Trinity College Dublin

Coláiste na Tríonóide, Baile Átha Cliath

The University of Dublin

PhD Thesis

Optimizing Multimedia Content Delivery over Next-Generation Optical Networks

Author:
Emanuele Di Pascale

Supervisor:
Prof. Marco Ruffini

10th July 2015

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Signed:

_____

Emanuele Di Pascale 10th July 2015

Summary

This thesis analyzes the performance of a Peer-to-Peer (P2P) multimedia content delivery system for a network architecture based on next-generation Passive Optical Networks (PONs).

A PON is an optical access technology that is able to deliver high bandwidth capacities at a fraction of the cost of traditional point-to-point fiber solutions; this is achieved by sharing the same feeder fiber among several customers through the use of optical splitters. Established standards such as GPON and EPON have a reach from the Central Office of about 20 Km and a fan-out of around 32-64 users. Next-generation PONs aim to increase this reach, in order to enable the consolidation of central offices; to increase the split size, in order to help reducing the cost per customer; and to increase the bandwidth capacity to multiple 10 Gbps channels.

One of the reasons why operators are investing on Fiber-to-the-X (FTTX) solutions is to increase the capacity of the access section in order to remove what traditionally was the bandwidth bottleneck of the network. Over the last decade multimedia streaming services have become increasingly popular, with companies like Netflix, Amazon Prime and many more reaching millions of customers and billions of revenues. With both the number of active subscribers and the quality of the streamed videos steadily on the rise, network infrastructures are being put under an increasing strain to support these data-intensive services. Several reports have shown that the vast majority of the data currently traversing the Internet on a daily basis is related in one form or another to multimedia content retrieval.

Increasing the capacity in the access is certainly going to remove one of the obstacles to high-definition streaming of multimedia content. However, as users start to take advantage of the increased bandwidth allowance granted to them by fiber, the aggregation section of the network (i.e., the core) will start to suffer; and as we do not have any better technology than fiber, the only way we have to improve core capacity is to stack up network equipment, a process that is neither efficient nor cost-effective. It is hence imperative to find alternative solutions that will allow us to support bandwidth-intensive multimedia services while keeping operators' costs to a minimum, ensuring that these services are sustainable in the long term.

One promising strategy consists in placing caches managed by the Internet Service Provider (ISP) at the edge of the network. Once content has been delivered to a customer, it can be stored and redistributed to other users in the area to minimize bandwidth consumption in the core. More specifically, in this work we show that, by reserving a small amount (4-16 GB) of the storage space typically available

on most Set-Top Boxes (STBs), and by allowing users to cache multimedia content that they requested for their own personal consumption, we can greatly improve the efficiency of these next-generation networks. Indeed, the combined effect of the large symmetric upstream/downstream capacity of PONs and the customer aggregation brought by long reach feeders greatly increase the efficacy of locality-awareness – i.e., a strategy by which content requests are redirected to local available sources whenever possible. More specifically, symmetric access bandwidth means that a single source is in principle sufficient to provide all the upload capacity that a requester is able to handle; at the same time, the consolidated architecture that results from bypassing the metro section greatly expands the pool of potential sources attached to the access section of the requester, thus increasing the chances of a local data transfer. Furthermore, since the caches are managed by the network operator itself and integrated into the equipment required to connect to the Internet, they are less susceptible to churn and allow for a simpler implementation of locality-aware policies.

Extensive simulation campaigns, carried out first using a steady-state analyzer and subsequently through our custom event-driven simulator PLACeS, show that a locality-aware P2P strategy can confine most multimedia traffic inside the metro/core node from which the requests originated, thus drastically reducing core bandwidth utilization. An energy consumption model taking into account both static and dynamic consumption of electronic devices was formulated, showing that locality-aware P2P is able to reduce the overall power required to run the network and to offer cost-saving opportunities for operators.

In order to implement locality-aware policies for content delivery, we postulate the existence of an oracle service, whose responsibilities include keeping track of the content of each user cache and matching requests with a local available source whenever possible. This thesis explores some of the issues that might arise when designing such a service, and it proposes two proof-of-concept implementations based on OpenFlow. In particular, the aim of these implementations is to show that such an oracle could be designed to be transparent to the underlying multimedia applications, allowing it to be used in conjunction with legacy services that were never expected to be locality-aware. Furthermore, using OpenFlow allows us to integrate the oracle functionality with the control plane of the operator's network, i.e., allowing us to implement redirection policies that can react to anomalous load conditions.

Finally, we present an optimization algorithm to reduce the amount of caching storage required to implement our proposed P2P solution. The algorithm takes as input the number of requests observed by the oracle service for previous elements of the catalog, and estimates future request patterns for elements with the same popularity rank. This information is then used to determine which contents should be cached at each STB. Our simulations show that such a solution is able to achieve the same levels of locality of a traditional caching eviction policy (such as Least Frequently Used or LFU) while reducing the amount of storage required by up to 77%. Even higher storage savings of up to 92% can be achieved if we are willing to accept a reduction of about 5 to 8% in the percentage of requests served locally to the access segment of the requester.

Acknowledgements

First and foremost, many thanks go to my supervisor Prof. Marco Ruffini for giving me a chance when I wasn't sure I deserved one, for his endless support during these four years and his ability to come up with new avenues of investigation whenever I thought there were none left.

Secondly, to Prof. David B. Payne, for pioneering the LR-PON concept that is at the heart of my thesis, for the very useful insights in the early stages of my research, and for his encouraging words, always given with a smile.

My deepest thanks go to Prof. Linda Doyle for having put together such an amazing group of talented people, for being a constant inspiration of what can be achieved with hard work and ingenuity, and for generally contributing to making CTVR the best working environment EVER.

I would like to thank profusely all the post-docs that have helped me with their insights and their experience: in no particular order Seamas, Konstas, Baris, Nicola, Johann, Irene, Avishek, Tim, Hamed, Ahmed and many others. Sometimes a quick tip given over lunch or a coffee break can do miracles.

To "the guys" from the lab, without whom my time here would have been incredibly dull: in rigorous alphabetical order, Arman, Carlo, Danny, Eamon, Elma, Francisco, Frank, Ioannis, Jacek, Jasmina, Jessica, João, Johnathan, Justin, Paolo, Pedro, Sandeep. If somebody told me four years ago that one could have so much fun at work and still be productive I would have not believed it.

Thanks to all of the good friends that stuck with me despite the increasing distance, decreasing spare time and my constant grumpiness: Marco, Francesca, Alessandro, Giuliana, Amelia, Esther, Giovanni, Valerio, and all the others. As long as you are part of my life I know everything's going to be alright.

To the fine folks I've met in Dublin, for making this little cold piece of earth lost at sea so much cozier. To Michele+Claudia+Gaia, Luca, Antonello, Marco/Bob, and everyone else who sat at least once with me to play board games, tell stories, eat terrible stuff that goes under the name of pizza around these parts and escape to our secret worlds. To Walter, Esther, Kat, Micky&Micky, and all the other Couchsurfers that accompanied me on our discovery of Ireland. To Alex, Ana, Gaetano, Luciana, Stephen, for being generally awesome.

My never-ending gratitude goes to my family, to which I owe everything. To my father Michelangelo, my mother Ida, my brother Gaetano, for so many things that I cannot possibly begin to list them here, thank you.

Finally, and most importantly, my thanks to Francine. For teaching me things about myself that I

did not know before; for making me "more human", as someone noted, and at the same time accepting my cattiness; for loving me the way you do, and for letting me free, as you do. For being the home I can go back to at the end of the day; for being a *pedacinho*; for all these things and much more, from the bottom of my heart – thank you.

Contents

List of Figures

1 Introduction

This thesis investigates the benefits of a peer-to-peer based locality-aware caching system for on-demand multimedia distribution. We will show that, compared to state of the art strategies such as Content Delivery Networks (CDNs) and unicast streaming, our solution is able to reduce core bandwidth utilization, thus reducing the overall energy consumption of the system and providing cost-saving opportunities for network operators. Furthermore, we sketch an OpenFlow-based prototype of a Locality Oracle, i.e., a service whose function is to track the content of each distributed cache and to transparently match requests for videos to local available sources. Finally, we propose an optimization strategy to minimize the cache storage utilization while retaining the high percentages of locally available content achieved by traditional caching policies.

1.1 Overview and Motivations

Multimedia streaming services have become extremely popular in recent years, also thanks to the increasing deployment of Fiber-To-The-X (FTTX) technologies, which support the capacities required to deliver a satisfactory Quality of Service (QoS) to customers. Video on Demand (VoD) and IPTV services have become the biggest sources of Internet traffic (SANDVINE 2013), and this trend is only going to consolidate as more and more companies enter the fray to compete in this booming market segment. However, multimedia services still pose a number of challenges to current network infrastructures. The combination of heavy bandwidth requirements and high concentration of requests during peak-hours means that serving each user individually (i.e., through unicast) is very costly and inefficient. Multicasting approaches, on the other hand, need to employ special strategies to merge requests coming at slightly different times into a single stream; cyclic broadcasting, batching and patching have been proposed to address this issue, each technique with its own drawbacks (Choi et al. 2012).

Peer-to-Peer (P2P) is another very popular approach to multimedia content delivery. As every new peer in the system adds its own upstream capacity to the pool of common resources, it is inherently more scalable and cost-efficient for service providers than client-server approaches. On the other hand, the efficiency of P2P strategies is limited by the dynamic behavior of peers; because users are free to join or leave the network at any time, the availability of critical resources in the system cannot be guaranteed in general.

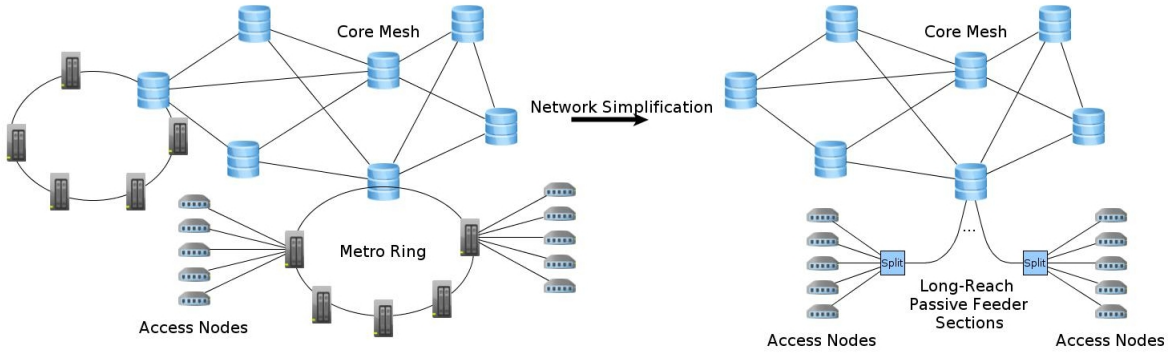

Figure 1.1: Thanks to their long reach, next-generation optical access technologies will enable node consolidation, allowing us to bypass traditional metro rings and aggregate customers through a limited set of metro/core nodes.

However, these limitations can be overcome if P2P exchanges are managed and coordinated by the network infrastructure itself, rather than by a user application. By using a small portion of the storage typically present in most customers' set-top boxes (or adding one when required), network operators and service providers can easily implement a cost-effective, always-on distributed caching system. The reduced gap between upstream and downstream capacity and the increased customer aggregation brought by current (GPON (ITU-T 2008), EPON (IEEE 2012), XG-PON (ITU-T 2010)) and next-generation access technologies (NG-PON2 (ITU-T 2013)), together with the decreasing price of mass storage devices, make such a network-managed P2P solution very promising. Furthermore, integrating peering mechanisms into the network infrastructure would greatly simplify the implementation of locality policies (Dai et al. 2010), thus improving the bandwidth and cost efficiency of this approach.

One fundamental advantage of locality-aware peer-to-peer based approaches to caching is that they allow us to turn around traffic as close as possible to the end user, without unnecessarily allocating network resources in the core. Traditionally, the Internet developed over the years into a hierarchical network, where a high-speed optical core aggregated traffic through statistical multiplexing, dispatching it to SDH/SONET metro rings and finally to copper-based access sections. However, FTTX is changing the rules of the game by greatly simplifying and "flattening" the structure of the network. Due to the low attenuation of fiber, Passive Optical Networks (PONs) are able to support a longer access reach, thus allowing us to bypass part or all of the metro transmission infrastructure (see Fig. 1.1).

This is particularly true for Long-Reach PON (Payne 2009), a technology whose aim is to achieve a reach in excess of 100 Km. This would enable a significant consolidation of the network distribution nodes: many of the local exchanges and central offices that are now required to support a large-scale network could be removed, with a consequent reduction of both capital and operational expenditure. The resulting architecture would only be comprised of a limited number of metro/core nodes, each providing access over fiber to a large number of users (see for example the DISCUS architecture discussed in Subsection 2.1.4).

A byproduct of these architectural modifications, however, is that now the same transmission tech-

nology (i.e., optical fiber) is installed in both the access and the core segments of the network. Since no better transmission technology than fiber currently exists, core bandwidth can only be increased by stacking more and more communication equipment, a practice that is neither effective nor sustainable from an economic perspective. Consequently, traffic aggregation in the core becomes less and less effective – a problem that is aggravated by the high concentration in time of multimedia requests. For these reasons, while FTTX is effectively removing the bandwidth bottleneck from the access section of the network, it is also shifting it towards the core. We believe that the synergy of P2P and next-generation long reach access architectures can solve (or at least strongly mitigate) the problem from an architecture point of view, rather than from a technology viewpoint. By turning around traffic at the edge of the network we can reduce the load on core links and thus improve the scalability of the system.

Another important metric is the power consumption associated with each content delivery strategy. Green ICT has been the subject of many research studies and continues to be a relevant topic. Peer-to-Peer is often believed to be more power-hungry than solutions based on Content Delivery Networks (CDN), due to the high power budget associated with many small storage devices in the users' premises. However, using P2P does not necessarily require *additional* storage devices: most set-top boxes nowadays already come with a hard drive or flash memory to allow for recording of TV shows or to provide a way to pause live programs. By taking advantage of a small portion of the space available on these drives, an operator might be able to reduce its energy bill without increasing the overall energy consumption.

1.1.1 Network-Managed P2P

The use of peer-to-peer to deliver multimedia content is certainly not a novelty: from live streaming application such as PPLive or SopCast to catch-up IPTV applications like the early versions of the BBC iPlayer, not to mention the large number of movies and TV-series downloaded every day through BitTorrent and other similar applications, P2P is widely used to disseminate video content both legally and illegally to millions of users.

Unfortunately, application-layer P2P services like the ones mentioned above have a number of drawbacks. Firstly, they can only take advantage of a peer's upload capacity when both its hardware and the P2P software running on it are turned on; as users continuously and dynamically enter or leave the P2P overlay network (*peer churn*), preserving QoS is not a trivial task. Secondly, P2P is typically very disruptive for Internet Service Providers (ISPs), which are called to cover the costs generated by the vast quantities of inter-AS traffic generated by these applications (see Section 3.1) without directly gaining any benefit from them (Dai et al. 2010). This leads to an arms race between ISPs trying to limit the effectiveness of these applications (e.g., through bandwidth throttling) and their developers trying to keep P2P exchanges undetected (e.g., through protocol obfuscation).

In this thesis, we investigate the merits of a P2P content delivery service managed by the network operator itself, either in partnership with a Video Service Provider (e.g., selling distributed caching

space to content providers interested in reaching its customers) or privately to enter the content delivery market, like many current operators offering "triple play" services. On one hand, this facilitates the use of dedicated set-top boxes (STBs), through which we expect to see a greatly reduced peer churn compared to a purely software-based application (for a discussion on the implications of removing the always-on assumption for STBs, please refer to Sec. 3.2.5); on the other, we can exploit the knowledge of the underlying network infrastructure to ensure that data exchange is performed in the most efficient way, both for the user (e.g., guaranteeing required QoS levels) and for the network (e.g., using locality-awareness to reduce core traffic and ISP costs).

It is worth noting that ISP-ownership is not strictly required to implement the service described here. It is possible to envision a cross-layer approach with a P2P application developed by the Service Provider (e.g., Netflix, Amazon Prime etc.) coordinating with the ISP's network infrastructure, e.g., through the use of the ALTO protocol (Alimi et al. 2013) or similar options. However these approaches, while technologically mature, are still mostly confined to research environments due to the natural mistrust between P2P application developers and network operators, with both parties reluctant to share what they consider sensitive information (Stiemerling & Kiesel 2013). ISP-ownership solves this issue while at the same time greatly simplifying the design and deployment processes, as the service can benefit from the full integration with the network management and control planes.

At the same time, there are some economical factors supporting an ISP-run, P2P-based content delivery system. Next generation PONs will offer upstream capacities which are in principle as large as their downstream equivalent. For example, a 10 Gbps PON can offer a sustained bandwidth between 160 and 20 Mbps, respectively for split sizes between 64 and 512. It is currently hard to envision applications that can make extensive use of such upstream bandwidth; this excess upstream capacity (due to the fact that every device in the access and core network uses bidirectional interfaces with symmetric capacity) represents a fixed cost which brings no revenue to the operators in the current business model. For this reason, implementing services that could generate profit by exploiting this dormant capacity seems like a logical option. Similarly, reserving a portion of the storage space already available on most set-top boxes for other purposes represents a marginal cost.

1.2 Key Contributions

As a result of our work, the following contributions have been made.

1.2.1 Simulation Tools for the Study of Large-Scale Multimedia Delivery Services

The first step of our work was to assess the efficiency of our proposed solution for content delivery over a reference network scenario, with 10G symmetric capacity in the access and a consolidated architecture thanks to the reach of next-generation PONs. We call this the *LR-PON scenario*, because it represents

the target of this access technology. However, due to the large scale of typical on-demand multimedia services and the lack of any commercial LR-PON deployment, simulation was really the only viable investigation tool. Even so, simulating the watching patterns of a large number of users over weeks or months in a reasonable computational time was a significant challenge. **Chapter 3** details the steps that led to the development of PLACeS, the custom event-based network simulator that we designed and used throughout most of our studies. We will also describe the popularity models that were chosen to simulate user behavior in the absence of real video usage traces.

Please note that PLACeS has been released as open source software; it can be freely used and/or modified to support further research in this field.

1.2.2 Analysis of Edge Caching in Next-Gen Optical Networks

The main contribution of our work is an analysis of the bandwidth and energy efficiency of our P2P-based content delivery strategy over a next-generation PON based network, compared to other state-of-the-art alternatives such as Content Delivery Networks (CDNs) and Unicast delivery.

The results of our investigations are presented in **Chapter 4**. We show that P2P is able to greatly reduce core traffic for both Video on Demand (VoD) and time-shifted IPTV services. Furthermore, we present an energy consumption model to evaluate the potential benefit of our content delivery strategy; indeed, a reduction in core traffic translates to a lower number of line cards and network interfaces required in the core, while the increased upstream access traffic generated by peer-to-peer is shown to be essentially cost-free.

1.2.3 Design of a Peer-to-Peer Multimedia Distribution System

Having shown that locality-aware P2P is a valid solution to the dissemination of multimedia content, we devoted our attention to the design of such a system, taking into consideration the challenges that one would face to implement it and proposing solutions to address the issues that might arise.

Locality Oracle Implementation

At the heart of our proposed P2P content distribution system is the so-called *Locality Oracle* (LO). The LO is a service with the double responsibility of tracking the content of each user cache and matching a content request with a local source (when one is available).

While this concept is not entirely new, there are still many considerations to be taken into account when designing such a service, particularly with regards to its scalability and resiliency. The LO can be monolithic and centralized, or distributed and hierarchical; deployed at any one of different layers of the protocol stack, as part of the network architecture, or run as an application on top of the network; it can be deployed as a public API that other applications can take advantage of, or as a network-managed

transparent service that does not require any coordination from the video client. We discuss some of these choices in **Chapter 5**, where we also present a proof-of-concept implementation of a transparent OpenFlow-based locality oracle.

Cache Storage Space Optimization

The system that we describe throughout this work is based on the idea of allowing users to cache whatever content they requested for their personal consumption, and then using this stored content to serve future requests from other customers. Intuitively this works well because of the Pareto-like nature of content popularity, with a small percentage of very popular videos being responsible for a very large percentage of the total traffic. However, given the small capacity of the user caches that might be set aside for network-managed P2P applications and the large number of users in each Access Section, chances are that very popular elements are going to be over-replicated, while content on the long-tail of the popularity curve might be overwritten off the caches - a situation that can potentially lead to unnecessary unicast re-transmissions.

