioritizing links, we can show that the mechanism is robust against free riders, as cooperative users are not punished for sharing among only free riders, since only about 3% of the bandwidth are lost in the considered case studies. On the other hand, if each system would be egoistic no bandwidth can be shared and no one could profit. Thus the system provides incentives to contribute to increase the overall system capacity.

Finally, we validate our model by simulation and consider Deterministic and Hyper-exponential service time distributions to assess the system performance in more general cases. Our results show that only if the service time is highly variant and if one of the links is overloaded, only slightly more bandwidth is received by the overloaded link than in the model. Hence, the analytic model can also be used to assess the system performance with general service time distributions.

5 Conclusion

Acronyms

DES Discrete Event Simulation. 2

Bibliography and References

Bibliography of the Author

General References

[1] G. Petropoulos, A. Lareida, V. Burger, M. Seufert, S. Soursos, and B. Stiller, "Wifi offloading and socially aware prefetching on augmented home routers", in *40th Conference on Local Computer Networks (LCN) 2015*, Clearwater Beach, FL, USA: IEEE, Oct. 2015, pp. 1–3. [Online]. Available: https://files.ifi.uzh.ch/CSG/staff/lareida/Extern/Publications/rbhorst-demo.pdf.

[2] V. Burger, G. Darzanos, I. Papafili, and M. Seufert, "Trade-off between qoe and operational cost in edge resource supported video streaming", in *10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, Krakow, Poland, Nov. 2015.

[3] V. Burger, J. Frances Pajo, O. R. Sanchez, M. Seufert, C. Schwartz, F. Wamser, F. Davoli, and P. Tran-Gia, "Load dynamics of a multiplayer online battle area and simulative assessment of edge server placements", in *ACM Multimedia Systems Conference (MMSys)*, Klagenfurt, Austria, May 2016.

[4] V. Burger, M. Hirth, C. Schwartz, and T. Hoßfeld, "Increasing the coverage of vantage points in distributed active network measurements by crowdsourcing", in *Measurement, Modelling and Evaluation of Computing Systems (MMB 2014)*, Bamberg, Germany, Mar. 2014.

[5] V. Burger, D. Hock, I. Scholtes, T. Hoßfeld, D. Garcia, and M. Seufert, "Social network analysis in the enterprise: Challenges and opportunities", in *Socioinformatics - The Social Impact of Interactions between Humans and IT*, K. Zweig, W. Neuser, V. Pipek, M. Rohde, and I. Scholtes, Eds., Springer International Publishing, Aug. 2014.

[6] V. Burger, T. Hoßfeld, D. Garcia, M. Seufert, I. Scholtes, and D. Hock, "Resilience in enterprise social networks", in *Workshop Sozioinformatik 2013*, Koblenz, Germany, Sep. 2013.

[7] V. Burger, F. Kaup, M. Seufert, M. Wichtlhuber, D. Hausheer, and P. Tran-Gia, "Energy considerations for wifi offloading of video streaming", in *7th EAI International Conference on Mobile Networks and Management (MONAMI)*, Santander, Spain, Sep. 2015.

[8] V. Burger, F. Lehrieder, T. Hoßfeld, and J. Seedorf, "Who profits from peer-to-peer file-sharing? traffic optimization potential in bittorrent swarms", in *International Teletraffic Congress (ITC 24)*, Krakow, Poland, Sep. 2012.

[9] V. Burger, M. Seufert, T. Hoßfeld, and P. Tran-Gia, "Performance evaluation of backhaul bandwidth aggregation using a partial sharing scheme", *Physical Communication*, vol. 19, pp. 135–144, Jun. 2016.

[10] V. Burger, M. Seufert, F. Kaup, M. Wichtlhuber, D. Hausheer, and P. Tran-Gia, "Impact of wifi offloading on video streaming qoe in urban environments", in *IEEE Workshop on Quality of Experience-based Management for Future Internet Applications and Services (QoE-FI)*, London, UK, Jun. 2015.

[11] V. Burger and T. Zinner, "Performance analysis of hierarchical caching systems with bandwidth constraints", in *Proceedings of the International Telecommunication Networks and Applications Conference*, Dunedin, New Zealand, Dec. 2016.

[12] V. Burger, M. Seufert, T. Zinner, and P. Tran-Gia, "An approximation of the backhaul bandwidth aggregation potential using a partial sharing scheme", in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, Lisbon, Portugal, May 2017.

[13] T. Hoßfeld, V. Burger, H. Hinrichsen, M. Hirth, and P. Tran-Gia, "On the computation of entropy production in stationary social networks", *Social Network Analysis and Mining*, vol. 4, Apr. 2014.

[14] A. Lareida, G. Petropoulos, V. Burger, M. Seufert, S. Soursos, and B. Stiller, "Augmenting home routers for socially-aware traffic management", in *40th Annual IEEE Conference on Local Computer Networks (LCN)*, Clearwater Beach, FL, USA, Oct. 2015.

[15] D. Schlosser, M. Jarschel, V. Burger, and R. Pries, "Monitoring the user perceived quality of silk-based voice calls", in *Australasian Telecommunication Networks and Applications Conference (ATNAC 2010)*, Auckland, New Zealand, Nov. 2010.

[16] M. Seufert, V. Burger, and T. Hoßfeld, "Horst - home router sharing based on trust", in *Workshop on Social-aware Economic Traffic Management for Overlay and Cloud Applications (SETM)*, Zurich, Switzerland, Oct. 2013.

[17] M. Seufert, V. Burger, K. Lorey, A. Seith, F. Loh, and P. Tran-Gia, "Assessment of subjective influence and trust with an online social network game", *Computers in Human Behavior*, 2016.

[18] M. Seufert, V. Burger, F. Wamser, P. Tran-Gia, C. Moldovan, and T. Hoßfeld, "Utilizing home router caches to augment cdns towards information-centric networking", in *European Conference on Networks and Communications (EuCNC)*, Paris, France, Jun. 2015.

[19] M. Seufert, G. Darzanos, V. Burger, I. Papafili, and T. Hoßfeld, "Socially-aware traffic management", in *Workshop Sozioinformatik*, Koblenz, Germany, Sep. 2013.

[20] M. Seufert, G. Darzanos, I. Papafili, R. Łapacz, V. Burger, and T. Hoßfeld, "Socially-aware traffic management", in *Socioinformatics - The Social Impact of Interactions between Humans and IT*, K. Zweig, W. Neuser, V. Pipek, M. Rohde, and I. Scholtes, Eds., Springer International Publishing, Aug. 2014.

[21] M. Seufert, T. Griepentrog, and V. Burger, "A simple wifi hotspot model for cities", *IEEE Communications Letters*, vol. 20, no. 2, pp. 384–387, Feb. 2016.

[22] M. Seufert, T. Hoßfeld, A. Schwind, V. Burger, and P. Tran-Gia, "Group-based communication in whatsapp", in *1st IFIP Internet of People Workshop (IoP)*, Vienna, Austria, May 2016.

[23] I. Cisco Systems, *Cisco visual networking index: forecast and methodology, 2010-2015*, White Paper, 2011.

[24] Cisco, "Forecast and methodology, 2015–2020", *Cisco Visual Networking Index*, 2016.

[25] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic", in *ACM SIGCOMM Computer Communication Review*, 2010.

[26] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez, "Greening the internet with nano data centers", in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, ACM, 2009, pp. 37–48.

[27] P. Tran-Gia, T. Hoßfeld, M. Menth, and R. Pries, "Emerging issues in current future internet design", *E&i Elektrotechnik und Informationstechnik, Special Issue 'Future Internet'*, vol. 07/08, 2009.

[28] L. Gao, "On inferring autonomous system relationships in the Internet", *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, 2001.

[29] B. Cohen, "Incentives build robustness in BitTorrent", in *1st Workshop on the Economics of Peer–to-Peer Systems*, 2003.

[30] Google, *Peering & content delivery*, https://peering.google.com/.

[31] V. Adhikari, S. Jain, Y. Chen, and Z. Zhang, "Vivisecting youtube: an active measurement study", in *Proceedings IEEE INFOCOM*, 2012.

[32] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution", *IEEE Communications Magazine*, vol. 51, no. 4, pp. 142–149, 2013.

[33] B. Carna, *Internet Census 2012 Port Scanning /0 Using Insecure Embedded Devices*, 2013. [Online]. Available: `http://internetcensus2012.bitbucket.org/paper.html`.

[34] A. Dainotti and A. King, *Caida blog: Carna botnet scans confirmed.*

[35] T. Krenc, O. Hohlfeld, and A. Feldmann, "An internet census taken by an illegal botnet: A qualitative assessment of published measurements", *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 103–111, 2014.