Furthermore, if we consider a scenario in which the storage space is shared among multiple services (e.g., in the case of a next-gen game console which is also used to access VoD services), we might be interested in optimizing the space used for caching even when there is no need to overwrite some previously cached content.

The question then is, can we exploit the knowledge of user request patterns of the LO to guide the caching decision process? In other words, can we be smarter about what we decide to cache, so that the storage occupancy is minimized without sacrificing the percentage of locally-served requests? In **Chapter 6** we describe an optimization algorithm that attempts to do just that.

1.3 Publications

- E. Di Pascale, M. Ruffini, "Cache Storage Optimization for Locality-Aware Peer-to-Peer Multimedia Distribution", *IEEE International Conference on Communications (ICC)*, London, United Kingdom, 2015 (To Appear).

- E. Di Pascale, D. B. Payne, L. Wosinska, and M. Ruffini, "Locality-Aware P2P Multimedia Delivery over Next-Generation Optical Networks", *Journal of Optical Communications and Networking (JOCN)*, Vol. 6, Issue 9, pp. 782-792, 2014.

- E. Di Pascale, F. Slyne, and M. Ruffini, "A Transparent OpenFlow-based Oracle for Locality-Aware Content Distribution", *IEEE 16th International Telecommunications Network Strategy and Planning Symposium (Networks)*, Funchal, Portugal, 2014.

- E. Di Pascale, D. B. Payne, and M. Ruffini, "Impact of Popularity Evolution on P2P-Based VoD Delivery over Next-Generation Optical Access Networks", *IEEE Global Communications Conference, Exhibition and Industrial Forum (GLOBECOM)*, Atlanta, USA, 2013.

- M. Ruffini, E. Di Pascale, and D. B. Payne, "Improving High Fidelity multimedia distribution in next-generation optical networks", *15th International Conference on Transparent Optical Networks (ICTON)*, Cartagena, Spain, 2013. (Invited)

- E. Di Pascale, D. B. Payne, and M. Ruffini, "Bandwidth and Energy Savings of Locality-Aware P2P Content Distribution in Next-Generation PONs", *16th International Conference on Optical Networking Design and Modeling (ONDM)*, Colchester, United Kingdom, 2012.

2 Background and Related Work

In this chapter we will explore the background of our work, detailing the relevant literature and showing how our research fits in this wider context. We start with Section 2.1, where we briefly describe the advances in next generation passive optical networks, with particular regard to Long-Reach PONs, and we delineate the DISCUS architecture which is the reference framework for this study. In Section 2.2 we then give an overview of content caching in telecommunications, the reasons that made it such a popular solution, and the most typical pitfalls and trade-off that one has to take into account when designing a caching system. Finally, we conclude in Section 2.3 with a review of the relevant work in literature on multimedia content delivery and its specific challenges, with particular regards to novel content delivery strategies, energy consumption studies, and the modeling and forecasting of user behavior in multimedia systems.

2.1 Next-Generation Optical Networks

In this section we set the scenario for our work by presenting an overview of Passive Optical Networks (PONs) and their evolution. We introduce basic architectural concepts, briefly describe the existing PON standards, and discuss the current investigation trends for the next generation of optical access technologies.

2.1.1 Network Hierarchies: Core, Metro, Access

A typical telecommunication network can generally be divided into three main sub-components: the core network (also known as the backbone), the metro network, and the access network.

Core networks are used for long-distance transmissions and aggregate traffic coming from the periphery of the network. The requirement for very high capacity and the possibility to share their costs among a large number of end-users typically mean that state-of-the-art optical technologies are deployed at the core, such as 100 Gbps coherent transmission over re-configurable Dense Wavelength Division Multiplexing (DWDM) mesh networks.

Metro networks are at an intermediate level in the hierarchy, providing traffic grooming and multiplexing functions. They are typically implemented as ring topologies interconnecting a number of

Central Offices (COs) or medium businesses sites, using a range of technologies (SONET/SDH, FC, Ethernet etc.).

Finally, connectivity to end-users and small businesses is provided through Access networks. These networks are located very close to the customers and are deployed in large numbers: thus their cost needs to be kept as low as possible. For this reason, network operators try to minimize their investments by taking advantage of existing infrastructures: DSL popularity stems from the ubiquitous presence of twisted copper pairs belonging to the legacy telephone network.

However, as end-users bandwidth requirements steadily increase due to the advent of new services and technologies, copper-based access technologies are rapidly reaching their limits. New technologies such as VDSL promise higher bandwidth, but are limited by their very short range (hundreds of meters) which makes it unfeasible or very costly for network providers operating in sparsely populated areas.

On the other hand, traditional FTTX solutions (Fiber-To-The-X, where the final X can stand for either Home, Curb, Premises etc.) are traditionally very costly, as they require the deployment of a large number of dedicated fiber links whose costs would fall entirely on the few users connected through each of them.

2.1.2 Passive Optical Networks

Passive Optical Networks (PONs) have long been known as a potential solution to these issues. PONs are essentially FTTX architectures in which a large part of the network is shared between several users. In its basic implementation, a feeder fiber is deployed from the Optical Line Terminal (OLT) in the CO to a remote node located close to the end-users; here, a passive splitter is used to connect multiple Optical Network Units (ONUs) to the OLT. Passive splitters can also be cascaded into multiple stages to allow for a more efficient fiber deployment. Each ONU can serve one (in the case of FTTH) or more (FTTC/FTTB) users, allowing for huge cost savings; moreover, the entire path between the OLT and ONUs only employs passive components, reducing OpEx and CO2 emissions.

Split ratios of traditional commercial PON implementations range between 1:16 and 1:64. While a high split ratio is obviously desirable as it allows for better cost-savings, it also directly affects the system power budget and transmission loss. Reach is another key limiting factor of PON networks: current standards support a maximum distance of about 20 km from the CO to the end-users. For these reasons, while the concept of PONs has been around since the 1990s, their actual deployment has only started recently, when the increased requirement for bandwidth and the reduced costs of some key optical components have made it a viable option.

Standard TDM-PON Infrastructures

The two major standards describing PON implementations are ITU-T G.984 series Gigabit-capable PON (GPON) ITU-T (2008) and IEEE 802.3ah Ethernet PON (EPON) IEEE (2012). While differing

on some specific aspects, these standards bear many similarities. In this paragraph we will describe their common traits; their respective peculiarities will be detailed further on.

In both GPON and EPON, an OLT is typically connected to the ONUs via a 1:32 passive splitter, over a maximum distance of 10 to 20 km. Each ONU features one or more ports for voice and client data connections. At the CO side, multiple OLTs are interconnected via a switch or cross-connect which is also responsible for the backbone network connection.

The 1.3 μm and 1.49 μm wavelengths are used respectively for upstream (from ONUs to OLT) and downstream (from OLT to ONUs) traffic. Signals are frame interleaved, with each frame carrying a unique ONU ID in the frame header to identify its intended recipient/sender. Notice that downstream transmissions are inherently broadcast transmissions, as the signal is equally split and relayed to every ONU. On the other hand, upstream transmissions need to rely on a channel access control mechanism in order to avoid collisions on the shared feeder section. In both GPON and EPON, this is achieved through a TDM protocol in which the OLT is the arbiter.

This has more implications than it may be immediately evident. For starters, it requires a proper ranging functionality to be available at the OLT. A typical TDM protocol is implemented through grant messages sent by the scheduler (in this case the OLT) to allocate a time window to a specific user (an ONU), based on its specific bandwidth requirements and a given optimization policy. However, as different ONUs are located at variable distances from the OLT, grant messages will experience different delays before reaching their intended targets. Taking these delays into account is essential to avoid potential collisions due to miscalculated RTTs of the grant messages.

Moreover, in a shared channel architecture such as the PON one, idle transmission patterns can no longer be used to maintain synchronization between the sender and the receiver in between actual data transfers, thus effectively forcing upstream communications to operate in burst mode. A preamble is sent before any actual data is transferred, in order to achieve synchronization and adjust decision thresholds at the OLT (as different distances between ONUs and OLTs also translate to different power levels at the receiver). Additionally, a guard time is enforced between bursts from different ONUs to allow the OLT to recover its initial state before the beginning of a new cycle. Both these techniques introduce additional overhead, limiting bandwidth efficiency.

GPON and EPON comparison

Despite their common basic traits, GPON and EPON differ on many other levels. The most evident is probably the supported bitrates, as shown in Table 2.1.

Another major difference between the two standards lies in the framing of data signals. GPON divides signals into 125 μs frames, then wraps those frames using the so-called GPON Encapsulation Mode (GEM). The purpose of GEM is both to allow for traffic multiplexing (through a 12-bit port ID) and to support payload data fragmentation; it also simplifies the adaptation of different signal formats.

	GPON	**EPON**
Downstream (Mbps)	1244.16 / 2488.32	1000
Upstream (Mbps)	155.52 / 622.08 / 1244.16 / 2488.32	1000

Table 2.1: GPON and EPON supported bitrates.

This mechanism is used both upstream and downstream; virtual upstream frames are composed of data bursts coming from different ONUs. Special pointers are used in the downstream frame header to assign time slots for individual ONUs transmissions, allowing the upstream bandwidth to be controlled with a granularity of 64 kbps. This enables the creation of virtual TDM circuits between OLT and ONUs with guaranteed bandwidth and fine grain control.

EPON on the other hand was engineered to provide native support for Ethernet frames. As such, its data frames have variable length; circuit emulation is thus required to implement fixed-bandwidth TDM circuits. MAC addresses are used to identify both source and destination of the frame. A Multipoint Control Protocol (MPCP) is used to arbitrate upstream access among ONUs; time slots are assigned with a 16 ns granularity. Time-stamps in the frame header are used to estimate RTTs for ranging purposes. Notice that vendors are free to implement their own Dynamic Bandwidth Allocation (DBA) strategy: MPCP only defines the required interface to achieve access control.

XG-PON and 10G-EPON

To keep up the pace with the continuous increase in bandwidth-intensive services and applications, both ITU-T and IEEE have been working on new standards able to deliver higher data-rates. These new systems are called respectively XG-PON (or 10G-PON) and 10G-EPON, and they are essentially a rate-multiplied (10 Gbps) version of their respective predecessors. However, support for such high data rates requires some non-trivial design modifications.

Both standards use the L-band (1575 - 1580 nm) for downstream transmissions and the O-minus band (1260 - 1280 nm) for upstream transmissions, essentially to avoid overlapping with legacy access technologies. Moreover, both employ Forward Error Correcting (FEC) codes to improve the BER and lower the power budget, at the cost of some bandwidth overhead – about 13% for the most typical candidate, a truncated RS(255,223) Reed-Solomon code.

There are many minor differences with respect to the lower rate standards examined earlier on. The preamble in 10G-EPON has been modified to minimize jitter effects and optimize peak detectors. Word-alignment was enforced in XG-PON frames to simplify commercial implementations and reserve space for future extensions. GEM has been revised and expanded to be more flexible and better integrated into the standard. A complete list of all these modifications is however beyond the scope of this section.

2.1.3 Long-Reach Passive Optical Networks

Besides increasing the available bitrate, researchers have been working on increasing the coverage of a PON, i.e., by extending its feeder section and increasing its split ratio; the resulting architectures typically go under the name of Long-Reach PON (LR-PON) or Super-PON Payne (2009). While we have already discussed the advantages in terms of cost-savings of an higher split ratio, achieving long reach would bring the additional benefit of effectively eliminating the need for a metro network. If, for example, the feeder section could be extended as far as 90-100 km from the metro/core node to the Local Exchange, then it would be possible to cover the whole of Ireland with as few as 20 metro/ core nodes (including the additional requirement of dual-homing for protection purposes), thus greatly simplifying network management procedures and drastically reducing Operational Expenditure (OpEx) Ruffini et al. (2012). Similar results have been shown for the U.K., where the node consolidation brought by LR-PONs could reduce the over 5600 current local exchanges to about 75 DISCUS nodes Ruffini et al. (2014).

However, LR-PON still presents several challenges that need to be addressed before they can go into deployment. More specifically, the increased distance between ONUs and the OLT affects various aspects of a traditional PON architecture:

- Longer distances mean higher power loss figures, requiring the adoption of active components (i.e., amplifiers) in the field in order to meet the power budget. Thus, while they are still identified as PONs, these new architectures are not technically completely passive. Besides limiting the cost-effectiveness of these systems, amplification introduces further complications. For starters, special care needs to be taken in order to keep Amplified Spontaneous Emission (ASE) effects low while maintaining a high Signal-to-Noise Ratio (SNR). Furthermore, power level variations due to the different distances of ONUs from the OLT make traditional Erbium-Doped Fiber Amplifiers (EDFAs) less attractive, due to their relatively slow speed in adjusting the gain to compensate these effects. Various solutions - ranging from the implementation of a separate wavelength gain control system to the adoption of Semiconductor Optical Amplifiers (SOAs) - are being considered to tackle this problem.

- The low-cost, uncooled transmitters employed in traditional PON ONUs are not designed to be used across the high distances of an LR-PON. Moreover, if WDM is used to accommodate a larger user base, the temperature-induced wavelength drift of these cheap devices suddenly becomes a major issue.

- Similarly, a better burst-mode receiver is required at the OLT to accommodate for different propagation attenuation levels of the signals coming from the ONUs. The broader range of DC levels introduced by the amplifiers, the larger attenuation introduced by high split ratios and long distances, and the strict timing control required to achieve bandwidth efficiency in TDM schemes

all contribute to increasing the complexity of the problem.

- Over such long distances, control-plane transmission delays become significant, introducing the need for efficient remote-scheduling solutions to address upstream bandwidth allocation.
- Finally, an increase in the number of users covered by a single LR-PON (be it through longer distance feeder sections, higher split ratios or both) translates into stricter protection requirements to avoid potential faults that could affect simultaneously thousands of customers.

Despite all of these challenges, LR-PONs represent a very attractive approach to next-generation access networking. Most of the issues presented above are only related to cost efficiency and power budget limitations rather than being real show-stoppers. A huge effort is being put in the scientific community to tackle these issues; it is reasonable to expect researchers to come up with practical solutions in the near future. For a survey of existing LR-PON demonstrations, please refer to Song et al. (2010). Furthermore, a survey on the standardization trends of next-generation optical access systems, including LR-PONs, can be found in Effenberger et al. (2010).

2.1.4 The DISCUS Architecture

DISCUS (Ruffini et al. 2014) is an FP7-funded research project investigating end-to-end solutions for ubiquitous broadband optical access. The project is coordinated by CTVR through its two PIs – Prof. David B. Payne and Prof. Marco Ruffini – and as such it represents the architectural vision of the center for next-generation networking. While my work is not strictly part of DISCUS, it shares similar motivations and follows almost identical assumptions. As such, DISCUS can be seen as the reference framework for most of my studies.

One of the main assumptions from DISCUS is that, in order to reduce costs and energy consumption, network resources should be shared among multiple users. While this is the norm in the core and metro sections of the network, it usually isn't the case for the access, where the low capacity of the existing copper infrastructure led to point-to-point architectural choices. However, technologies like the LR-PON allow sharing a single high-capacity fiber access among as many as 1024 users, while at the same time reducing the amount of electronic components typically required in a DSL-based infrastructure.

As mentioned in the previous sections, the longer optical reach of these fibers allows us to reduce the number of termination points required, thus aggregating a large number of small Central Offices (COs) into a handful of so-called metro/core (MC) nodes. It is then possible to connect these MC nodes into fully transparent optical islands, so as to minimize the number of IP switches and electronic converters required. This helps reducing both the costs and the energy footprint of the considered network.

In short, the DISCUS architecture is based on LR-PON in the access and a collection of flat optical islands in the core (see Fig. 2.1). However, besides investigating the technical challenges of this proposal, DISCUS aims to evaluate its economic feasibility, to identify the regulatory framework that would

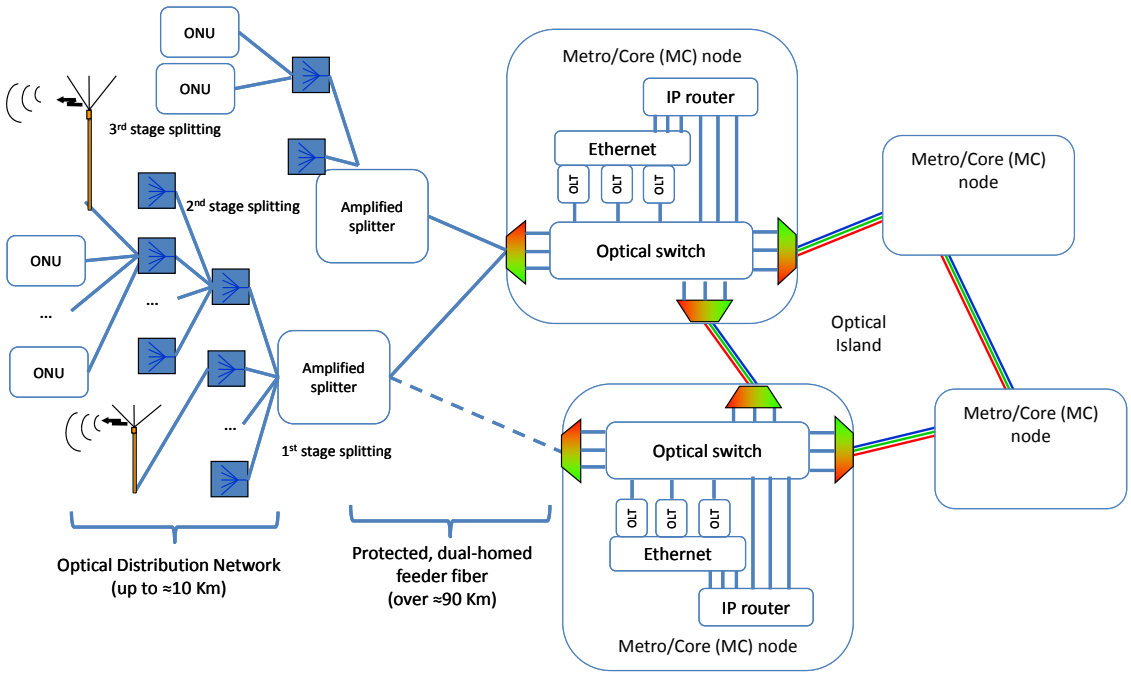

Figure 2.1: Diagram of the DISCUS architecture.

benefit such an architecture, and to model the services and usage patterns that the network would need to support.

The target scenario of our studies borrows heavily from the DISCUS architectural view, assuming a next-generation network deployment based on LR-PON trees with up to 512-1024 users terminating in a small set of MC nodes. However, in our model these MC nodes are typically interconnected through IP or OTN links, rather than transparently through all-optical switches; in other words, we forego the flat optical core, as this aspect is still being investigated in order to assess its feasibility and the dimensions to which it would be able to scale.

2.2 Overview of Network Caching

In the previous section we briefly described the network architectures supporting the multimedia services we are interested in studying. Indeed, the focus of this work is essentially on multimedia content caching and distribution, rather than on the technical aspects of future optical access networks. In the following paragraphs, we present an overview of network caching and the challenges typically associated with it, before delving into the literature specifically targeting multimedia caching over next generation networks.

In brief, caching consists in storing replicated copies of data that is assumed to be of interests to end users, so that future requests for it can be served more efficiently. Typically, caches are located closer to the users with respect to the origin of the data; the aim of caching can then be either to improve user experience (i.e., by reducing delays), to reduce bandwidth usage (i.e., by having the data traverse

a smaller number of links to reach its destination), to reduce the load on the origin server (i.e., by distributing it among a number of cache servers), or a combination of the three.

Whenever a new request is received from a user, the system checks whether the requested content can be found in the cache; if that's the case (*cache hit*), the system proceeds to serve the content directly from the cache. Otherwise, we say there was a *cache miss*, and the content has to be fetched from the origin server. When this happens, typically the newly requested content is added to the cache, in order to intercept future requests from other users.

2.2.1 Eviction Policies

Naturally, caching content implies a certain degree of redundancy, as the same data is copied and stored in multiple locations. There is an obvious trade-off between minimizing the level of redundancy required and increasing the *hit rate* of a cache – that is, the probability of having a cache hit on a new request. Replicating the entire catalog available on the origin server would ensure a 100% hit rate, but it would also be highly wasteful in terms of storage resources. Luckily, in most practical cases that is not required, as there is usually a modest portion of the catalog that is responsible for the vast majority of the requests (see Subsection 2.2.3). It is then possible to design caches that are much smaller than the origin server, and hence can only store a subset of its original catalog, while still being able to greatly reduce cache misses.

The problem then becomes determining what to store in the cache. Unless the system designer has some previous information on the future popularity of content, typically caches are instructed to store every object for which they observe a request – in other words, at every cache miss the requested content is added to the cache. Whenever there is not enough available space to do that, one (or more) of the previously cached elements is erased to make room for the new entry. The way in which the element(s) that need to be deleted are selected is defined by the *eviction policy* of the cache.

Since the objective of the cache is to maximize its hit rate, most eviction policies try to estimate the popularity of the elements currently stored in the cache using some simple metric, so that the items that are less likely to be requested again can be erased. Two of the simplest and most popular eviction policies are Least Recently Used (LRU) and Least Frequently Used (LFU).

With LRU, each object in the cache has an associated time-stamp relative to its latest request. The basic assumption of this policy is that an item that has not been requested in a long time might be no longer relevant or popular, and as such constitutes a good candidate for eviction. The advantage of this strategy is that it is simple to implement and, for most applications, it works quite well; however, particularly in multimedia caching systems, this is not always an optimal policy. For example, it is not uncommon for an "old" element of the catalog to see a sudden increase in popularity – e.g., in the case of old episodes of a TV series immediately before the release of a new one. More in general, if the cache size is too limited, a number of requests for unpopular contents could erase from the cache several

popular elements with an older time-stamp. The main drawback of LRU, in other words, is that it is short-sighted, in that it only stores information on the amount of time elapsed since the last access – with no regards for the actual rate of request of a video element.

LFU tries to address this issue by keeping a history of the frequency with which an element has been requested. In the last example shown, an element that was recently accessed but received only a few requests would be prioritized for eviction over some other object which was requested several times, albeit some time ago. This policy seems to better fit our intuitive idea of what popularity means, but it is not exempt from drawbacks: since the only relevant metric is the number of times an object has been accessed, items that were popular in the past will be kept in the cache long after their popularity has faded, often at the expense of young elements with higher request rates. For this reason, modified versions of this algorithm have been proposed where the weight of the requests received decreases over time to account for aging content (LFU-Aging).

The two simple algorithms presented above are just a small sample of the vast variety of strategies that have been devised over time to improve the efficiency of caches under specific assumptions. For example, in some applications the difference in popularity between the elements of the catalog could be marginal compared to their difference in size, and a policy that prioritizes small objects over larger ones might be more beneficial. Other policies attempt to use previous knowledge of the popularity distributions of the element in the catalog to make educated guesses on future request patterns – indeed, that is the principle behind the caching optimization scheme presented in Chapter 6. An exhaustive coverage of all the caching eviction policies in literature is beyond the scope of this work; the interested reader can refer to the survey by Podlipnig & Böszörmenyi (2003). We will however return to this topic when describing the specific issues of popularity estimation for multimedia caching in Section 2.3.3.

2.2.2 Freshness and Consistency

Storing multiple copies of the same content in different locations of the network becomes particularly problematic when the data is subject to changes and updates; this introduces the additional problem of ensuring that the cache data is *fresh*, or, in other words, up-to-date. A typical example where this problem arises is the caching of dynamic web-pages, like the content of an on-line newspaper; if we decide to add today's front-page to the cache, there is only a limited window of time in which that content is going to be relevant, after which we will need to fetch the updated page from the origin regardless of whether we have a local copy available.

The problem becomes even more complicated if the objects that we are caching can be modified by any of the distributed clients that hold a local copy. Without taking specific precautions, chances are that the *consistency* of the caches will be compromised, with different users seeing different versions of the same content. To prevent this, cache designers implement *coherence protocols* that ensure that any modification to a cached object is propagated throughout the system, e.g., by signaling the other caches

to invalidate the outdated content so that at the next request it will be fetched from the up-to-date source.

While both of these issues are significant in a generic cache system, none of them really applies to the focus of our studies, that is, multimedia caching systems. Unlike web pages, videos are usually not modified over their life span, neither by the publisher (freshness) nor by the consumer (consistency). The only real issue with multimedia elements is their expiration and consequent retirement from the catalog, either due to a decrease in relevance or because of limitations in the copyright license of the publisher. When this happens, however, it is sufficient to instruct all caches to erase any residual copy of the expired content.

2.2.3 Web-Caching and Multimedia Caching

As shown by the last example in the previous section, different kinds of content often have distinctive traits that must be taken into account when designing the caching system. This is particularly true for the popularity distribution of elements in the catalog.

Historically, network caching was introduced to support the World Wide Web and Internet browsing. Researchers quickly found that the popularity of web pages could be fitted to a Zipf distribution – in other words, on a log-log graphs mapping the number of average requests for each content based on its popularity rank, the resulting curve is close to a straight line. Mathematically, if we rank the items in our catalog sequentially from 1 to N, where lower ranks imply higher popularity, an item with rank i will be selected with probability

$$P(i) = \frac{1/i^e}{\sum_{n=1}^{N} 1/n^e} \tag{2.1}$$

where e is the *exponent parameter* of the distribution.

In layman's terms, this means that, depending on the slope of the popularity curve, a very small percentage of the content of the catalog is responsible for a very large portion of the user requests. The 20-80 rule, which is often quoted in literature, states that the most popular 20% content will account for about 80% of the total requests. When this happens, we say that the popularity curve is *Pareto-like*, from the name of the Italian mathematician who first observed this phenomenon.

However, as shown first by Gummadi et al. (2003) and later re-affirmed by Yu et al. (2006), the popularity of multimedia elements is not correctly modeled by a pure Zipf distribution. Studies on several traces from real-world deployment of VoD services usually show a lower number of requests for the most popular elements and a longer tail of low-popularity items compared to what one would have in a purely Pareto-like distribution. Gummadi et al. Gummadi et al. (2003) speculate that this might be due to the "fetch-at-most-once" nature of video on-demand catalogs, where most users will only access a given element once regardless of its popularity; in other words, once a movie has been watched, it is unlikely to be requested by the same user any time soon. This contrasts with the "fetch-repeatedly"

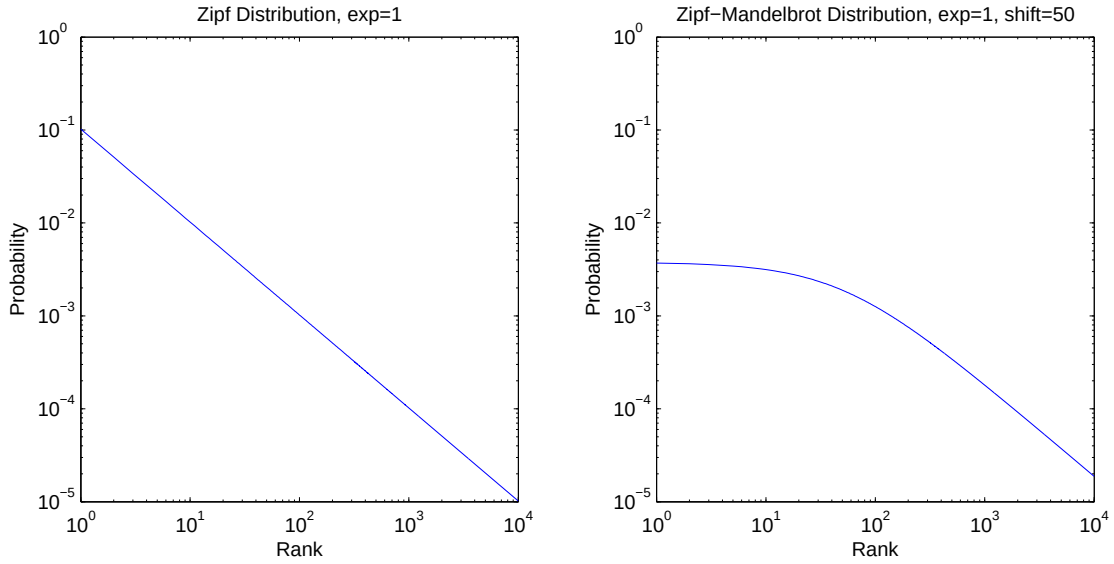

Figure 2.2: Probability mass function graphs of a Zipf distribution with unitary exponent (left) and a Zipf-Mandelbrot distribution with the same exponent and a shift parameter of 50 (right), for a catalog of 10^4 items.

nature of web pages, where popular websites might be accessed several times in a day by the same users.

As a result of this, some researchers suggested that the popularity of multimedia content is better represented by a Zipf-Mandelbrot distribution – a generalized version of the Zipf distribution with an additional *shape parameter*, whose effect is to reduce the slope of the curve for the lowest ranks. More specifically, according to a Zipf-Mandelbrot distribution of exponent e and shift s, an element of rank i will be selected with probability

$$P(i) = \frac{1/(i+s)^e}{\sum_{n=1}^{N} 1/(n+s)^e} \tag{2.2}$$

Both a Zipf and a Zipf-Mandelbrot distribution are shown in Fig. 2.2, with an exponent of 1 and, for the Zipf-Mandelbrot case, a shift parameter of 50.

As a consequence of this reduced relevance of the most popular items, multimedia caches tend to under-perform compared to web caches with the same level of replication (e.g., some percentage of the catalog size). It is important to keep this in mind when designing a cache system for multimedia content.

Another important difference regards the granularity of caching. Web objects tend to have a limited size and as such are usually cached as a whole. Multimedia items, on the other hand, can span several Gigabytes in the case of high definition movies. The storage and transmission of such huge objects can often lead to inefficiencies, especially when users are only interested in a small portion of the object – indeed many videos are not watched in their entirety. For these reasons, many multimedia providers divide their content into multiple *chunks*, i.e., fixed-size portions which are easier to handle and transmit. The size of these chunks is a sensitive parameter of the system: very large chunks do little in terms of solving the issues outlined above, while very small chunks introduce a significant overhead in terms of management, indexing and signaling.

Throughout our work, in order to simplify our model we do not consider the issue of chunking; in other words, we will assume that videos are transmitted and cached in their entirety. For an analysis of the consequences of introducing chunking in our model, please refer to Section 7.2.1.

2.3 Related Work on Multimedia Caching and Delivery

After an overview of the basic concepts behind our thesis, we can now devote our attention to the works already present in literature. The most relevant papers will be reviewed and divided into three main sections. We first examine the various strategies and architectures proposed to optimize the network delivery of multimedia content in Section 2.3.1. In Section 2.3.2 we sample the literature dealing with the problem of minimizing the energy consumption of multimedia content delivery. Finally, in Section 2.3.3 we cover the publications focusing on the estimation of multimedia content popularity for the purposes of caching.

2.3.1 Novel Architectures for Multimedia Content Delivery

Following the widespread popularity of video services over the Internet, the problem of how to efficiently deliver multimedia content to users has been the subject of a very large body of work. We limit our attention to a sample of the relevant papers published in recent years.

Live-Streaming

Many of the studies in literature focus on live streaming services, i.e., applications where the user is interested in watching a live program such as a sport event. The peculiar nature of live streaming introduces both simplifying assumptions and additional constraints. On one hand, all users are synchronized, i.e., they are requesting (roughly) the same portion of the content at the same time; this obviously simplifies the delivery of content, as the same piece of data is valuable to all of the users. On the other hand, the live nature of these events means that much stricter timing constraints are in place: re-buffering events do not simply translate into momentary pauses, but rather in portions of the stream being missed in order to catch up with the live progression of the video.

Due to the synchronous nature of the streams, most service providers (SPs) select multicast as the preferred delivery strategy. However, P2P is often used either in conjunction with or in substitution of multicast, e.g., to reduce the load on the video servers, to enable time-shifting functionalities, or simply to reduce costs for new SPs. Here we will list some relevant studies that investigate the usage of P2P in live streaming services.

Cha et al. (2008) propose a P2P-assisted multicast delivery for a telco-managed live IPTV service with rewind functionality, i.e., allowing users to go back to the beginning of an ongoing stream regardless

of their joining time. The traces used in the study were taken from an operator using IP multicast over a DSL-based broadband network. P2P is only used for patching, i.e., to distribute content that was already broadcasted before the joining time of a user. Meanwhile, the in-sync portion of the stream distributed through multicast is cached for future use, to minimize the impact of P2P. A similar architecture is described by Gallo et al. (2009).

Carta et al. (2010) focus on the signaling protocol required to trade chunks between peers in a purely P2P-based live streaming service. They investigate optimal strategies to minimize queues, reduce delays, and avoid losses or re-buffering events.

The work by Jin & Kwok (2011) analyzes the impact of both locality awareness and capacity awareness (i.e., prioritizing peers with higher upload capacities) on the effectiveness of a P2P-based live media streaming service, both in terms of user experience and ISP costs. The authors show that both strategies, when not properly mitigated, can hinder efficient chunk dissemination due to clustering effects between peers. They propose a hybrid neighbor selection strategy to overcome these issues, and they introduce a Decentralized Network Awareness (DNA) protocol to improve the overlay construction process.

Wu et al. (2011) propose Ration, an algorithm to forecast the amount of bandwidth that a media server in an ISP network is required to allocate in order to support a live P2P-based streaming channel. Using an array of dynamic learning techniques (e.g., time series forecasting and dynamic regression), Ration is reported to be able to dynamically predict the demand in each channel and to proactively provision optimal server capacities across different channels. Furthermore, since Ration is carried out on a per-ISP basis, it has full ISP-awareness and is thus able to guide the deployment of server capacities so to constrain P2P traffic inside ISP boundaries as much as possible.

Bikfalvi et al. (2011) propose a live streaming IPTV architecture using multicast for popular channels and unicast P2P flows for less popular channels. They argue that while dynamic IP Multicast is in general more efficient in terms of bandwidth, it carries an overhead in terms of scalability (linked to the number of forwarding entries to be stored at each core network router) and signaling. They show that, for unpopular content, the bandwidth overhead of P2P is minimal and thus it represents a valid option to make the system more scalable.

The paper by Xu Cheng et al. (2012) deals with P2P distribution of User Generated Content (UGC) over social network services. Only the flash crowd scenario is addressed, since it's considered the most problematic. The authors distinguish between so-called storage users, who do not wish to watch immediately the content they are downloading (e.g., because they're waiting for it to reach a certain threshold or maybe downloading it for a later time), and users who are streaming live content. Storage users are less likely to leave the overlay and thus they are prioritized in the distribution tree (i.e., they are placed closer to the source), with the exception of the initial phase of the flash crowd, where streamers are prioritized through a separate tree (which is later merged with the main one) to ensure their perceived start-up delay is low.

The work by Mu Mu et al. (2012) describes the Lancaster Living Lab, a walled garden testbed for P2P-based IPTV services, whose users are either campus students (circa 6000 households) or people living in a nearby village cabled with fiber links (circa 300 families). Set top boxes are used to disseminate content, both for live streaming and catch-up TV up to 30 days after the original broadcast. QoE measurements are taken thanks to interactive feedback from the users mediated by the STBs. QoS and various other network statistics can also be collected.

Please note that in our work, we focus on on-demand multimedia delivery and we do not take live streaming into consideration. The main reason behind this choice is that in our architecture of reference, based on PON trees directly connecting users to the core, multicast is an obvious choice for live streaming scenarios, due to the broadcast nature of the PON downstream channel. The usual lack of support in core networks for IP multicast is not a problem either, since we are envisioning an ISP-managed IPTV service and it would be in the operator's own interest to put an infrastructure in place to support it; furthermore, the flat-optical core of DISCUS would remove this issue by allowing us to transparently bypass IP routers while inside an all-optical island. However, the problem of supporting time-shifted streaming of live TV channels over our architecture might be an interesting topic for further investigation, e.g., analyzing how to optimize the P2P overlay building process, and how to make use of the broadcast feature of our PON architecture to improve the efficiency of patching for multiple users.

Content Delivery Networks for Multimedia

Content Delivery Networks (CDNs) are arguably the de-facto default solution for content dissemination over the Internet. There are a number of publications investigating strategies to optimize the specific scenario of on-demand multimedia caching. In many cases, these works discuss a massively distributed CDN, which can be easily mapped to our P2P case by assuming that the STBs are the basic component of the content delivery infrastructure.

The paper by Borst et al. (2010) attempts to determine an optimal replacement policy for objects in a distributed cache, with the aim of minimizing bandwidth usage in the network. The authors consider both a two-tier cache architecture – with the origin server acting as a root, a single parent representing a larger server closer to the core, and many smaller leaves – and a similar architecture where only the leaf nodes are allowed to cache content. The paper shows that under certain assumptions (namely uniformity of cache sizes, content request patterns and bandwidth constraints across users) a distributed greedy algorithm that uses popularity estimations to decide whether to replace a cached content with a newly requested one is a 4/3 approximation of the optimal algorithm. Furthermore, even when some of these assumptions are relaxed, the same greedy algorithm is still a 2-approximation of the optimum. Naturally, the algorithm requires knowledge of the content of other caches and some sort of estimation of the popularity of each content. The authors acknowledge that when dynamics in popularity are present a larger gap from the optimal is likely to be obtained as a result of errors in the estimation of future requests.

Applegate et al. (2010) define a Mixed Integer Problem attempting to optimize the placement of videos in caches located in each metro node. They initially assume that each content item must be replicated in its entirety (hence the integer nature of the problem), but this constraint is later relaxed to convert the problem into a Linear Programming one. This is then solved through heuristics whose solutions are proven to be within 1% of the optimum. Link capacity constraints of 1 Gbps are only enforced at a selected number of peak-time slices. User requests are modeled using traces from a real VoD service deployment. Video elements are assumed to be encoded at 2 Mbps and to have one of four possible lengths, i.e., 5m, 30m, 1h, or 2h. Popularity estimation is done comparing new videos with similar previous items (e.g., the previous episode of a TV series). 5% of each cache is reserved to be used with a simple LRU policy to mitigate the effects of erroneous popularity predictions. Experiments are run on a custom-made simulator. Results show an improvement in terms of total bytes transferred and peak link bandwidth utilization with respect to LRU, LFU and a popularity-proportional top-100 placing. However, because the link capacity constraints are enforced only at specific time windows of 1h each, the expected bandwidth utilization is reported to be about 30% higher than the actual available capacity. This is a relevant issue in our opinion, since exceeding the available capacity in a real deployment could generate re-buffering events and generally degrade the Quality of Experience perceived by the users.

Spagna et al. (2013) examine the design challenges of a telco-owned highly distributed CDN. The focus in on mobile network operators, but most principles apply to our fixed-network scenario too. It is also interesting that the authors use the ALTO protocol to implement locality-awareness in their CDN infrastructure. Their solution introduces a hierarchical caching architecture where multiple local data-center clusters are coordinated through global CDN Control Function elements. Requests are addressed in a bottom-up fashion by attempting to find a local cached copy and pushing the request to the upper layers only in case of a miss. Content is divided into three classes based on the stability of its popularity dynamics and the geographical extent of its intended audience; content expected to be popular can be pre-fetched, while more volatile content is only cached on-demand. They suggest that caching at edge core router level is the optimal solution in terms of balancing costs and traffic reductions; note however that in the mobile case, caching a the end-user location is not a viable solution, unlike in our fixed-network scenario.

The solution presented by Abeywickrama & Wong (2013) is tailored for PON networks, specifically GPON or equivalent (i.e., 32-64 ONUs per PON, 20Km distance between OLT and remote node). It revolves around a Local Server (LS) placed in the remote node, where the OLT is located. They propose a modification to the Dynamic Bandwidth Assignment (DBA) algorithm so that the LS can keep track of video requests coming from the ONUs and start caching popular content directly from the downstream channel. Their architecture requires an additional 2xN splitter at each OLT to allow the LS to “eavesdrop” and intercept VoD requests; furthermore, an additional wavelength is required to push the cached content to ONUs, which then need to be fitted with an additional receiver. The

authors analyze the QoS of the proposed system and show that it leads to an increase in jitter of up to $\sim 30ms$, which they claim is acceptable. They also perform a power consumption analysis using the model by Baliga, Ayre, Hinton & Tucker (2009), and show an average increase of 10.4% in power consumption per customer.

Chan & Xu (2013) present a distributed CDN architecture for VoD. They show that the problem of jointly optimizing content placement and retrieval to minimize costs for the Service Provider is NP-Hard. They propose to decompose it in two linear problems (for placement and retrieval respectively), and show that this solution converges asymptotically to the optimum as the number of chunks in which the content is divided approaches infinity. In reality, the authors show that a modest number of chunks (in the order of magnitude of 5-10) is sufficient to remain within 7% of the optimum in their case studies.