[36] MaxMind, *Geolocation service.* [Online]. Available: `http://www.maxmind.com/`.

[37] International Telecommunication Union, "Ict facts and figures – the world in 2015", *Report*, 2015.

[38] F. Wamser, R. Pries, D. Staehle, K. Heck, and P. Tran-Gia, "Traffic characterization of a residential wireless Internet access", *Special Issue of the Telecommunication Systems (TS) Journal*, vol. 48: 1-2, 2010.

[39] T. Hoßfeld, F. Lehrieder, D. Hock, S. Oechsner, Z. Despotovic, W. Kellerer, and M. Michel, "Characterization of BitTorrent swarms and their distribution in the Internet", *Computer Networks*, vol. 55, no. 5, pp. 1197–1215, 2011. DOI: `10.1016/j.comnet.2010.11.011`.

[40] M. Kryczka, R. Cuevas, C. Guerrero, and A. Azcorra, "Unrevealing the structure of live BitTorrent swarms: Methodology and analysis", in *Proc. IEEE P2P*, 2011.

[41] M. Izal, G. Urvoy-Keller, E. W. Biersack, P. A. Felber, A. A. Hamra, and L. Garces-Erice, "Dissecting BitTorrent: Five months in a torrent's lifetime", in *Proc. PAM*, 2004.

[42] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, "The Bittorrent P2P file-sharing system: Measurements and analysis", in *Proc. IPTPS*, 2005.

[43] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, analysis, and modeling of BitTorrent-like systems", in *Proc. SIGCOMM IMC*, 2005, pp. 4–4.

[44] C. Zhang, P. Dhungel, D. Wu, and K. W. Ross, "Unraveling the BitTorrent ecosystem", *IEEE Transactions on Parallel and Distributed Systems*, vol. 99, 2010, ISSN: 1045-9219. DOI: http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.123.

[45] F. Lehrieder, G. Dán, T. Hoßfeld, S. Oechsner, and V. Singeorzan, "The impact of caching on BitTorrent-like peer-to-peer systems", in *International Conference on Peer-to-Peer Computing (P2P)*, 2010.

[46] ——, "Caching for BitTorrent-like P2P systems: A simple fluid model and its implications", *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, 2012. DOI: 10.1109/TNET.2011.2175246.

[47] V. Pacifici, F. Lehrieder, and G. Dán, "Cache capacity allocation for BitTorrent-like systems to minimize inter-ISP traffic", in *Proc. IEEE INFOCOM*, 2012.

[48] V. Aggarwal, A. Feldmann, and C. Scheideler, "Can ISPs and P2P users cooperate for improved performance?", *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, pp. 29–40, 2007.

[49] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang, "Improving traffic locality in BitTorrent via biased neighbor selection", in *Proc. ICDCS*, 2006.

[50] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz, "P4p: Provider portal for application", *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 351–362, 2008.

[51] S. Oechsner, F. Lehrieder, T. Hoßfeld, F. Metzger, K. Pussep, and D. Staehle, "Pushing the performance of biased neighbor selection through biased unchoking", in *Proc. IEEE P2P*, 2009.

[52] D. R. Choffnes and F. E. Bustamante, "Taming the torrent: a practical approach to reducing cross-ISP traffic in peer-to-peer systems", *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 363–374, 2008.

[53] C. Gkantsidis, T. Karagiannis, and M. VojnoviC, "Planet scale software updates", in *SIGCOMM Comput. Commun. Review*, vol. 36, 2006, pp. 423–434.

[54] R. Cuevas, N. Laoutaris, X. Yang, G. Siganos, and P. Rodriguez, "Deep diving into BitTorrent locality", in *Proc. INFOCOM*, 2011. DOI: 10.1109/INFCOM.2011.5935324.

[55] S. L. Blond, A. Legout, and W. Dabbous, "Pushing BitTorrent locality to the limit", *Computer Networks*, vol. 55, no. 3, pp. 541–557, 2011. DOI: 10.1016/j.comnet.2010.09.014.

[56] F. Lehrieder, S. Oechsner, T. Hoßfeld, D. Staehle, Z. Despotovic, W. Kellerer, and M. Michel, "Mitigating unfairness in locality-aware peer-to-peer networks", *International Journal of Network Management*, vol. 21, no. 1, pp. 3–20, 2011. DOI: 10.1002/nem.772.