Zhe Li & Simon (2013) look at the case of an ISP wanting to implement its own CDN platform, based on metro-regional repositories, and coexisting with traditional inter-ISP CDNs. Their approach consists in determining the optimal placement of content in the ISP-controlled repositories based on the videos individual users are expected to request (i.e., based on the recommendations to the users made by the VoD service). The objective is to maximize the hit ratio while at the same time minimizing some network cost function. The linear problem is NP-complete, and it's solved with a meta-heuristic (genetic algorithm) implemented and solved through MapReduce on a small cluster. Their topology is based on the French backbone topology, and they evaluate their results using real traces from the Orange VoD service. The paper has, however, a number of limitations. Firstly, the algorithm relies heavily on the prediction of user requests. Their results require an extensive warm-up period (7 days) after which their scenario only lasts between 1 and 7 days, with the latter being problematic because the estimated popularity no longer matches the real request patterns. Furthermore, computation of the optimal placement takes 12 hours on a cluster of 10 machines and considering only a limited data set. The authors claim that some of the inconveniences of their approach could be reduced or eliminated by running the replacement algorithm daily, but given its computational complexity this seems unfeasible. Finally, the results section mostly focuses on the comparison between LRU caching and *perf-opt*, which is an upper bound of what the algorithm can really achieve based on perfect knowledge of the future, while very limited results are presented for the realistic case.

Similarly, Claeys et al. (2014) describe a scenario in which an ISP leases caching storage space located at the edge of its network to third-party content providers. Their algorithm attempts to jointly optimize content placement and routing with the objective of minimizing the total bandwidth used. However, content popularity is assumed to be known a priori and it is expressed as request rates for each element from each edge network node.

Fratini et al. (2014) study the performance of VoD content placement at various levels of the metro/access network hierarchy. They propose an integrated metro/access architecture where, regardless of the network technology employed in the access (e.g., DSL or PON), there is a long-reach branch-tree structure connecting customers in the access to a root point in the core. The nodes in these trees can

then be connected with peer-nodes from other access trees in a mesh structure that allows the cache servers to increase their potential pool of users. The authors also distinguish between active and passive architectures, based on the technology employed in the metro/access segment and consequently on the ability to switch flows without reaching the root of the tree. Their results show that the additional mesh links between peering nodes are only useful for active architectures. More in general, active networks are shown to perform better than passive ones in terms of bandwidth efficiency, but they are also more costly and less energy-efficient.

Information-Centric Networking

Information-Centric Networking (ICN), also referred to as Content-Centric Networking (CCN), is a branch of telecommunications that advocates a re-design of computer networks to acknowledge the fact that we are not usually interested in connecting to a specific machine, as postulated by the current dominant network protocols, but rather in retrieving some information. Under this assumption, a typical user does not really care *where* the information is coming from, as long as its retrieval is as quick and efficient as possible. Users should then be allowed to request content using some naming mechanism (rather than providing the address of the machine holding it), and the network should use its knowledge of the locations in which that content is replicated to pick the best source among those available. Very often, ICN proposals support the idea of in-network caching, turning routers into a distributed network of caches that are able to hold replicated copies of the content they are forwarding. A good survey of the most relevant ICN architectures can be found in (Ahlgren et al. 2012).

ICN has received much attention and as such it's no longer a uniform field but rather a complex galaxy encompassing different viewpoints. For example, some authors argue that a "clean slate" redesign of the networking protocol stack is required to overcome the limitations of the current IP-centric model; on the other hand, many researchers believe that these approaches are either unfeasible (due to economic reasons) or even unnecessary, as similar benefits can be achieved by modifying and evolving the current network architectures. The work by Ghodsi et al. (2011) does a great job of surveying the major differences and commonalities in the various ICN designs proposed, showing the downsides of clean slate approaches, and evidencing the areas that need further investigation, such as verifying the actual benefits of cooperative caching.

Cho et al. (2011) detail a telco-managed infrastructure using modified routers to store content. Since it's ISP-specific, this strategy does not require a clean slate approach for the whole Internet and it's also designed to be backward compatible. Naming is performed trough a unique URL associated to each content. Items are downloaded through a BitTorrent-like swarm where the seeds are the multiple routers holding the content. Simulations compare the performance of their solution with CDN, unicast and locality-aware P2P – however they strive for 10 concurrent uploading peers, and there are only 100 routers in the whole network (with 1 transit domain and 5 stub domains), so the chances of finding

enough local peers to satisfy a request are low. By considering symmetric bandwidth in the access, we move to a model where a single peer is sufficient, thus overcoming these limitations.

In (Cho et al. 2012) the same authors present a popularity-based decentralized ICN caching scheme, where the upstream node offers recommendations to the downstream one on the number of chunks of a given content that should be cached based on the request rate observed. Note that the statistics on request patterns that are saved at each router are related to the whole file, not to the individual chunk, in order to reduce storage overhead. LRU is used to evict chunks when necessary. In the results presented, this strategy beats other router caching policies in minimizing inter-ISP traffic, but it still out-performed by a traditional CDN solution.

The paper by García et al. (2011) describes the vision of the COMET project, whose main contribution is the introduction of a content mediation plane which is responsible for name resolution and path preparation. At the core of it there's a hierarchical pub/sub system where each block has full knowledge of what has been published in its area of control. The system is agnostic to the specific location of the caches (e.g., routers/servers/STBs); the emphasis is on the global name resolution system and the path selection mechanism, which takes as input information on the state of the network.

Li & Simon (2011) propose a cooperative caching strategy for content-storing routers to reduce inter-AS traffic. According to their label-based policy, content routers are only allowed to store chunks whose progressive ID is a multiple of the label assigned to the router. This simple way of associating content chunks to routers also allows them to easily identify viable neighboring sources for chunks when required.

The literature on ICN is vast and an extensive review of it would be outside the scope of this thesis. Undoubtedly, there are many common aspects between ICN and the vision supported by our work: locality-awareness, a better integration between the networking layer and the application layer, and the recognition that "content is key" are staples of both approaches. However there are some aspects of ICN solutions that do not fully convince us, such as the idea of caching content pervasively in routers, or the necessity to re-invent existing protocols to accommodate this new networking paradigm. Our perplexities are well captured and described by Fayazbakhsh et al. (2013); their paper shows that the performance of ICN in an optimistic best-case scenario is still within 6% of an edge caching solution for all the metrics of interest (e.g., response time, network congestion, origin server load etc.).

Peer-to-Peer Based Architectures

Peer-to-Peer (P2P) is at the core of our proposed solution. The idea of using P2P to disseminate multimedia content, however, is certainly not novel in itself; in this section we will examine other works that have addressed this topic in recent years.

Huang et al. (2007) were among the first to analyze P2P-assisted VoD. They use traces taken from a real VoD deployment, namely MSN videos. The paper focuses on a model where peers can only upload

pieces of the video they are currently watching. The VoD server only intervenes when there is not enough upload bandwidth to satisfy the demand. They consider pre-fetching by allowing customers to use surplus bandwidth to fetch future chunks of the video currently being watched. This is shown to be beneficial, for intuitive reasons.

Several papers have been published on the optimization or adaptation of the BitTorrent protocol for the delivery of on-demand multimedia content. Janardhan & Schulzrinne (2007) present a high level description of a P2P-assisted VoD service architecture taking advantage of the storage of STBs, using a BitTorrent-like protocol. The VoD server is expected to seed or possibly pre-fetch content expected to be popular. A large part of the paper describes design consideration regarding various components of the system (e.g., the sliding window module, chunking, where to place anchor points in a video etc.). There is no locality-awareness – peers use DHTs instead to fetch content of interest. No experimental results are presented, simulated or otherwise.

AntFarm, proposed by Peterson & Sirer (2009), is also a pure P2P architecture based on BitTorrent. The authors introduce a coordinator, i.e., a modified BitTorrent tracker whose job is to optimize how much bandwidth is assigned to each swarm by the seeds, so that the global average latency for all downloads is minimized. This is achieved through a constrained optimization problem, where the coordinator uses the slope of the response curve (i.e., the plot of the aggregate download bandwidth as a function of the assigned seed bandwidth) to estimate the residual utility of the bandwidth assigned to each swarm. Intuitively, once the peers in a swarm have their upload capacity saturated, additional bandwidth only benefits individual users receiving it, rather than the whole swarm.

The work by Yang et al. (2010), which describes a BitTorrent-like P2P protocol for both live streaming and VoD, attempts to improve the chunk prioritization policy for video streaming. Indeed BitTorrent was designed to optimize the throughput of file transfers, and thus it tries to prioritize the dissemination of rare chunks (so that more replicas are available in the network) and to encourage fairness in sharing through a so-called tit-for-tat incentive mechanism. The chunks of a multimedia file, however, are usually accessed sequentially, imposing a constraint on the order in which they are required; this naturally clashes with the rarest-first approach of BitTorrent. Furthermore, the incentives to fairness are not ideal for live streaming environments due to the fact that new peers have no content of interest to be shared with the early joiners of the swarm.

Similarly, Parvez et al. (2012) develop an analytic model to characterize the behavior of BitTorrent-like protocols for VoD streaming. They focus on the efficiency of various chunk selection schemes, including rarest-first (the default BitTorrent policy for file sharing), in-order, and probabilistic policies. They show that the rate at which the stream progresses is a function of both the sequential progress (i.e., the usefulness of the obtained pieces for sequential playback) and the overall download progress. For this reason, they argue that a naive in-order policy, which strongly favors the former while hampering the latter, is not an optimal choice, while a probabilistic policy can achieve better results by striving for the best of both words.

In our opinion, however, a purely BitTorrent-based system is always going to be sub-optimal for a VoD service like ours for all the reasons already mentioned, such as the out-of-order retrieval of chunks, the lack of integration with the network infrastructure etc. Furthermore, much of the success of BitTorrent is due to its tit-for-tat fairness mechanism, which would not be strictly required in a network-managed scenario like the one we are considering.

Considerable research has already been done on the use of STBs as the basic units of a distributed P2P cache. More specifically, in ECHOS (Laoutaris & Rodriguez 2008) the authors sketch the basic idea that is behind our own work – pushing content at the edge of the network by turning STBs in nano-datacenters, as the authors call them. However, no analytic study or simulation of the performance of such a system is shown.

In Zebroid, a work by Chen et al. (2009), not only multimedia content is divided into chunks, but those chunks are then striped with erasure codes so that a subset of them is sufficient to reconstruct the original object (*Wikipedia - Erasure Code* n.d.). This is done to circumvent the problem of set top boxes shutting down; based on observed measurements, in their simulations 20% of the STBs on average might become unavailable, while the rest are considered to be always on. Stripes are sent to set-top boxes during off peak hours, in a typical pre-fetching scheme. The VoD servers only kick in when there's not enough bandwidth from the peers. Their reference architecture uses FTTN from the core and DSL for the last mile. In their paper, a community of users (the equivalent of our Access Section) is the set of customers attached to the same DSLAM switch. Experiments are based on a testbed of 64 VM or on simulations of 300 nodes in a community with 500 movies. A discussion of the applicability of the Zebroid approach to our study can be found in Sec. 3.2.5.

In the work by Han et al. (2011) content is pre-fetched via multicast to STBs during off-peak hours, based on estimated future request rates. The peculiarity of their proposal is the use of alternative connection routes to distribute this content among neighbors (e.g., home wireless networks, power line infrastructures etc.), which however introduce security and access concerns. STBs are assumed to be always on but with a non-negligible chance to fail; for this reason content is purposely over-provisioned.

The work by Chellouche et al. (2012) is part of the ALICANTE project, and it has many similarities with our study. The authors design a hybrid P2P+CDN system for VoD, where the term is used loosely to include User-Generated Content (UGC). The P2P part is implemented over home-boxes (HB) which are equivalent to our network-managed set-top boxes. There is a Service Manager (SM) which acts as our Locality Oracle: it keeps track of content popularity, redirects requests to caches, and enforces the replication strategy. Furthermore, there is a Service Register (SR) which keeps track of the content available in the catalog and of who is caching it. Users are served by a HB if possible, or by the CDN if not (either because the content wasn't cached or because the QoS could not be guaranteed by the HB). Like us, they rely on user requests to populate the caches, rather than enforcing a content placement allocation. Caching in HBs however is only allowed for the most popular M_p contents. The optimum value of M_p is obtained by gradually increasing it from an initial conservative estimate and calculating

the derived system cost for each value until it is maximized – a process which can take significant amounts of time. Their approach is evaluated through an NS2 simulation, but the parameters of the simulation are chosen in a somewhat arbitrary way: request arrivals are modeled with a Poisson process, videos are assumed to be just 5 minutes long, the topology is artificially generated through Brite, and there are no traces or insights taken from a real world scenario. Simulation are cut at 30 minutes of simulated time, and we are not told if the system reaches some sort of regime at that point – indeed, from the graph it does not appear like that is the case.

The previous work is extended by Bruneau-Queyreix et al. (2014); here the HBs are allowed to keep a list of neighbors they can use to relay requests, bypassing the need for the SM/SR introduced above. The authors claim that this additional layer of caching can coexist with and improve the performance of a traditional CDN infrastructure; however our experiments (see Section 4.1.2) and established works in literature (Wolman et al. 1999) show that hierarchical caching isn't very helpful, and that a single layer can provide almost exactly the same benefits than those achievable by multiple layers of caching.

Jiang et al. (2012) describe what they define as a massively distributed CDN system, but really it is a telco-managed P2P system, as they use storage on NAS and STBs inside users' premises. They attempt to perform a joint optimization of content routing and content placement, in order to minimize CDN traffic costs. They assume to know the desirable replication ratio for each content item, given a fixed Poisson request rate. When they relax this assumption, they use the rates from the previous day as a projection of the requests for the current one. The paper, however, has a number of shortcomings. Contrary to what one would expect, in their results increasing the number of content items that can be cached at each STB also increases the loss probability, i.e., the probability of having to use the CDN infrastructure because there is no available local upload slot for a content item. When they have to redistribute content in the STBs, they make it so that a minimal number of transfers is required, in order to minimize bandwidth overhead – however they do not investigate the cost of keeping the system in a sub-optimal state for a longer time (they assume transfers are done during off-peak hours). Finally, when showing the performance of their algorithm against non-coordinated schemes like LFU or LRU, they do not include the overhead due to content pre-placement - indeed, their benchmark offline optimal scheme has a cost of 0, since it can just push content that it's going to be requested to STBs for "free".

Gramatikov et al. (2013) analyze a walled-garden peer-assisted system for the streaming of VoD content, where the STBs are under the control of the service provider, but servers are still required to guarantee QoS. The authors define a detailed mathematical model of the system behavior. Their reference network is based on DSL – hence the assumption that uplink streaming capacity is lower than the streaming bitrates and the reason why servers are deemed necessary. Users are connected to a DSLAM and can only share content with other people in the same community, so locality is enforced. Caching content is not based on viewing history but instead is chosen by the SP (pre-fetching). Naturally this content has to be redistributed periodically to optimize the storage policy. The model only performs a stationary state analysis and has a number of simplifying assumptions (e.g., the usual Poisson modeling

of request arrivals).

Most of the studies mentioned so far are either network-agnostic or assume a DSL access network infrastructure. There are, however, some papers specifically targeting PON-based access networks. For example, Kerpez et al. (2010) explicitly investigates P2P delivery of multimedia content over both GPON and Next Generation PONs. A hybrid hyperbolic (i.e., power law) and exponential distribution is used to model content popularity. An arbitrary limit of one third of the total upstream capacity of each peer is imposed on uploads, to ensure that the VoD service does not saturate it – this seems unnecessary. The authors also use a very high encoding bitrate of 19.3 Mbps for all content, much higher than the current typical HD quality and even of the 15 Mbps envisioned for 4k content. Montecarlo simulations are used to evaluate the proposed strategy, with a static content popularity and a random user distribution. The content storage is also random and not a function of what has been requested (i.e., their simulations are not time-driven, like in our steady-state analysis). In general their conclusions are in agreement with ours in terms of bandwidth efficiency.

Similarly, Jayasundara, Nirmalathas & Wong (2011) specifically target P2P caching for VoD services running over PON (but not LR-PON). Like other of the studies mentioned above, they pre-fetch content during off-peak hours; however videos are stored in the ONUs rather than the STBs. The idea is that the former are under the control of the ISPs and hence more stable; furthermore, one can use the PON broadcast nature to push content to multiple ONUs at the same time. However, in practice this is not really useful, as ONUs on the same PON share the same upload capacity pool, and in general storing multiple copies of the same movie in different ONUs on the same PON is intuitively sub-optimal. Furthermore, it is likely that STBs and ONUs would be integrated in a network-managed P2P scenario like the one we consider. The authors use an optimization algorithm to determine which content to replicate. More specifically, they are interested in the estimated number of simultaneous requests a content item will receive (similar to what we propose in Chapter 6). However they use a Poisson distribution to model request arrivals, and they assume that this request rate is known, without mentioning how it is estimated. Results show that their solution works better than a purely popularity based caching strategy. However they do not take into account the bandwidth consumed for the pre-fetching and they do not compare it to a pull-based system, i.e., with the content being determined by the requests of the users.

Hwang & Liem (2012) also propose a P2P scheme for VoD over PON-based networks. Their reference network, however, is based on EPON rather than LR-PON; furthermore, caches (called buffer maps) are very small – just 10 MB – and integrated into the ONUs. These are equipped with an additional receiver tuned at the upstream wavelength; a star coupler with an isolator is used to broadcast upstream traffic to all local ONUs when required. By doing so, all optical inter-ONU communication is made possible, i.e., without having to route packets electronically through the switch at the remote node. ONUs keep a routing table to determine which flows are local (intra-PON) and which are inter-PON. However it is not clear how these tables are built or maintained, nor which entity is in charge of redirecting requests

to an available cached copy of the content. Furthermore, to simplify the process of granting upstream bandwidth to P2P flows, ONUs use their RTT to the OLT also for P2P purposes. The assumption is that this value is higher than the actual ONU-to-ONU RTT, which makes sense if the star coupler is the only splitting stage, but it might be troublesome in our LR-PON architecture with multiple stages of splitting. Finally, the cache size seems to be exceedingly small to be significant; their simulations use artificially generated traffic and the amount of requests that can be served locally through P2P redirection is fixed as a simulation parameter, rather than a consequence of the caching process.

Naturally, there are a number of publications that address the application of P2P to VoD services without falling into any of the categories above. Some of these studies are listed below.

The residential gateways (RGWs) are the peers in FLaCoSt, a work by Garcia-Reinoso et al. (2009); however, while this proposal markets itself as P2P, it is really based on multi-tier Application Layer Multicast (ALM) trees. The P2P flavor comes from the fact that these trees are constructed as dynamic overlays between the RGWs, using a modified version of Pastry/Scribe. Each RGW decides whether or not it can support children in the multicast tree based on its available resources. Furthermore, peers can use SIP sessions to reserve bandwidth along these trees so to assure their sustainability. Unfortunately, this proposal suffers from all the typical disadvantages of network-unaware P2P, such as overlay/underlay mismatch, randomness and high deviation in the depth of the trees etc., without the benefits of pure IP multicast. As a consequence of these imbalances in the building of the multicast trees, weird artifacts appear in their simulation results, such as scenarios with abundant network resources performing worse than others with less.

Liu & Simon (2011) focus on peer-assisted time-shifted video service, where the users are allowed to rewind a live stream to watch segments they missed. In their measurements, performed over real usage traces, about 40% of the users are lagging behind the live stream by no more than 3 hours and about 80% by no more than 84 hours. Requests are generated synthetically to model the data obtained from the traces. A multiple-interval graph is kept by the tracker to know which peer can serve which other (based on the chunks that they currently hold). Each chunk represents one minute of video. Whenever a new chunk is needed, the tracker sends a set of potential sources to the requester, which in turn picks one randomly. However, the subset of peers sent by the tracker to the requester can be chosen either randomly, based on the available capacity of the peers, or using locality considerations. For their simulations, the authors use a worldwide topology with 28421 ASes, taken from the CAIDA project. Their findings are in line with what we would expect, with a lower load on the multimedia server thanks to P2P and a much smaller number of ASes traversed on average for content delivery compared to a centralized unicast solution. However, they do not perform any comparison with a CDN scenario.

Zhou et al. (2011) perform an analytic study of a push-based P2P caching system for a VoD service. However they use a rather simplistic popularity model – a transfer matrix that determines the probability of watching movie j after finishing movie i – in order to keep the stochastic content request process stationary. Furthermore, the paper does not take into account any sort of distance measure between

peers, nor any form of locality, nor the cost to push content to the caches. As as a results of these limitations, their caching algorithm, which is based on a random selection of video elements up to the point where the number of replicas are sufficient to meet the estimated demand for that content, is not particularly interesting in our opinion.

Wu & Lui (2012) discuss optimal caching strategies for a P2P based VoD system. They distinguish between active peers, which are currently watching a given content item, and inactive peers, which are not actively watching that item but can contribute to the stream through caching. Given these definitions, popular contents will naturally have much more active peers, so the authors argue that caching these contents will not provide much benefits. This argument is somewhat fallacious in that, in a non-live streaming service, active peers will be at very different stage of the content and, if no caching is enforced, there is going to be significant churn (i.e., users are likely to leave the system once they are done with their watching activity). Furthermore, in our study, content transfers are very fast due to the high capacity of next-gen PONs, and as such a peer would be active for a very limited amount of time during each request. Finally, their results are based on a very limited scenario with just a tiny number of videos and other parameter choices that in general do not seem to be sufficiently justified.

To summarize, our work extends the existing literature on P2P approaches to multimedia distribution by specifically targeting next-generation optical access network (the so-called LR-PON scenario); to the best of our knowledge we are the first to explore the benefits of symmetric access bandwidth for locality-aware policies. Furthermore, while there is a number of related publications in literature, many of those completely lack simulation results (e.g., Janardhan & Schulzrinne (2007), Laoutaris & Rodriguez (2008)) or limit themselves to simplistic simulation scenarios. For example, Kerpez et al. (2010) only perform a static Montecarlo analysis; Chellouche et al. (2012) run limited 30-minute simulations that do not reach a regime, using only a small catalog of 10000 elements of 5 minutes each of which only the most popular 10% can be cached by their HomeBoxes; Jayasundara, Nirmalathas & Wong (2011) mostly focus on pre-fetching without taking into account the bandwidth cost of pushing content to STBs, thus missing the benefit of a pull-based caching strategy; Hwang & Liem (2012) use exceedingly small buffers of just 10MB to support contents that on average can be in the order of magnitude of the Gigabytes, etc. Some studies presented here depart from our work by adopting BitTorrent-like protocols, which are sub-optimal for multimedia streaming due to the erratic order in which the content is retrieved (despite the work by Parvez et al. (2012) attempting to mitigate this issue); others do not take into consideration locality awareness, focusing instead on other aspects such as resiliency to peer churn (Chen et al. 2009) or maximizing the overall throughput of the system (Peterson & Sirer 2009).

2.3.2 Analysis of the Energy Consumption of Multimedia Caching

Various authors attempted to investigate the power consumption of telecommunication networks; in this section we give a brief overview of the papers that tackle the problem of multimedia distribution

in their analysis.

Baliga, Ayre, Hinton, Sorin & Tucker (2009) present an analytic model based on a "paper design" of a network that can provide a certain access rate to customers of a national broadband service. The model calculates the per-customer power consumption for all the equipment in the network as a function of the peak access rate of the customers. With regards to multimedia services, the authors assume the existence of a separate Video Distribution Network (VDN), which uses IP multicast for static IPTV channels and has some bandwidth reserved for VoD and premium IP services. The VDN is assumed to bypass the metro/core routers and plugged in directly into the aggregation Ethernet switch at the edge between metro and access segments. However, the authors do compare their results with an alternative architecture lacking a dedicated VDN, showing that the its presence greatly helps in reducing the total power consumption.

The model presented in this paper has been widely cited and used as a reference in many of the successive works in literature. In particular, the authors extend their work in (Baliga, Ayre, Hinton & Tucker 2009) by applying their energy consumption model to the specific problem of on-demand multimedia content delivery over the common Internet infrastructure. The authors investigate the efficiency of replicating content at different layers of the network based on the number of downloads per hours that they will receive. They show that for low download rates the storage component dominates the total power consumption, while for very popular content transmission costs are dominant. This translates to an optimal policy of replicating popular elements near the access section of the customers, while keeping the less popular content in a small number of servers in the core. The authors also investigate the merits of P2P; their results indicate that it is only beneficial for movies that are not very popular. Our findings are in complete disagreement with theirs, possibly due to the limitations of their paper-based model, the arbitrary limit of 4 Mbps that they impose on the upload capacity of the STBs, and the lack of recognition of the fact that content requested by users can be cached on their STBs without any additional power cost.

Feldmann et al. (2010) use the same model to analyze the power consumption of various multimedia delivery strategies, e.g., based on a central data server, using CDNs or through P2P-enabled set-top boxes. Their conclusions show that while P2P lowers the energy consumption of ISPs, it increases the overall consumption of the network, and thus it is sub-optimal compared to CDN. It is worth noting that their reference network is based on VDSL in the access, the upstream capacity of their set-top boxes is limited to 5 Mbps, their traffic model (e.g. average number of hops, locality of content etc.) is based on what the authors loosely call "common sense estimates", and no FTTH analysis is provided. The network is quite small compared to ours (10000 users) and the data catalog has the same size of the customer base (10000 elements).

The paper by Chan et al. (2011) further extends the work of Baliga, Ayre, Hinton & Tucker (2009) by allowing the streaming of different portions of a video from caches sitting at different network layers (i.e., at the OLT, at the metro edge router, at a cache in the core, or from the origin server in a data

center). They then evaluate the benefits of a single-layer of caching in the access versus a multi-layer approach, as a function of the similarity of requests between users from a same community, i.e., of the likelihood of users located geographically near to each other to request the same content items due to a similarity in interests. Their results show that single-layer caching is always less convenient than multi-layer caching, but also that for communities with very high similarity scores this difference is negligible. As already noted above, there is a number of existing publications showing that the benefits of additional layers of caching past the first are moderate at best.

Similarly, Jayasundara & Nirmalathas (2011) use Baliga's model to evaluate the power consumption of VoD caches at different depths in the network, i.e., closer or further away from the end user. Note that however they only consider a pre-fetching scheme where content is replicated uniformly in every single cache at a given layer, with no possibility of P2P flows from other caches at the same layer. For this reason, all but the most popular elements are better cached further away from the user, since replicating less popular videos in each ONU would require immense storage and provide little benefits. This work is extended in (Jayasundara, Nirmalathas, Wong & Chan 2011) where, by using a Zipf popularity distribution and Little's Theorem, the authors show the minimum rate of requests that a content item must receive on average before it's convenient to store it at a certain layer in the network. We believe that these studies, while interesting in their own right, fail to recognize the potential of caching requested content and allowing for P2P flows between caches at the same network depth.

Guan et al. (2011) compare the energy efficiency of traditional CDNs, ICNs, and a centralized CDN solution with a dynamic optical bypass for the purpose of content delivery. In other words, they attempt to evaluate the benefits of dynamically provisioning a transparent optical path across the core between source and destination, thus bypassing core IP routers. They find that on average ICNs are the most efficient solution for popular content items, which can be cached at routers near the edge. For lower popularity objects, an optical bypass is the most energy-efficient alternative, although the costs of setting up such dynamic transparent paths are not assessed. This paper does not analyze the energy efficiency of P2P.

Savi et al. (2014) propose an energy saving algorithm for VoD distribution over a tree-shaped active network, i.e., with switches at every splitting point in the tree, as opposed to the passive splitters of PONs. The authors define a usage-proportional model for both the transport and caching components of the total energy consumption equation. Their architecture assumes that the whole catalog is replicated both in a central repository for the network and in CDN servers located closer to the users, i.e., at the second splitting stage of the distribution tree. The idea then is to switch the CDN servers on or off depending on the measured request rate from VoD customers, and the corresponding aggregated VoD traffic. Using this strategy their algorithm is shown to be able to reduce energy usage compared to both a pure unicast solution and a pure CDN one. Note however that their model assumes a static multimedia catalog and thus ignores the cost associated with the transport of copies of the new elements from the central repository to all the CDN servers. Naturally these distributed servers cannot be updated through

pull-based updates from customers when they are turned off.

Fiorani et al. (2014) perform an extensive analysis of the energy consumption of a network based on their Hybrid Optical Switching (HOS) concept – a technology that allows the coexistence of different optical switching paradigms (i.e., packet, burst and circuit switching). In particular, in this work they propose an integrated intra-datacenter and core network architecture based on the HOS paradigm, and they investigate the efficiency of edge-caching for the purpose of multimedia content delivery. They find that the size of the edge caches is a crucial parameter for the overall efficiency of the system, and that in general edge caching is advantageous for their architecture only when considering a high load scenario.

Mandal et al. (2014) propose an analytic model of the energy consumption of a hybrid CDN-P2P system for multimedia delivery. In order to do so, they make a number of simplifying assumptions, such as Poisson modeling of request arrivals and a catalog with a single element (i.e., no inter-catalog competition for caching and no popularity dynamics). They propose two algorithms to determine the most appropriate strategy to serve a request, either using the CDN infrastructure or a number of available peers; these algorithms attempt to respectively minimize server load or reduce the instantaneous energy consumption. Their results show that significant bandwidth savings can be achieved without increasing energy consumption or, in some scenarios, even reducing the total energy consumed (depending on the arrival rates and the time that peers spend in the system after completing their download).

Our work differentiates itself from all those inspired by the work from Baliga, Ayre, Hinton, Sorin & Tucker (2009) in that we do not attempt to build an analytic model of the network, but rather rely on simulation results to estimate the power consumption of the network. While a theoretical model can be a useful tool of analysis, it is probably not the best approach to investigate the benefits of a pull-based caching system, where the dynamics of content availability are too complex to be modeled formally. Indeed, all the aforementioned studies only consider pre-fetching schemes, where content availability at each caching location can be perfectly predicted; these strategies heavily penalize P2P by enforcing a full replication of the selected content in every single STB, which leads to heavy energy costs related to both the transmission and storage of all these replicas.

2.3.3 Popularity Estimation Algorithms for Multimedia Services

This section provides an overview of the relevant works in literature attempting to model the popularity of multimedia elements for the purpose of improving the efficiency of a multimedia distribution system.

The work by Yu et al. (2006) is one of the first studies analyzing a real VoD deployment, namely a free streaming service offered by a large Chinese ISP in 2004. The authors find that the use of a Poisson distribution to model user arrivals over the peak-load time window tends to over-estimate the number of large groups of users and conversely to under-estimate smaller arrivals. The study also shows that a very large portion of the video requests have a very short session length (i.e., less than 10 minutes), due to users sampling the catalog while looking for content of interest.

Thouin & Coates (2007) describe the design principles and possible challenges of a VoD system supported by P2P content distribution. With regards to the issue of estimating video popularity, it simply elaborates on the "fetch-at-most-once" model proposed in Gummadi et al. (2003), showing that in a typical VoD system the top-ranked items contribute to a smaller percentage of the total requests compared to a typical Zipf-like model.

Hefeeda & Saleh (2008) propose a caching strategy to reduce inter-AS traffic in P2P file-sharing applications. In order to do this, they perform an extensive measurement of the popularity of objects in the Gnutella network. Their results are similar to those observed in Gummadi et al. (2003), with a popularity distribution showing a flat-head towards the top-ranked objects. The authors show that a Mandelbrot-Zipf distribution, which is a generalized Zipf distribution with an additional additive parameter, is able to model the object popularity with a greater accuracy. This is the model we used in most of our experiments.

Avramova et al. (2009) collect traces from a number of different sources, monitoring the number of views that a set of video elements had collected over time. They propose a parametric formula to model the number of requests that an object will have received after a certain amount of time; depending on the choice of parameters, this function can turn into an exponential-like distribution or a heavy-tail power-law distribution. The authors find that a varying percentage of these elements exhibit exponential behavior (about 20% for YouTube and 50% for a catch-up TV portal), while the rest of the traces follow a power-law distribution.

Based on the statistics on the top-50 DVD rentals in the US for a number of years, De Vleeschauwer & Laevens (2009) show that the popularity of these video objects decreases exponentially over time together with their generated revenue. The average parameters extracted from these statistics are then used to model the number of requests a certain video will receive from its introduction in the VoD system to a given instant in time. A tracking algorithm attempts to use this model to determine the actual popularity of video elements compared to other videos in the catalog; this algorithm informs a caching policy, which is shown to out-perform LFU and marginally improve on LRU, given an appropriate choice of parameters. There are however a number of simplifying assumptions which in our opinion would deserve more attention, such as: equating physical DVD rentals with VoD requests, Poisson distribution for both requests and the introduction of new elements, memory occupation of the tracking algorithm for a large number of elements, independence of the popularity of various objects in the catalog, no modeling of user behavior outside of the requests for a single element (i.e., lack of a limit on the number of viewing hours that can be spent by the user, or actual viewing patterns).

The previous two works were expanded and improved upon in (Avramova et al. 2010), where the performance of the caching algorithm is compared with those of an ideal algorithm with perfect knowledge of future requests, showing a significant gap in all cases. Furthermore, it is shown that it is possible to accurately model the captured traces through a set of clustered distributions based on the values assumed by the parameters of their demand function; however, identifying the correct parameter ranges

that define these clusters is not straightforward.

The paper by Cha et al. (2009) focuses on an analysis of the popularity of User Generated Content (UGC), e.g., in systems like YouTube. The authors show that UGC has distinctly different characteristics from traditional VoD services: a UGC catalog tends to be several orders of magnitude bigger, composed mostly by short videos, many of which hardly receive any request. Their popularity is shown to be well modeled by a power-law distribution with an exponential cutoff.

The work by Jayasundara et al. (2010) is based on the idea of the paper from the same author detailed in Subsection 2.3.1, i.e., placing small caches in a PON and using the broadcast nature of the downstream channel to automatically store popular contents. In here, however, the focus is on the popularity prediction algorithm that determines which elements should be cached. The authors propose a Last-k algorithm which keeps track of the intervals of time between the latest arrival and the previous k-1 ones. The sum of these arrivals is used as a measure of the popularity of an object, with higher sums equating to lower popularity. This approach is showed to be as efficient as LFU, but with a quicker response time to changes, e.g., due to the introduction of new elements. The tuning of the parameter k is however not investigated, and larger values of k appear to increase the response time of the algorithm; at the same time, the authors claim that larger k will increase the accuracy of the predictions.

Borghol et al. (2011) analyze an extensive set of traces from YouTube, showing that the amount of weekly requests for user-generated videos is largely week-invariant and only depends on the phase of the video with regards to its peaking week (i.e., the week in which it will receive its biggest surge of new requests). This allows the formulation of an algorithm to artificially generate weekly views closely matching the behavior of the real traces. In our work, we use this model to approximate the popularity evolution of a VoD catalog (see Section 3.2.2).

Similarly Figueiredo et al. (2011) analyze three datasets of YouTube videos, namely Top (including all the videos from the various national top-100 lists on the day of the sampling), YouTomb (including videos which had been removed from YouTube due to copyright infringement) and Random (including the first results of searches for random words or tags). They show that YouTomb videos gain much of their popularity in their early life stage, followed by Top and Random. Top videos seem to have the highest bursts of popularity over a single day, but all datasets present a somewhat bursty nature. Finally, the authors show that internal YouTube mechanics such as the search function and related videos list are key mechanisms to attract new users to a video.

Famaey et al. (2011) investigate the benefits that could be achieved by theoretic caching algorithms that have a perfect knowledge of future requests, albeit over a finite time window. Two such algorithms are described, based respectively on recency (P-LRU) and frequency (P-LFU). Their performance is compared to those of an algorithm whose knowledge of the future is not limited, representing the optimal benchmark. All tests are performed using a dataset of traces taken from a real VoD deployment. The authors also show that the optimal size of the prediction window is dependent on the cache size, and

they investigate these interactions. However, it is obvious that the proposed algorithms are not usable in real deployments and are only meant as a way to estimate upper bounds of what can be achieved.

As an extension of this work, in (Famaey et al. 2013) the authors use non-linear optimization techniques to fit the historical data of observed requests for a given video element to a set of statistical distributions. The extrapolated model is then used to predict future request patterns for that video, and hence to inform the cache eviction policy. However the fitting operation was shown to be computationally intensive and to require an extensive set of datapoints to perform adequately. Furthermore, once again the authors only describe the performance of theoretical versions of their prediction algorithm, which use future information not available in a real system deployment.

Abrahamsson & Nordmark (2012) present another analysis of traces taken from a real time-shifted IPTV and VoD deployment from Northern Europe. The main contribution of their work is showing how different types of videos have different popularity curves: i.e., on-demand movies show a long tail of requests, TV news only stay relevant for a few hours after their first airing, and episodes of TV series see a small bump in popularity near the release of another episode. They also show how different types of contents peak at different times, with cartoons predominantly being watched in the morning and movies being accessed at prime time in the evening or during weekends.

Ling et al. (2014) use an exponential weighted moving average of the access intervals of video elements to determine the popularity of contents for the purpose of defining the cache eviction policy. They also introduce a separation between the segment size in which a video is divided for the purpose of managing requests and the block size which is the minimal unit of caching storage. This generates additional complexity, as each segment might only be cached partially, but allows for a better utilization of the caching space.

Compared to these studies, the algorithm we use in our work on caching optimization (see Chapter 6) is much simpler and admittedly less ambitious, as we only calculate the average number of requests that a content of a given popularity rank has received in the past and then use this as a measure of the future number of requests for elements of similar popularity. While being less sophisticated, this solution has the advantage of being much less onerous computationally than some of the strategies presented here (e.g., in (Jayasundara et al. 2010) or (Famaey et al. 2013)) and it is shown to be sufficient for our purposes. Furthermore, unlike in (Avramova et al. 2010) and (Famaey et al. 2011), our algorithm uses only past historical information that is actually available through the locality oracle, and as such it could be implemented in a real world system.

3 Simulation Methodology

In this chapter we detail the two simulation tools that we developed to perform our studies: a steady-state simulator which was created as a preliminary step to verify the feasibility of our approach, and a more detailed event-driven tool which was used for the reminder of our analyses.

Indeed, several challenges had to be tackled in order to evaluate the efficiency of our proposed approach to multimedia content distribution. The most obvious one was the lack of a physical implementation of our reference scenario, i.e., a symmetric 10Gbps upstream/downstream access network with enough reach to allow for a consolidated architecture, with a limited number of metro/core nodes replacing the thousand local exchanges of today's networks, as promised by current LR-PON proposals. Unfortunately, the technology required to implement this vision is still being perfected in research labs around the world, and as such it is not yet available for testing.

In addition to that, due to the nature of the multimedia delivery problem, testing relevant use cases requires the reproduction of the behavior of a very large set of users. Since our strategy relies on opportunistic caching of content requested by users (i.e., *pull-based caching*), its effectiveness is dependent on the size of the user base. A larger number of customers will generate more requests, but also provide a larger distributed caching space. As we aimed to study a nation-wide multimedia on-demand service, we needed to be able to scale to very large numbers, up to the order of magnitude of millions of users – something that is obviously not practical over a physical testbed. For these reasons, simulations were really the only available option.

Naturally, even within a simulation environment, scaling up to these numbers is not a trivial task. In order to do so within reasonable execution times we had to be willing to sacrifice accuracy in some departments. This is, in our opinion, an acceptable trade-off, since we are not really interested in the fine-grained detail of physical layer emulation, nor in the intricacies of network protocol implementations, but rather in analyzing the effect of user watching patterns and cache size on the traffic generated in the various segments of the network.

As none of the simulators available seemed to satisfy these scalability requirements, we decided to develop our own tools. The first step in this direction was the creation of a steady-state traffic load analyzer, which was used to test the soundness of our assumptions. Section 3.1 describes the development process and the functionality of this basic simulator.

Unfortunately our steady-state analyzer was unable to take into account the dynamic aspects of multimedia delivery and caching, such as the evolving popularity of video elements, the volatility of the content stored on user Set-Top Boxes (due to their limited cache size and the replacement policies running on each of them), and the bandwidth constraints of the network links when facing a surge of simultaneous requests. In order to assess the impact of all of these factors on our proposed solution, we developed a flow-based, event-driven simulator, which we called PLACeS (Peer-to-peer Locality Aware Content dElivery Simulator).

From Section 3.2 onwards, this chapter describes the design principles that were used in the development of PLACeS, the models that were used to emulate popularity dynamics, and the limitations of the simulator.