[57] T. Böttger, F. Cuadrado, G. Tyson, I. Castro, and S. Uhlig, "Open connect everywhere: A glimpse at the internet ecosystem through the lens of the netflix cdn", *ArXiv preprint arXiv:1606.05519*, 2016.

[58] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao, "Dissecting video server selection strategies in the youtube cdn", in *31st International Conference on Distributed Computing Systems (ICDCS)*, 2011.

[59] V. Adhikari, S. Jain, and Z. Zhang, “Where do you "tube"? uncovering youtube server selection strategy”, in *IEEE ICCCN*, 2011.

[60] A. Rafetseder, F. Metzger, D. Stezenbach, and K. Tutschku, “Exploring youtube’s content distribution network through distributed application-layer measurements: A first view”, in *Proceedings of the 2011 International Workshop on Modeling, Analysis, and Control of Complex Networks*, 2011.

[61] Z. S. Bischof, J. S. Otto, M. A. Sánchez, J. P. Rula, D. R. Choffnes, and F. E. Bustamante, “Crowdsourcing isp characterization to the network edge”, in *Proceedings of the first ACM SIGCOMM workshop on Measurements up the stack*, 2011.

[62] P. Tran-Gia, T. Hoßfeld, M. Hartmann, and M. Hirth, “Crowdsourcing and its impact on future internet usage”, *It - Information Technology*, vol. 55, 2013.

[63] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, “Recaptcha: Human-based character recognition via web security measures”, *Science*, no. 5895, 2008.

[64] InnoCentive, Inc, *Innocentive*, http://www.innocentive.com/.

[65] T. Hoßfeld, M. Hirth, and P. Tran-Gia, “Modeling of crowdsourcing platforms and granularity of work organization in future internet”, in *In Proceedings of the International Teletraffic Congress (ITC)*, 2011.

[66] Microworkers, *Work & earn or offer a micro job*, http://microworkers.com/.

[67] MaxMind, *Geolite databases*, http://dev.maxmind.com/geoip/geolite/.

[68] M. Hirth, T. Hoßfeld, and P. Tran-Gia, “Anatomy of a crowdsourcing platform - using the example of microworkers.com”, in *Workshop on Future Internet and Next Generation Networks (FINGNet)*, Seoul, Korea, 2011.

[69] *The CAIDA AS relationships dataset, as-rel.2011.01.16.txt.* http://www.caida.org/data/active/as-relationships/.

[70] G. Yang and W. Dou, “An efficient algorithm for AS path inferring”, in *Proc. ICHIT*, 2009, pp. 151–154.

[71] *Internet topology collection*, http://irl.cs.ucla.edu/topology/.

[72] T. Hoßfeld et al., “An economic traffic management approach to enable the triplewin for users, ISPs, and overlay providers”, in *FIA Prague Book, ISBN 978-1-60750-007-0*, 2009.

[73] H. Che, Y. Tung, and Z. Wang, “Hierarchical web caching systems: Modeling, design and experimental results”, *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 7, pp. 1305–1314, 2002.

[74] V. Martina, M. Garetto, and E. Leonardi, “A unified approach to the performance analysis of caching systems”, in *INFOCOM, 2014 Proceedings IEEE*, IEEE, 2014, pp. 2040–2048.

[75] B. Tan and L. Massoulié, “Optimal content placement for peer-to-peer video-on-demand systems”, *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 2, pp. 566–579, 2013.

[76] E. G. Coffman and P. J. Denning, *Operating systems theory.* Prentice-Hall Englewood Cliffs, NJ, 1973, vol. 973.

[77] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini, “Temporal locality in today’s content caching: Why it matters and how to model it”, *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 5, pp. 5–12, 2013.

[78] M. Garetto, E. Leonardi, and S. Traverso, “Efficient analysis of caching strategies under dynamic content popularity”, *CoRR*, vol. abs/1411.7224, 2014. [Online]. Available: http://arxiv.org/abs/1411.7224.

[79] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, “Youtube traffic characterization: A view from the edge”, in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, ACM, 2007, pp. 15–28.

[80] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, “Analyzing the video popularity characteristics of large-scale user generated content systems”, *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 5, pp. 1357–1370, 2009.

[81] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, “Impact of traffic mix on caching performance in a content-centric network”, in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, IEEE, 2012, pp. 310–315.