3.1 Steady-State Analyzer

The steady-state simulator was developed to better understand the effects of symmetric access bandwidth and node consolidation on the efficiency of locality-aware mechanisms for peer-to-peer (P2P) content distribution. Locality-aware policies have been repeatedly proposed by researchers (e.g., Xie et al. (2007), Choffnes & Bustamante (2008), Aggarwal et al. (2007), Seedorf et al. (2009)) for their intuitive benefits: by limiting P2P exchanges to peers residing in the same Autonomous System (AS) it is possible to reduce traffic over the Internet backbone, speed up data transfers by reducing network latency and, most importantly, drastically reduce costs for ISPs.

In order to understand why this is the case, we need to make a brief digression on network topologies and service providers. The Internet is composed by a number of backbone networks interconnected at some specific *peering points*. Since handling incoming traffic requires the provision of network resources, some ISPs do not allow external flows to transit over their network unless the flow's destination resides in the portion of the network under they control or, alternatively, unless some sort of agreement is reached between the involved parties. Sometimes, if the amount of traffic that crosses an AS boundary in each direction is roughly equal, the respective network owners reach a settlement-free peering agreement, in which no payment is made under normal circumstances. In many other cases, small-sized operators are forced to buy connectivity from larger ISPs in the form of IP transit rights, i.e., paying a fee that is roughly proportional to the amount of traffic they send across the larger operator's network. While a complete coverage of these issues is outside the scope of this thesis, it should be evident that it's in the interest of most ISPs to prevent traffic from crossing their AS boundaries unless it's necessary, and particularly over those peering links for which no settlement-free agreement exists. Unfortunately, P2P applications in general have no knowledge of such agreements (or of the location of AS boundaries, for that matter), and their typical random selection peer policy can generate high amounts of costly inter-AS traffic for operators; hence the research interest in locality-aware policies for P2P.

Regardless of how they are implemented, however, these policies can only be successful for a given

P2P client when there are enough local peers sharing its desired content. P2P protocols such as BitTorrent typically aim for a ratio of 5-10 uploading peers for each downloading client, in order to overcome the discrepancies between upload and download bandwidth of residential DSL customers. Considering that on average 82% of all torrents have less than 10 active peers in the entire network at any given moment (Zhang et al. 2010), it should be evident that finding enough local peers to satisfy the 1:10 ratio is often hard if not impossible; therefore, locality schemes have limited applicability in traditional networks.

However, this imbalance between upload and download bandwidth is only due to the current limitations of copper-pair based technology. As mentioned in Section 2.1, current Fiber-To-The-Home (FTTH) deployments have already reduced this ratio to 4:1 or even 2:1 in some cases (e.g., GPON), with next-generation optical access network architectures such as LR-PONs expected to further reduce it to an even 1:1 ratio [1]. We argue that in a symmetric bandwidth framework such as our next-gen PON scenario, a single uploading peer with the desired content would be able to completely fulfill a client's request by matching its download capacity. Obviously in a real environment it would still be advisable to have more than one active peering connection at any time for robustness and resiliency reasons; however, as we do not include peer churn and node failures in our analysis, in the simulations we assumed that symmetric bandwidth clients only need to contact a single peer to retrieve their desired content at an acceptable speed.

Similarly, the architectural transformations and node consolidation enabled by the increased reach of these next-generation PON technologies will likely affect locality policies by expanding the pool of neighboring users from which a desired content can be retrieved. To investigate the impact of these architectural changes on locality-awareness, we run our simulations on both a traditional three-tier topology with distinct Core, Metro and Access sections, and an envisioned future network with long-reach PONs bypassing the metro area to directly interconnect end-users to a limited set of metro/core nodes.

The aim of our steady-state simulations is to test the hypothesis that symmetric access bandwidth and the customer aggregation enabled by future PON architectures will improve the efficiency of locality-aware policies, both by (a) reducing the number of uploading peers required at each client to maximize downstream bandwidth and (b) greatly increasing the chance to find a local peer sharing the required content. Note that, while locality policies usually aim to confine P2P traffic in its originating Autonomous System network, in this thesis we loosely use the term locality to indicate any policy which takes into account topology information to identify the peers located closest to the target client. Specifically, the distance between any two nodes is measured as the number of hops required to go from source to

[1] This imbalance is not due to a bandwidth limitation as in the case of copper-based DSL access technologies, but rather to the high cost of burst-mode receivers capable of supporting higher data rates. These components, which are required to handle the different levels of attenuation and dispersion in the upstream channel affecting distinct users, are more expensive than their continuous-wave counterpart because they are a younger technology with a smaller market, but their price is going to decrease as FTTX solutions become more widespread. Furthermore, burst-mode receivers are deployed in the OLTs and shared among the multiple users of a PON, making their additional cost per user small.

destination, and a request is considered to be served locally if it does not traverse any core link.

A simple simulation tool was hence developed in C++ to perform a steady-state evaluation of traffic loads imposed by different content delivery schemes under various network conditions. In each simulation run, we compare the performance of the following content delivery methods:

- unicast client-server, from a central repository in the network which holds a copy of every element in the catalog;
- Content Delivery Networks (CDN), i.e., delivery from one of a set of cache servers located at the edge of the core network;
- non-local peer-to-peer (P2P), i.e., retrieving content from randomly chosen users holding a cached copy of it;
- locality-aware P2P, i.e., as above, but prioritizing P2P sources which belong to the same access section of the requester.

Note that for locality-aware P2P we simulated two different scenarios, respectively with asymmetric (1:10) and symmetric (1:1) upstream/downstream access bandwidth in order to assess the impact of symmetric access bandwidth, as detailed above.

Simulations were run on a sample topology generated in the following fashion: the European optical backbone prototype network described in Aiyarak et al. (1997) was chosen as a reference for a small national core network of 20 nodes. Each of the core nodes was subsequently been expanded into a ring of metro nodes; each metro node was in turn connected to an access node, aggregating up to 20 000 end-users. As a result, a 3-tier network of up to 2 millions customers could be simulated. We also considered a potential evolution of this sample network topology, in which the introduction of LR-PON access technology eliminates the need for metro rings; in this case, core nodes directly aggregate access traffic, as shown in Fig. 1.1. On each of the target topologies, we ran simulations with 1%, 10% and 100% active users. The first two scenarios represent different adoption rates of the proposed content delivery technologies; the 100% case models a network-managed system in which the content distributor is able to leverage always-on equipment deployed at the customers premises, such as set-top boxes.

Content popularity, which is assumed to be static, is modeled using a Zipf-Mandelbrot distribution; the exponent and shift parameters of the distribution were chosen in accordance with the findings of Saleh & Hefeeda (2006). Placement of CDN servers in a subset of the core nodes is performed using a nearly-optimal greedy algorithm which solves the k-median problem (Qiu & Padmanabhan 2001), with $k = 10$; more specifically, at each iteration a single server is added in the location that minimizes total transmission costs, without taking in consideration the future iteration steps. Each CDN replica hosts half of the total data catalog; elements to be stored in the cache are chosen randomly according to the Zipf-Mandelbrot popularity distribution. In the *CDN* scenario, content is served from the nearest server to the requester holding a copy of the desired content.

Table 3.1: Parameters for the steady-state simulation tool

Parameter	**Typical Values**
% of active PON clients	1%, 10%, 100%
Peers to contact at each client	10, 1
Content elements in the catalog	10 000
Exponent parameter of the ZM Distribution	0.6
Shift parameter of the ZM Distribution	20
CDN replicas in the core network	10
Content types cached at each peer	1, 2
Content types cached at each CDN replica	5 000

Similarly, peers are assumed to only cache 1 or 2 elements from the content catalog, in accordance with the findings of Aperjis & Johari (2011); these items are chosen randomly for each peer based on their popularity. For each P2P request, viable sources are sought either randomly across the whole network (for the *random P2P* scenario) or attempting to find the closest source in terms of number of hops required (for the *locality-aware P2P* scenario). Furthermore, in the *asymmetric* scenarios we require 10 sources each uploading one tenth of a traffic unit, while in the *symmetric* scenario a single source will be responsible for the whole data transfer. When the desired content is not available at a sufficient number of peers, it is retrieved from a central server located in the core section of the network. The most significant simulation parameters and their typical values can be found in Table 3.1.

The size of each element of the video catalog is assumed to be constant and equal to one generic traffic unit; by summing those units of load over each of the hops required to deliver content to the interested customers, our simulator measures the aggregated traffic imposed on each of the network links. Routes are calculated statically using Dijkstra's algorithm. Note that for peer-to-peer we have to account for the fraction of content that is shared from each peer, e.g., one tenth of a unit per peer in the case of asymmetric 1:10 schemes.

We differentiate traffic belonging to the core, metro and access regions respectively, as the energy and economic cost of transferring one unit of data varies depending on the characteristics of the network. As our tool does not perform any time analysis, link congestion scenarios are not simulated; hence, link capacities are not a constraint of the model.

The results of this simulation campaign, published in a 2012 ONDM paper (Di Pascale et al. 2012), are reported in Section 4.1.1 and Section 4.2.1, respectively for the bandwidth and energy efficiency. While they provided us with good data on the quality of our proposed solution, it is evident that there are many aspects of a real world system that are not correctly modeled by this simple tool. More specifically, a steady-state analysis prevents us from studying the impact of any kind of time-driven dynamic, be it related to changes in the popularity of content, bandwidth constraints on network links during flash crowds, or to the volatility of the elements stored in P2P and CDN caches. In order to test our system in the face of these dynamic changes, we had to move to an event-driven simulator.

3.2 PLACeS: An Event-Driven, Flow-Based Simulator

PLACeS, or Peer-to-peer Locality Aware Content dElivery Simulator, represents our attempt to overcome the limitations of the steady-state traffic analyzer. It is an event-driven simulator, which allows us to introduce time-dependent dynamics such as popularity evolutions, link congestions etc. It models data exchanges as end-to-end flows between a source and a destination, abstracting the complexity of networking protocols in favor of a centralized bandwidth sharing mechanism among flows competing on the same network links. It supports two different popularity models, representing respectively a time-shifted IPTV service (e.g., like the BBC iPlayer, Sky Go/Sky Anytime+, or the RTE Player) and a more traditional Video on Demand service (e.g., like Netflix or YouTube).

The next paragraphs detail key aspects of its software development process, the models that were used to determine the behavior of customers and map the popularity of video contents, and the bandwidth sharing algorithm used. The source code for the simulator is freely available on GitHub; a link is provided at `https://sites.google.com/a/tcd.ie/edipascale/software`.

3.2.1 Software Development

PLACeS was developed as a C++ stand-alone application on Linux/Unix. It relies heavily on the Boost libraries for various tasks, including:

- management of command-line options and parameters, through the `program_options` library;
- random generators and statistical distributions (such as Zipf, Uniform, Gaussian etc.) through the `random` library, with the addition of the open-source Zipf-Mandelbrot library implemented by Kenta Murata;
- the `binomial_heap` structure used to implement the event queue;
- the `graph/adjacent_list` structure to store a representation of the network topology;
- the `dijkstra_shortest_path` algorithm implementation to implement routing;

and many more. The purpose of relying on well-tested third-party libraries whenever possible was both to reduce development complexity and to ensure reliability of the key components of the simulator.

Sometimes we could not find suitable implementations of a data structure we required; when this happened, we developed our own version as a generic template library, which was unit-tested and, in some cases, released as stand-alone project on Github. As a notable example, we developed a generic cache data structure, supporting both Least-Recently Used (LRU) and Least-Frequently Used (LFU) eviction policies; this was used to model the caches in users' set-top boxes, as well as the large cache servers used for CDN systems. We also implemented a ranking table to dynamically keep track of the rank of video elements based on the number of hits observed for each of them.

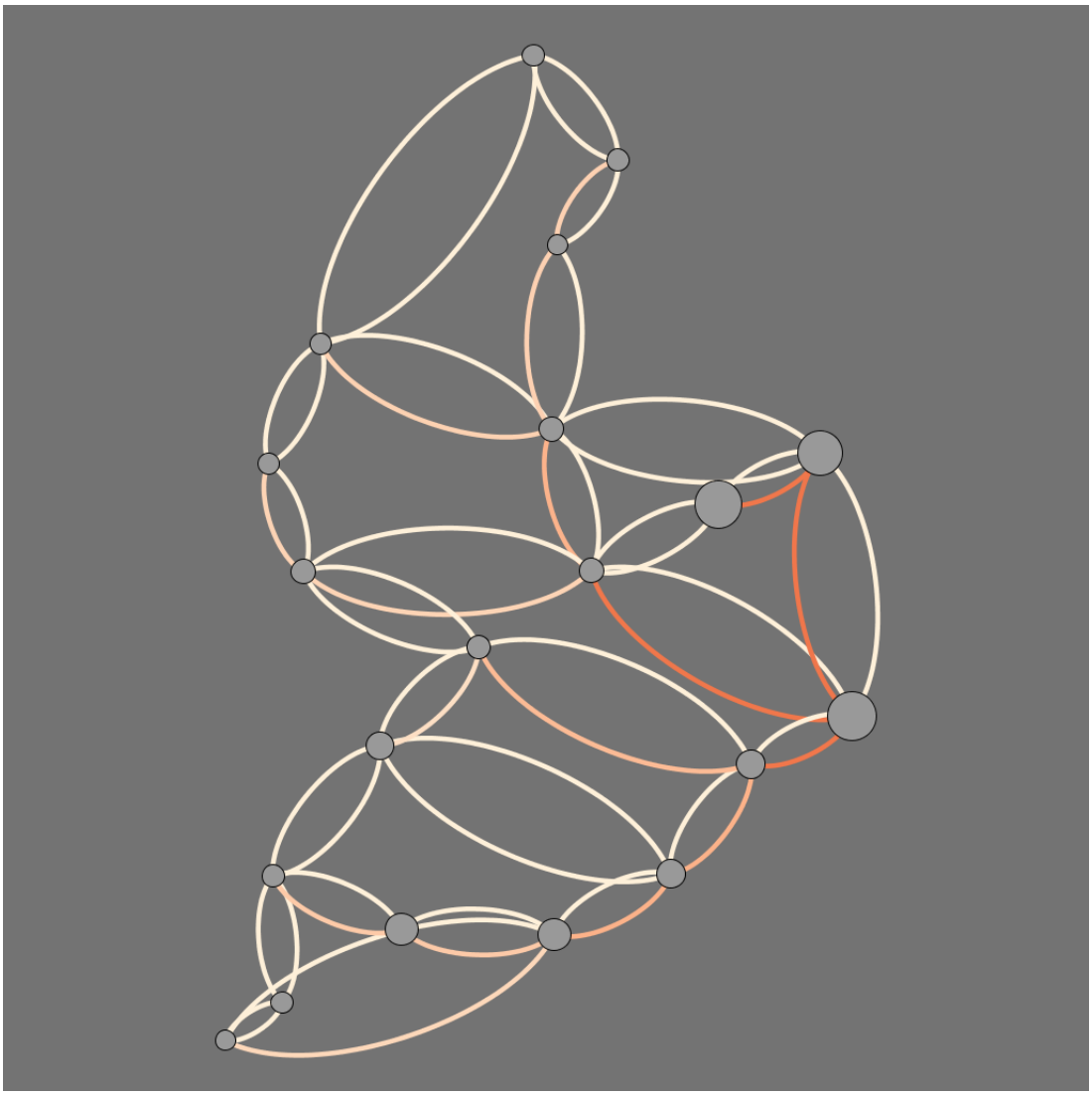

Figure 3.1: A snapshot of a traffic simulation using Unicast from a central repository located in the rightmost node. The color of the links indicate the amount of traffic observed on average, with "hot" links being close to saturation. The size of each node is proportional to the number of users attached to it.

The simulator is able to import network topology descriptions either as a standard *graphml* file or as a text file using a custom syntax, describing the number of nodes, the links connecting them with their respective bandwidth capacity in each direction, and the users connected to each node (in terms of an average number of users and a standard deviation, which will be used to generate a Gaussian distribution from which the actual number will be extracted). The simulator is also able to routinely output a *graphml* snapshot of the topology under simulation, which includes the average traffic observed on each of its links. These snapshots can then be edited with any standard graph visualization platform, such as Gephi[2], to generate a heatmap like the one in Fig. 3.1.

[2]http://gephi.github.io/

Results are shown on screen at the end of each simulation round, and then printed to file at the end of the simulation, both on a per-round and an aggregate basis. The metrics included in the output file include:

- average and peak traffic observed on upstream and downstream access links, core links and, if present, metro links;
- number of video requests served or blocked (i.e., for lack of an un-congested source, as described in Sec. 3.2.3);
- number of requests served locally to the access section of the requester, also expressed as a percentage of the total requests served;
- number (and percentage of the total) of requests served respectively through peer-to-peer, core-located caches or via unicast from a central repository;
- average time required to complete a transfer for requests served through P2P and through CDN caches;
- time-weighted average cache occupancy of the user set-top boxes and of CDN cache servers.

3.2.2 Popularity Evolution

A very relevant step towards the simulation of a multimedia distribution system is the modeling of content requests. Capturing the complexity behind the dynamics of video popularity has proved to be an elusive task, also partly due to the reluctance of the major VoD players to share what they perceive as sensitive data on user behavior. While some authors have tried to come up with a closed-form expression based on tunable parameters to model the amount of requests that a video element would receive throughout its life, what we required for our analysis was some sort of predictive algorithm to adequately simulate user requests.

Video On Demand

As mentioned in Subsection 2.3.3, Borghol et al. (2011) have shown that the amount of weekly requests received by a user-generated video is largely week-invariant throughout its life, and only depends on the phase of the video with regards to its peaking week (i.e., the week in which it will receive its biggest surge of new requests). This allows the formulation of an algorithm to artificially generate weekly views closely matching the behavior of real YouTube traces. In PLACeS, we use this model to approximate the popularity evolution of a VoD catalog; while some caveats are required, as the difference in size between typical user generated content items and VoD movies might affect usage patterns, we believe that this is a reasonable approximation. In fact, it is becoming more and more common to upload entire

movies or series episodes on YouTube, and at the same time the catalog of many VoD service providers is expanding to include content items of shorter length such as series or animated shorts.

Specifically, the aforementioned algorithm generates the peaking week and the weekly views for each content element of the catalog. These views are then scaled to the number of active users in the system, to ensure that the total hours of streamed content per user converge to an average of 4 hours per week. Finally, the views predicted by the popularity model are mapped to content requests from randomly chosen PON users. The starting time of each request is determined by a probability distribution modeling real peak-usage statistics, both in terms of hour of the day and day of the week; relevant data was taken from (Choi et al. 2012).

Our VoD catalog is composed of 50000 elements; in comparison, at the time when the VoD study was conducted, Netflix's catalog was believed to include 60000 different videos, while Amazon Prime was estimated to have around 17000 items. Each video is encoded at a constant 3 Mbps bitrate. The length of each content element is normally distributed with a mean of 120 minutes and a standard deviation of 15 minutes, to match the length of an average blockbuster movie. For VoD, unlike in the IPTV scenario, we do not consider the introduction of new video elements to the catalog after the start of the simulations.

Time-Shifted IPTV

Compared to Video On Demand, Time-shifted IPTV (also known as catch-up TV) has very peculiar characteristics in terms of popularity evolution. A very large number of new videos is made available each day by each channel, generating a constant stream of new content. On the other hand, most service providers remove older contents from their catalog after a limited time from their first publication (typically 7 to 14 days), partly to limit the size of the catalog and partly to protect the interests of copyright holders. This has inevitably consequences on the effectiveness of caching algorithms.

For IPTV, each round of simulation spans a single day, rather than a week as in the VoD case. At the beginning of each day we generate a new TV session for each user in the system, whose length is extracted from a normal distribution with a mean of 5 hours and a variance of 1 hour, to model the typical amount of TV time for an average US household. A time of the day representing the middle point of each user's viewing session is extracted from a discrete distribution modeling real usage patterns, in order to capture the concentration in time of content requests during peak hours; once again, the relevant data was taken from (Choi et al. 2012).

In order to actually generate video requests for each active user, we use a Zipf distribution to select the release day of the desired content (with newer videos having a higher probability of being chosen), and then a Zipf-Mandelbrot distribution to choose a video among those released on the selected day (see Section 2.2.3). To account for the "zapping" users, we assume that half of the time users will switch to a new video element after a random fraction of the original content length (between 10% and 90%),

rather than watching the entire content. The request generation process is repeated for each user until its daily viewing session has been completed.

Our catalog is composed of 100 TV channels, each releasing 30 new videos per day. Video elements have a mean length of 45 minutes and a standard deviation of 5 minutes, as shorter videos such as news programs are typically not included in the time-shifted catalog. At the end of each day, after releasing the new content, video elements older than a week are removed from the catalog.

A 5 Mbps bitrate is assumed, equivalent to Netflix current High Definition (HD) quality encoding (*Internet Connection Speed Recommendations - Netflix Help Center* n.d.). While time-shifted IPTV is often not delivered at such high quality bitrates at the moment, we assume that in the future HD content will become more and more the norm, especially with higher resolutions (e.g., 4k video) already making their first appearances. In order to model such future increased resolutions, we also consider a 15 Mbps encoding scenario.

3.2.3 Traffic Simulation

Video content is distributed over our envisioned future network topology, in which a small number of Metro/Core nodes aggregate thousands of customers through LR-PON trees with a shared feeder section of up to 100 Km. For the VoD study we modeled the core section of the network on the well-known German backbone topology, with 21 core links interconnecting 14 metro/core nodes (see Fig. 3.2). Each of these nodes is then connected to 1 000 LR-PON trees, with a symmetric upstream/downstream capacity of 10 Gbps shared between the customers of each PON. The number of households served by each PON is a key simulation parameter (we will use the letter p throughout the thesis to indicate this variable); in our VoD experiments we varied it between 16 and 512, with a maximum capacity of around 7 million customers served by the network. As a result, up to 57 million VoD requests were served over a simulated span of 4 weeks.

For the IPTV case, we used a different topology based on the results by Ruffini et al. (2012) – a study attempting to map LR-PON to Ireland by aggregating the Local Exchanges of the major Irish network operator into a minimum number of metro/core nodes. The set of locations coming from that study was then inter-connected with fiber rings, resulting in a network with 27 physical core links interconnecting 20 metro/core nodes (see Fig. 3.3). The number of users per node shown in the figure is the number of customers of the original Irish operator's dataset that are attached to each node as a result of the placement in (Ruffini et al. 2012). These customers are connected to their assigned node through LR-PON trees, with a symmetric upstream/downstream capacity of 10 Gbps shared between the customers of each PON tree. For our experiments, we extract the number of PON trees of each metro/core node by dividing the number of users attached to it by a maximum occupation of 512 users per PON; we can then scale the size of our active user base by varying the number p of users per PON over different simulation instances. In our IPTV experiments we used $64 < p < 512$, with a maximum

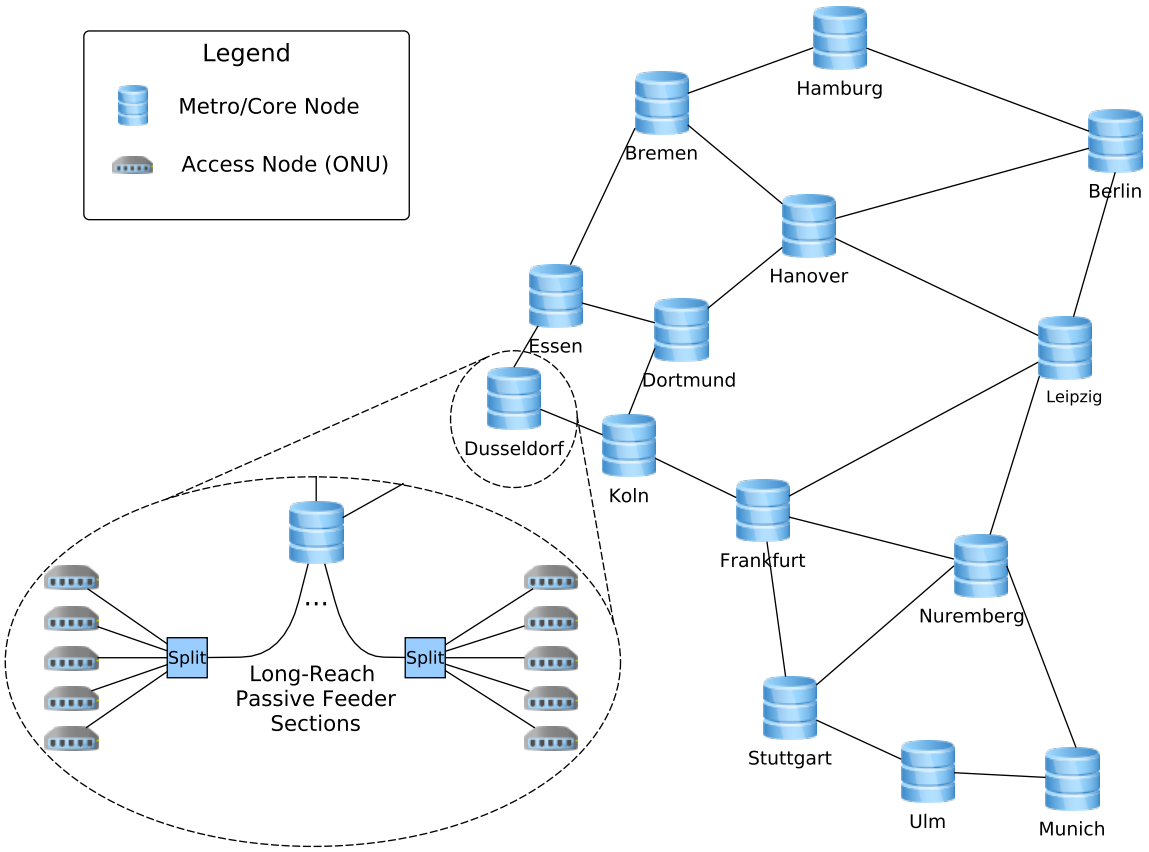

Figure 3.2: The core network topology used in our VoD simulations, based on the German backbone topology, and its expansion through LR-PON trees. Note that the single splitter block in the figure is purely symbolic; in practice, a multi-stage splitter architecture would be used.

capacity of about 2.2 million customers served by the network (i.e., the whole dataset of customers at our disposal). This translates to over 300 million video requests for the busiest IPTV scenarios over a simulated span of 2 weeks.

To simulate such an amount of traffic in a reasonable computational time, a centralized bandwidth sharing algorithm was adopted. Each streaming flow transiting on a particular link is entitled to at least an equal share of the available capacity of that link. This process is repeated across each link of the flow's route to identify the tightest capacity constraint (i.e., the bottleneck), which determines the capacity assigned to the flow. This capacity is updated dynamically whenever a flow enters (respectively leaves) the network; however, only the bottleneck for the new (respectively leaving) flow is used to calculate the new capacity allocation. This is done to alleviate the time complexity of the bandwidth sharing algorithm, at the expense of some loss in terms of optimal resource allocation.

No individual flow is allowed to receive more than 1 Gbps of bandwidth, in an attempt to model both network operators constraints on maximum available capacity per user and the limits imposed by TCP mechanisms (e.g., due to the congestion control mechanisms, as we do not simulate packet loss). Furthermore, if a new request would not be able to achieve a transmission capacity sufficient to

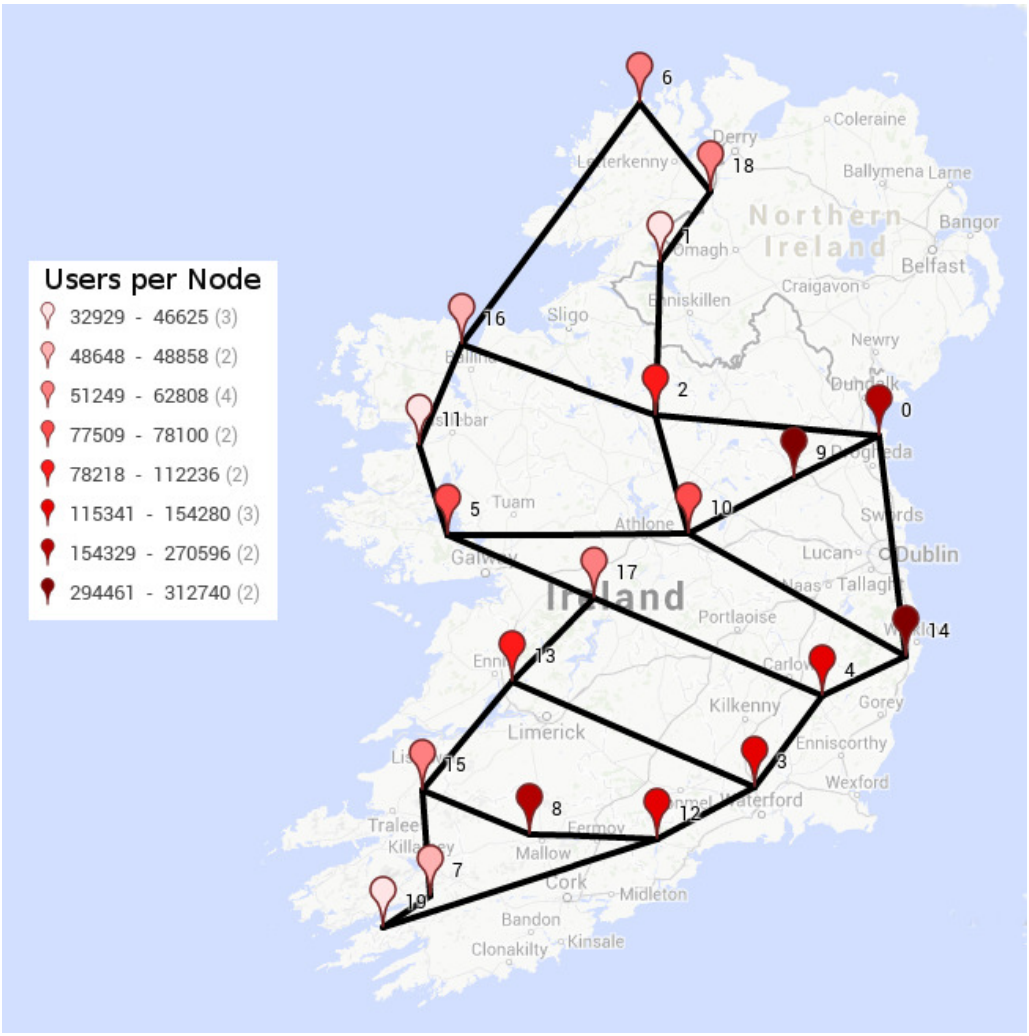

Figure 3.3: The logical core topology for Ireland that was used in our IPTV simulations.

transmit the requested content at its encoded bitrate, the system will attempt to find an alternative, un-congested source for that content. In case an appropriate un-congested source cannot be found, the request will be blocked; however, this never happens in any of the scenarios analyzed in this work.

Each user in the system acts as an element of a distributed cache, storing a certain number of video contents previously downloaded for personal consumption. The dimension of each P2P cache in our scenarios varies between 4 and 16 GB for the VoD study, and is fixed at 16 GB for the IPTV case. Videos are stored in the cache until the replacement policy (Least Frequently Used, or LFU) determines that they need to be deleted to make space for new content. With the exception of the work shown in Chapter 6, no coordination is enforced between the distributed caches; every user simply stores the latest videos they requested.

Content requests generated by users are intercepted by a centralized locality oracle, which will try to redirect them to a local P2P source if possible, i.e., to a user attached to the same metro/core node of the requester; if more than one local source is available, the choice will be made randomly, as a way to

Table 3.2: Simulation Parameters

Parameter	**VoD**	**IPTV**
Number of Metro/Core nodes	14	20
Number of physical core links	21	27
Capacity of a PON	10 Gbps up/down	
Users per PON	16 to 512	64 to 512
Number of video elements	50000	3000 per day, 7-day catalog
Length of a video content (mean)	120 min	45 min
Length of a video content (std. dev.)	15 min	5 min
Encoding bitrate	3 Mbps	5-15 Mbps
Size of an STB cache	4/8/16 GB	16 GB
Size of a CDN cache	16/32 TB	12/24 TB

implement an elementary load-balancing measure. If no such local source can be found, the oracle will look for a non-local P2P source (i.e., a user attached to a different node, thus requiring transit over the core backbone); sources closer to the requesting customer in terms of core hops required are prioritized over more remote sources. Finally, if no suitable P2P source is available, the request is redirected to a central video repository. It is assumed that all the users belong to the same service provider, or equivalently, that multiple providers agreed to share the same caching infrastructure for their catalog.

Using a single centralized oracle for the entire user base, as opposed to having a locality oracle for each of the metro/core nodes, allows for a great simplification of the simulation model. However, this does not significantly affect the obtained results, because the percentage of non-local P2P flows (i.e., between users residing in different access sections) that are made possible by this single database of available sources is negligible – less than 3% of the total, as shown by our exploratory tests. Furthermore, it is possible to achieve the same result by implementing a hierarchy of oracles and coordinators, at the price of an increased complexity in the request serving protocol. The design and implementation of a Locality Oracle service is explored in Chapter 5.

As a benchmark for the efficiency of our system, we study both the performance of a unicast solution (i.e., based on a central repository server for the entire national network) and a distributed CDN caching approach, with dedicated media servers located in each of the metro/core nodes. The size of these media caches varies between 16 and 32 TB for the VoD case; for the IPTV case, we used values of 12/24 TB based on the specifications in (Edgeware 2013). The central repository acts both as the last resort for video requests and as the source for content items that have never been requested before. In all scenarios, the caches of both P2P users and media servers are empty at the start of the simulation, while the central server holds a copy of every element in the catalog.

A recap of the simulation parameters used in our tests can be found in Table 3.2.

Table 3.3: 95% Confidence Intervals for Locality % in IPTV, 10 runs

p	**SD (σ)**	**SE**	**95% CI**
64	0.3144	0.0994	$90.6761\% \pm 0.2215$
128	0.1675	0.0530	$95.4248\% \pm 0.1180$
256	0.1462	0.0462	$97.7544\% \pm 0.1030$

3.2.4 Simulation Error and Confidence Intervals

The results of our simulations campaign are presented in Sections 4.1.2 and 4.1.3. These results are taken from a single simulation instance for each scenario; however, since each run encompasses several days (for IPTV) or weeks (for VoD) of simulated time, and since the results are aggregated and averaged over a very large number of video requests – in the order of the tens of millions – the results obtained with different seeds for the pseudo-random number generator show a very limited variance. For example, running ten simulations with different seeds for our P2P IPTV scenario and for $p = 64$ users per PON resulted in a standard error of 0.0994% for the percentage of locally served requests and a 95% confidence interval of $\pm 0.2215\%$ around the mean value. This interval becomes even smaller for higher values of p, as shown in Table 3.3, due to the larger number of requests generated in those cases. For this reason, the graphs describing our results do not include error bars.

3.2.5 Model Discussion and Limitations

One possible question is whether such a detailed simulation model was actually required for our study. Indeed, as we will show in Chapter 4, the results obtained through the simple steady-state simulator are qualitatively in line with those achieved with the much more complex event-driven tool. However, since our results are entirely based on simulations, and given the relative ease with which a malicious experimenter could manipulate the input parameters of a simulation to obtain favorable outcomes, we wanted to ensure that our model was at the very least defensible under careful scrutiny. Furthermore, some feasibility constraints relative to our proposed P2P strategy can only be enforced by taking into account the full complexity of our target system, i.e., its large scale nature and the time-dependent dynamics of pull-based content caching. Finally, a more detailed simulation model gives us more knobs to operate on, thus allowing us to estimate more precisely the effects of various aspects of the system at hand on the performance of our approach.

Naturally, the opposite objection is also possible: despite our efforts to simulate multimedia content delivery in a realistic way, our model still suffers from some shortcomings. Firstly, a download model is assumed throughout the simulations; streaming traffic has peculiar characteristics (Rao et al. 2011) that are not captured by our simulator. There is certainly a trade-off between the overhead associated with the full download of a partially watched video (i.e., due to zapping) and the traffic savings achieved by using that full replica to serve a future request. Our results show that this second approach still benefits network operators, but further optimization techniques might be employed, such as switching

to a more bandwidth-conservative streaming model when serving over-replicated popular content. For a discussion on the effect of content segmentation (i.e., *chunking*) and adaptive streaming policies on our traffic model, please refer to Sections 7.2.1 and 7.2.2 respectively.

Furthermore, each video content item is assumed to be encoded in a single format, while in real systems the same video (e.g., a TV show episode) would typically be encoded in multiple formats depending on the requirements of the device over which the video will be consumed. For example, a smartphone might only support a lower resolutions stream with stereo audio, while the same movie watched on a 40" TV screen might be requested in full HD and with surround audio tracks. In practical terms, however, this can be modeled in our system as a scenario with an extended multimedia catalog, in which the same video can appear multiple times with different bitrates. The requests for each original single video would be split among this multiple versions, making caching somewhat less efficient. However, running simulations with a doubled catalog size (i.e., 200 IPTV channels) didn't significantly impact the results of this study.

Finally, in this study we consider a pure P2P-based content delivery system using always-on set-top boxes (STBs). In a real world scenario, however, even STBs can be turned off or be subject to faults, and it's conceivable that content providers might want to maintain a stricter control on their infrastructure by using CDN servers in addition to P2P caches. In such a system, some sort of coordination scheme should be devised to ensure that the two layers of caching do not simply overlap. Furthermore, a mechanism to recover from a failure during a file transfer should be implemented, i.e., switching to a different local source or to the central repository in case of a sudden disconnection. However, such occurrences are bound to be relatively rare, given the very fast download speeds that we observe in our simulations (typically less than 10 seconds) and the low churn rates that we expect from Set-Top Boxes. For these reasons, the effect of a reasonably limited rate of STBs abandoning the network should be negligible; indeed our simulations show significant benefits even in scenarios with a small number of active users per PON (e.g., 16-32 out of the maximum 512), which can be seen as an indication of the performance of a system with a higher number of subscribers but a non-zero probability of users leaving the system. Note that an approach such as the one proposed by Chen et al. (2009) for Zebroid cannot be applied straightforwardly to our case since we do not pre-fetch content to STBs and we do not use chunking; however, redundancy could be accounted for in the caching optimization problem described in Chapter 6, i.e., by ensuring that enough replicas are stored in the distributed cache under the expected rate of STBs leaving the system.

4 Analysis of Edge Caching

In this chapter we present the results on bandwidth usage and energy consumption of our proposed P2P multimedia delivery system, obtained through extensive simulations using the tools described in Chapter 3. We first go through the results related to bandwidth efficiency, starting from the steady-state analysis in Section 4.1.1 and then moving to the event-driven studies for VoD (Section 4.1.2) and IPTV (Section 4.1.3). Then we analyze the energy efficiency of our approach, once again starting from the results of the steady-state study in Section 4.2.1 and then exploring the dynamic consumption model developed in conjunction with our event-driven simulations in Section 4.2.2.

4.1 Bandwidth Efficiency

4.1.1 Steady-State Analysis

The results shown here were obtained through the steady-state analyzer described in detail in Section 3.1. As a brief recap, through these simulations we calculate the load in terms of generic traffic units (each equivalent to the bandwidth required to transfer a generic element of the multimedia catalog) imposed at regime on the various sections of the network for different content delivery strategies, such as unicast, CDN and P2P, both with and without locality-awareness. For locality-aware P2P we also consider the case of symmetric access bandwidth. This is done on both a traditional 3-tier hierarchic network with metro rings aggregating traditional PONs, and on an envisioned evolution of this network with LR-PON in the access bypassing the metro section.

Fig. 4.1 shows the results related to our reference 3-tier topology. With the conservative hypothesis of only 1% active customers per access node, localized symmetric P2P significantly outperforms asymmetric local P2P, achieving a 40% reduction in core traffic – almost on par with CDN; on the other hand, higher metro and access network traffic loads make CDN a better solution for such low adoption rates.