[82] G. Hasslinger and K. Ntougias, “Evaluation of caching strategies based on access statistics of past requests”, English, in *Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, ser. Lecture Notes in Computer Science, K. Fischbach and U. Krieger, Eds., vol. 8376, Springer International Publishing, 2014, pp. 120–135, ISBN: 978-3-319-05358-5. DOI: `10.1007/978-3-319-05359-2_9`. [Online]. Available: `http://dx.doi.org/10.1007/978-3-319-05359-2_9`.

[83] M. Leconte, M. Lelarge, and L. Massoulié, “Adaptive replication in distributed content delivery networks”, *CoRR*, vol. abs/1401.1770, 2014. [Online]. Available: `http://arxiv.org/abs/1401.1770`.

[84] G. Rossini and D. Rossi, “Coupling caching and forwarding: Benefits, analysis, and implementation”, in *Proceedings of the 1st international conference on Information-centric networking*, ACM, 2014, pp. 127–136.

[85] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, “Optimal content placement for a large-scale vod system”, in *Proceedings of the 6th International COnference*, ACM, 2010, p. 4.

[86] E. J. Rosensweig, J. Kurose, and D. Towsley, “Approximate models for general cache networks”, in *INFOCOM, 2010 Proceedings IEEE*, IEEE, 2010, pp. 1–9.

[87] C. Fricker, P. Robert, and J. Roberts, “A versatile and accurate approximation for lru cache performance”, in *Proceedings of the 24th International Teletraffic Congress*, International Teletraffic Congress, 2012, p. 8.

[88] Y. Zhou, T. Z. J. Fu, and D. M. Chiu, “A unifying model and analysis of p2p vod replication and scheduling”, *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1163–1175, Aug. 2015, ISSN: 1063-6692. DOI: 10.1109/TNET.2014.2321422. [Online]. Available: http://dx.doi.org/10.1109/TNET.2014.2321422.

[89] M. Leconte, G. S. Paschos, L. Gkatzikis, M. Draief, S. Vassilaras, and S. Chouvardas, “Placing dynamic content in caches with small population”, *CoRR*, vol. abs/1601.03926, 2016. [Online]. Available: http://arxiv.org/abs/1601.03926.

[90] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graphs over time: Densification laws, shrinking diameters and possible explanations”, in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 177–187.

[91] Caida.org, *The CAIDA AS Relationships Dataset, 20150101*, 2015. [Online]. Available: http://www.caida.org/data/as-relationships/.

[92] Cisco, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015-2020”, Cisco, Tech. Rep., 2016.

[93] Wireless Broadband Alliance, “WBA Industry Report 2011: Global Developments in Public Wi-Fi”, Tech. Rep., 2011.

[94] P. Sapiezynski, A. Stopczynski, R. Gatej, and S. Lehmann, “Tracking human mobility using wifi signals”, *PloS one*, vol. 10, no. 7, e0130824, 2015.

[95] L. Mamatas, I. Psaras, and G. Pavlou, “Incentives and Algorithms for Broadband Access Sharing”, in *Proceedings of the ACM SIGCOMM Workshop on Home Networks*, New Delhi, India, 2010.

[96] N. Sastry, J. Crowcroft, and K. Sollins, “Architecting Citywide Ubiquitous Wi-Fi Access”, in *Proceedings of the 6th Workshop on Hot Topics in Networks (HotNets)*, Atlanta, GA, USA, 2007.

[97] P. Vidales, A. Manecke, and M. Solarski, “Metropolitan Public WiFi Access Based on Broadband Sharing”, in *Proceedings of the Mexican International Conference on Computer Science (ENC)*, Mexico City, Mexico, 2009.

[98] C. B. Lafuente, X. Titi, and J.-M. Seigneur, “Flexible Communication: A Secure and Trust-Based Free Wi-Fi Password Sharing Service”, in *Proceedings of the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Changsha, China, 2011.

[99] L. J. Donelson and C. W. Sweet, *Method, Apparatus and System for Wireless Network Authentication Through Social Networking*, US Patent App. 13/287,931, 2012.

[100] M. Seufert, V. Burger, and T. Hoßfeld, “Horst - home router sharing based on trust”, in *Proceedings of the Workshop on Social-aware Economic Traffic Management for Overlay and Cloud Applications (SETM)*, Zurich, Switzerland, 2013.

[101] G. Camponovo and D. Cerutti, “WLAN communities and Internet Access Sharing: A Regulatory Overview”, in *Proceedings of the International Conference on Mobile Business (ICMB)*, Sydney, Australia, 2005.

[102] C. Rossi, N. Vallina-Rodriguez, V. Erramilli, Y. Grunenberger, L. Gyarmati, N. Laoutaris, R. Stanojevic, K. Papagiannaki, and P. Rodriguez, “3GOL: Power-boosting ADSL using 3G OnLoading”, in *Pro-*

ceedings of the 9th Conference on Emerging Networking Experiments and Technologies (CoNEXT), Santa Barbara, CA, USA, 2013.

[103] E. Goma Llairo, K. Papagiannaki, and Y. Grunenberger, *A Method and a System for Bandwidth Aggregation in an Access Point*, WO Patent App. PCT/EP2012/064,179, 2013.

[104] S. Kandula, K. C.-J. Lin, T. Badirkhanli, and D. Katabi, "FatVAP: Aggregating AP Backhaul Capacity to Maximize Throughput.", in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, USA, 2008.

[105] D. Giustiniano, E. Goma, A. Lopez Toledo, I. Dangerfield, J. Morillo, and P. Rodriguez, "Fair WLAN Backhaul Aggregation", in *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Chicago, IL, USA, 2010.

[106] M. Gonzalez, T. Higashino, and M. Okada, "Radio Access Considerations for Data Offloading with Multipath TCP in Cellular/WiFi Networks", in *Proceedings of the International Conference on Information Networking (ICOIN)*, Bangkok, Thailand, 2013.

[107] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, "Exploring Mobile/WiFi Handover with Multipath TCP", in *Proceedings of the ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*, Helsinki, Finland, 2012.

[108] S. Chen, Z. Yuan, and G.-M. Muntean, "An Energy-aware Multipath-TCP-based Content Delivery Scheme in Heterogeneous Wireless Networks", in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Shanghai, China, 2013.

[109] Y. Khadraoui, X. Lagrange, and A. Gravey, "A Survey of Available Features for Mobile Traffic Offload", in *Proceedings of the 20th European Wireless Conference*, Barcelona, Spain, 2014.

[110] A. Gladisch, R. Daher, and D. Tavangarian, "Survey on Mobility and Multihoming in Future Internet", *Wireless Personal Communications*, vol. 74, no. 1, 2014.

[111] P. Tran-Gia and F. Hübner, "An analysis of trunk reservation and grade-of-service balancing mechanisms in multiservice broadband networks", in *IFIP-TC6 Workshop on Modelling and Performance Evaluation of ATM Technology*, Martinique, France, 1993.

[112] W.-Y. Chen, J.-L. Wu, and H.-H. Liu, "Performance Analysis of Radio Resource Allocation in GSM/GPRS Networks", in *Proceedings of the IEEE 56th Vehicular Technology Conference (VTC)*, Vancouver, BC, Canada, 2002.

[113] Y. Zhang, B.-H. Soong, and M. Ma, "A Dynamic Channel Assignment Scheme for Voice/data Integration in GPRS Networks", *Computer Communications*, vol. 29, no. 8, pp. 1163–1173, 2006.

[114] K.-W. Ke, C.-N. Tsai, and H.-T. Wu, "Performance Analysis for Hierarchical Resource Allocation in Multiplexed Mobile Packet Data Networks", *Computer Networks*, vol. 54, no. 10, pp. 1707–1725, 2010.

[115] J. Kaufman, "Blocking in a Completely Shared Resource Environment with State Dependent Resource and Residency Requirements", in *Proceedings of the IEEE INFOCOM*, Florence, Italy, 1992.

[116] G. Fodor and M. Telek, "Bounding the Blocking Probabilities in Multirate CDMA Networks Supporting Elastic Services", *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 4, pp. 944–956, 2007.

[117] P. Tran-Gia, "Analysis of polling systems with general input process and finite capacity", *IEEE Transactions on Communications*, vol. 40, no. 2, 1992.

[118] D. Staehle, K. Leibnitz, K. Heck, B. Schröder, A. Weller, and P. Tran-Gia, “An approximation of othercell interference distributions for umts systems using fixed-point equations”, University of Würzburg, Tech. Rep. 292, 2002.

[119] L. Kleinrock, *Queuing Systems.* Wiley, 1975.

ISSN 1432-8801