However, while client-server, CDN and random P2P strategies all show an approximately linear growth in the estimated traffic load as the number of clients in the system increases, localized P2P techniques scale better than linearly by efficiently leveraging the resources made available by the additional peers in the network. The difference between symmetric and asymmetric bandwidth P2P schemes

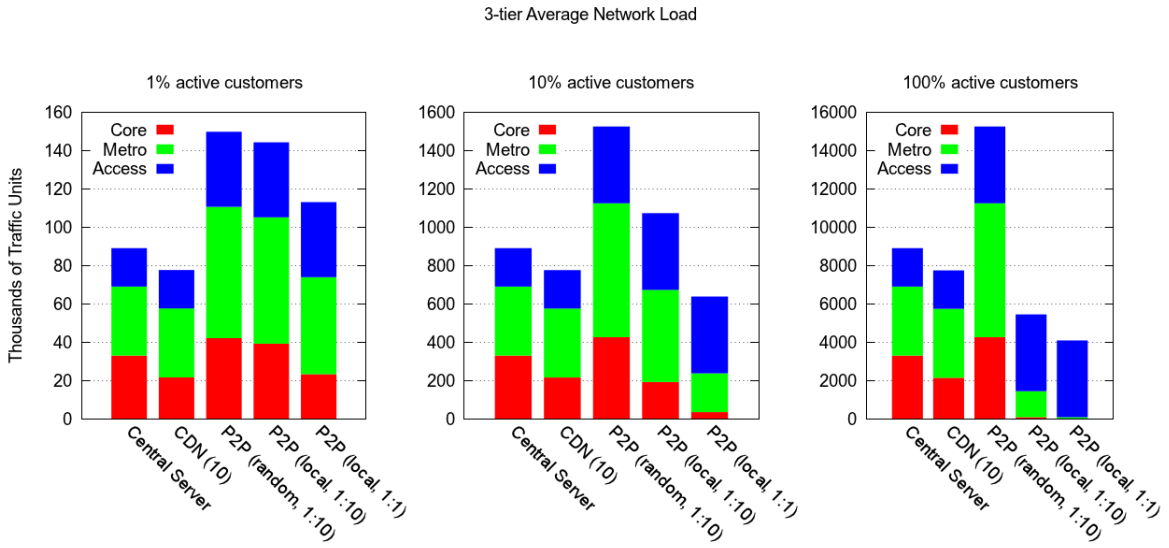

Figure 4.1: Average traffic load with respectively 1% (left), 10% (middle) and 100% (right) active customers per access node in a traditional 3-tier hierarchical topology.

mainly resides in the speed at which those strategies converge towards the optimum scenario, in which all traffic is served locally through nearby peers. Specifically, with a 10% adoption rate, symmetric local P2P already shows better performance than CDN, with a core traffic reduction of 83% and a total traffic load reduction of 18%; asymmetric local P2P, while still offering some savings in the core section, imposes a higher load on the overall network, with a traffic increase of 38% compared to CDN.

Even better results can be achieved in the network-managed scenario, with a 100% adoption rate from the customers: under these assumptions, localized asymmetric P2P generates almost no core traffic, as the probability of finding enough peers with the desired content in the local metro ring approaches 1. Symmetric local P2P further improves these results by eliminating even metro ring traffic and keeping all data exchanges local to the PONs. It is important to stress that this is achieved without enforcing any sort of content-selection algorithm for caching purposes; it is sufficient to store the latest 1-2 elements downloaded by the user from the catalog.

In Fig. 4.2 we present the output of the same simulation set on the LR-PON evolution of the previous network. Consequently, the overall load on the network is reduced (as the average distance between any two nodes is now lower).

Results are in line with those already presented: localized P2P techniques perform better and better as the number of customers in the system increases. However, in this context CDN becomes more competitive, as replica servers deployed in the core are now located much closer to end-users and packets don't have to be routed through the metro ring to reach their destinations. Similarly, the switch from asymmetric to symmetric locality-aware policies becomes slightly less significant – particularly in the network-managed scenario – as the increased customer aggregation factor alleviates the penalties imposed by the higher downloaders:uploaders ratio.

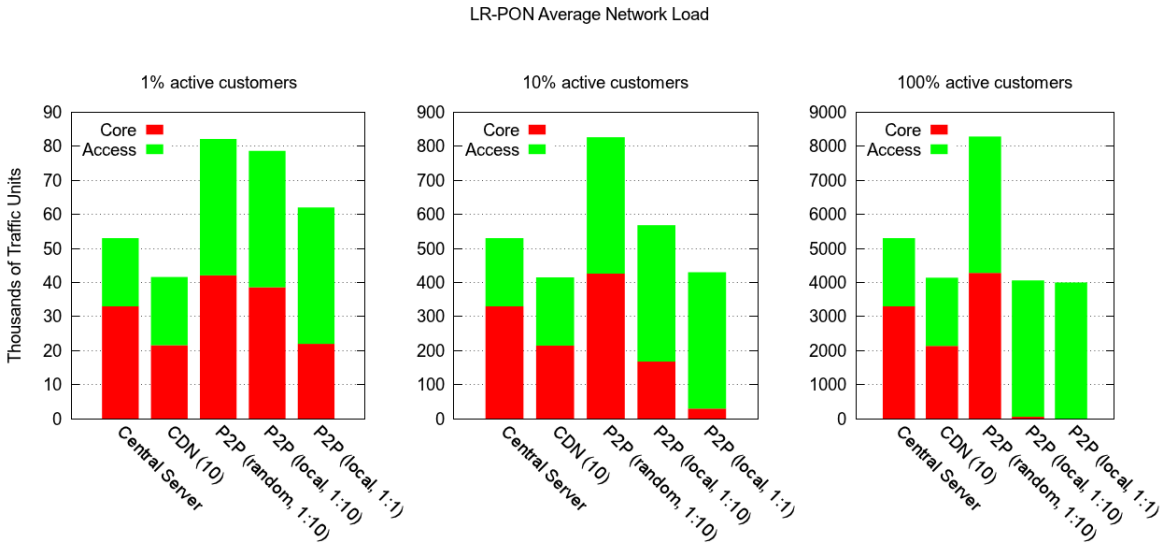

Figure 4.2: Average traffic load with respectively 1% (left), 10% (middle) and 100% (right) active customers per access node in a next-generation network based on LR-PON.

To summarize, in all of the simulated scenarios symmetric locality-aware P2P shows a significant reduction of core/metro traffic and an increase in the load on the upstream channel in the access section.

We argue that reducing core traffic in exchange for access traffic is economically convenient for the network operator for two reasons. Firstly, the access network can be considered a fixed cost. Since the access is required to carry data to the customer premises, its deployment cannot be avoided, and it needs to be provisioned for an adequate amount of capacity. In addition optical access technologies will provide a large increase of available user bandwidth (e.g., with GPON up to two orders of magnitude) for a fixed cost. On the other hand the cost of the core segment of the network is proportional to the capacity offered, which in turn is connected to the amount of data that needs to traverse metro/ core links. By turning traffic around at the access, i.e., before it reaches the core, locality-aware P2P networks can largely reduce the load and thus the cost of the core portion of the network. Secondly, P2P makes large use of upstream bandwidth, which is provided at high speed in optical access networks at no additional cost (1.25 Gbps with a 1:2 ratio for GPON and 10 Gbps with a 1:1 ratio for LR-PON) but is usually under-utilized. By keeping traffic local and turning it around at the first opportunity, P2P locality can exploit this available resource instead of increasing traffic in the metro and core networks.

4.1.2 Event-Driven VoD Analysis

The following sections detail the results of the simulation campaigns conducted with PLACeS, our custom event-driven simulator. Compared to the steady-state analysis, here the focus in on the effect of the dynamic evolution of video popularity and user behavior on the efficiency of a pull-based P2P caching system. Hence, we now only consider our target LR-PON scenario with symmetric bandwidth in the access.

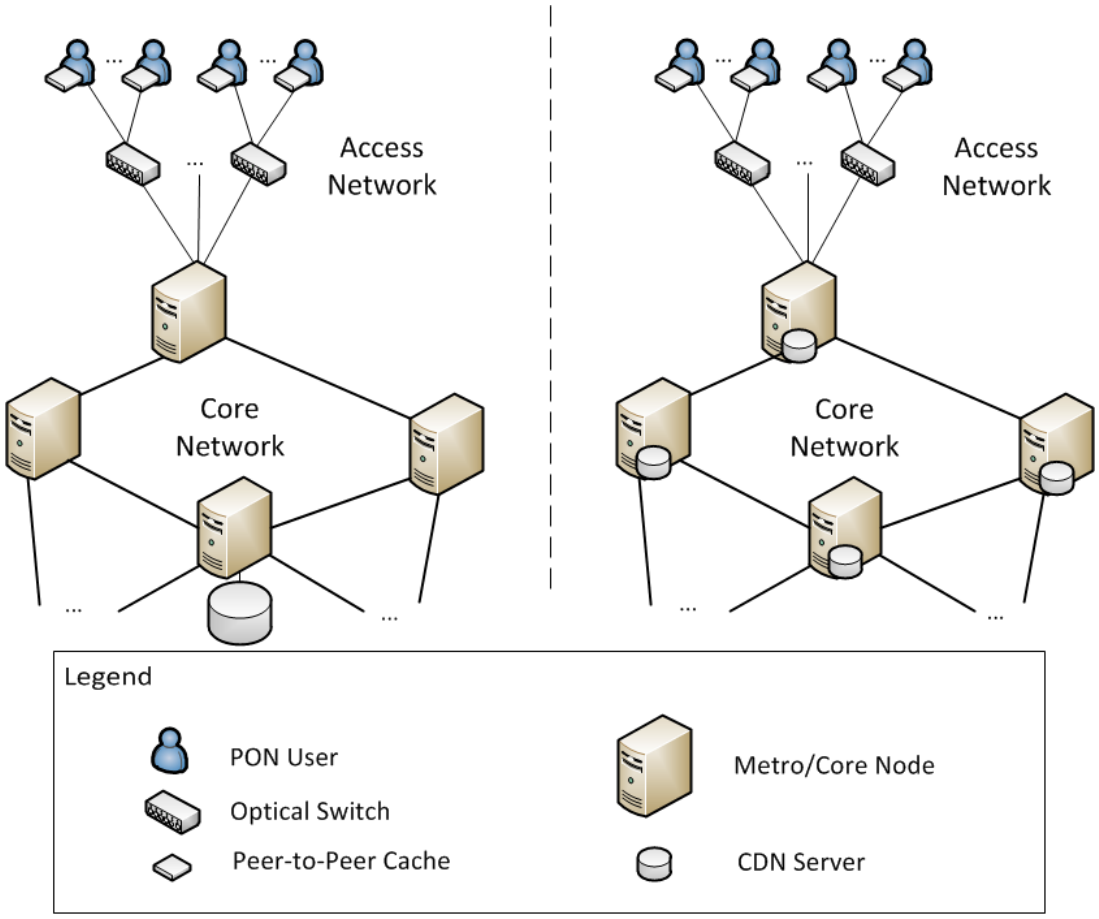

Figure 4.3: (left) The *reduced caching* scenario with a single server for the entire network; (right) The *mixed caching* scenario with CDN caches in each metro/core node. P2P caches are present in both scenarios.

For a recap of the details of the popularity model used for the Video on Demand analysis, please refer to section 3.2.2. As a reminder, each round of simulation in these experiments represents a week in real time.

Once again, as a benchmark for the efficiency of our P2P system we study both the performance of a unicast solution (i.e. based on a central CDN server for the entire national network) and a pure CDN approach, with small servers located in each of the metro/core nodes. Furthermore, to investigate the interactions between our P2P infrastructure and an integrative CDN cache, we designed two different types of P2P scenarios (see Fig. 4.3). In the first, which we refer to as the *reduced caching* scenario, we assume that there's only a single head-server supporting the whole national network; the P2P results presented in Fig. 4.4-4.5 are related to this approach. In the second, which we call the *mixed caching* scenario, we combine P2P caches (8 GB) with the CDN caches (32 TB) in each of the metro/core nodes, to try to keep even more traffic confined to the access section. In this mixed scenario, peers are redirected to their local CDN cache only if there is no viable local P2P source, and to non-local P2P sources only if the content is unavailable at the local CDN cache. The central repository acts both as the last resort for video requests and as the source for previously un-requested contents. In all scenarios,

caches (both P2P and CDN) are initially empty at the start of the simulation, and the central server holds a copy of every element in the catalog.

Our simulations show that the average core traffic on each link is highest with a unicast approach and lowest with a P2P-based approach, even when considering the impact of popularity evolution and the constraint of access links capacity (see Fig. 4.4). P2P remains competitive even with a limited caching size of 4 GB, and it almost nullifies core traffic as the number of users in the system increases. This can be seen even more clearly by looking at the percentage of requests that are served locally (see Fig. 4.5): the percentage increases with the number of users per PON (p) and the size of the P2P caches, up to a whopping 98%.

It's particularly interesting to notice how the percentage of locally served requests for CDN scenarios diminishes over the weeks, while it increases with P2P. This can be explained as the combined result of two factors. Firstly, as more videos get past their peak popularity week, requests tend to shift towards less popular contents. Secondly, as the big CDN caches start to get full, the application of the LFU algorithm tends to favor massively popular content; by contrast, it's easier for at least some P2P caches to retain a copy of less popular videos somewhere in the access section, as the LFU algorithm is applied individually to each of those user storages and users only request a limited number of contents per week.

Furthermore, adding CDN multimedia servers of 32 TB in each metro/core node hardly improves the performance of the P2P caches, to the point where the *reduced caching* and the *mixed caching* P2P scenarios are almost indistinguishable. This can be explained by the preponderance of P2P caching, which is prioritized over non-peering caches in our routing algorithm. For this reason, CDN caches only receive a very limited number of hits even in the most favorable cases, i.e. those in which the small number of users per PON limits the size of the distributed P2P cache (see Fig. 4.6). Unless a different caching strategies is enforced (e.g. replicating less popular content on the CDN and relying on the P2P set-top boxes for the popular content), those micro caches are effectively redundant.

Finally, no content request is ever blocked due to lack of sufficient available bandwidth in any of our scenarios, and the average time needed to download a video element is barely affected by the use of P2P, with the largest additional delay measured with respect to a server-based transfer in the order of 100 ms. In other words, even when relying entirely on P2P for content delivery, the system is always able to find a source with enough bandwidth to satisfy the bitrate requirement of every requester with almost no additional delay compared to a server-based strategy. Note that we calculate transfer times only as a function of bandwidth and not of distance – in other words, data transfers are only limited by the available capacity on the route from source to destination. However, since locality-awareness inherently favors local sources over more distant ones, including data transfer times in the latency computation should not further penalize P2P.

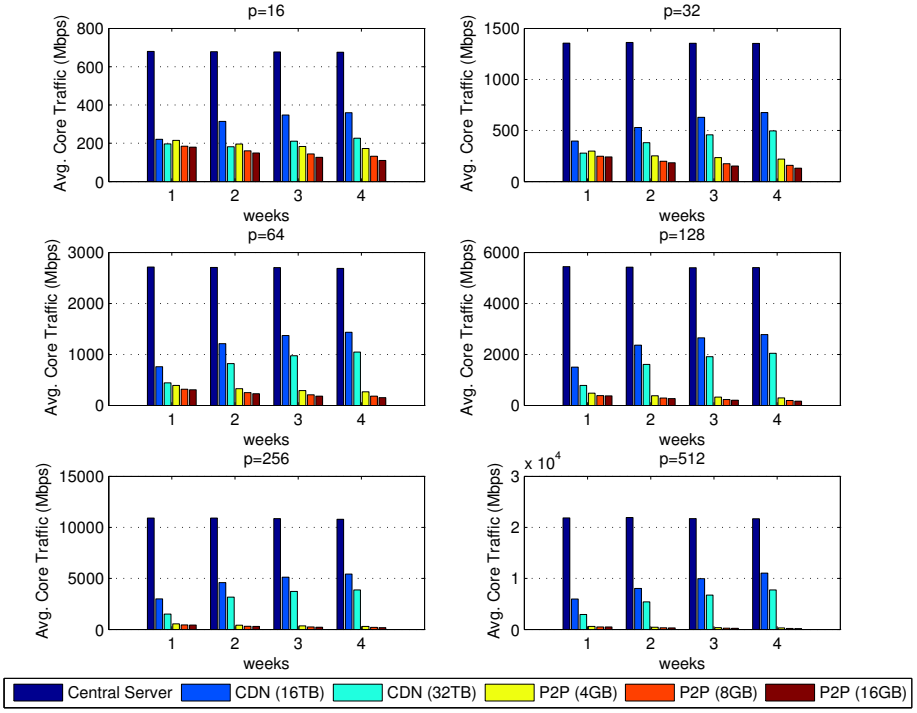

Figure 4.4: Average traffic on core links for increasing values of users per PON (p). The numbers in brackets represent the size of each cache.

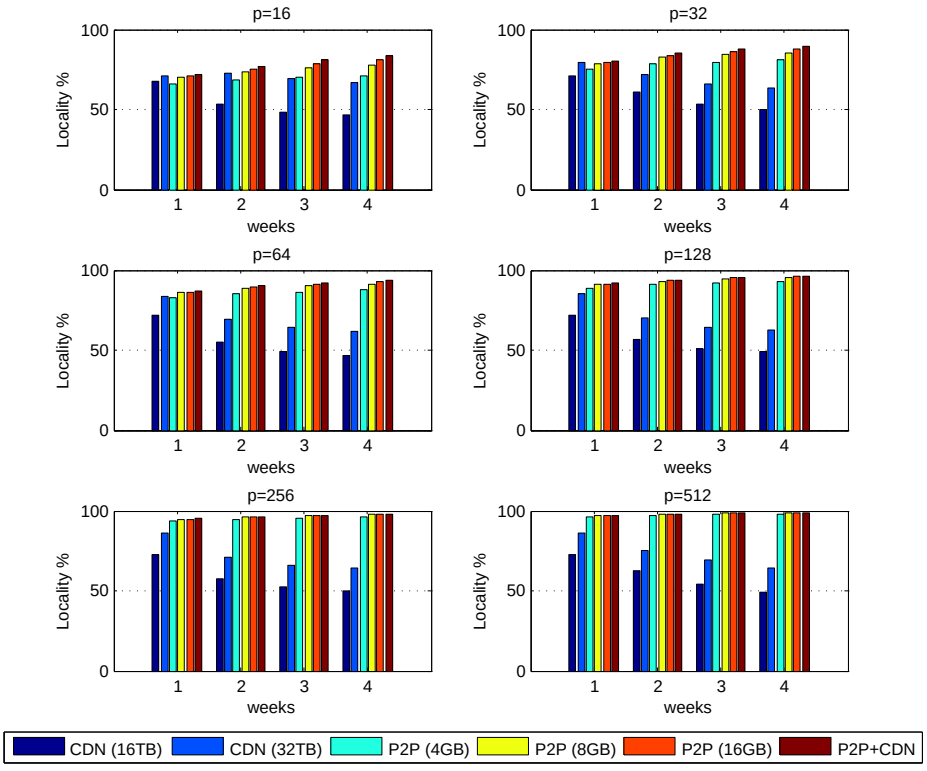

Figure 4.5: Percentage of requests served locally, i.e. in the same access section of the requester, for increasing values of users per PON (p). The central server approach has by definition 0% locality and as such it has not been included.

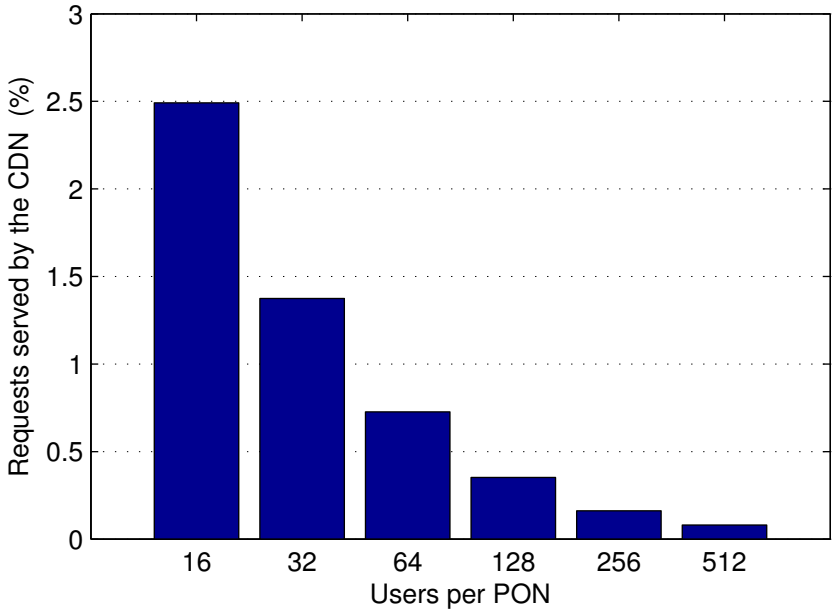

Figure 4.6: Percentage of user requests served by the integrative CDN caches in the *mixed caching* scenario.

4.1.3 Time-Shifted IPTV Analysis

Time-shifted IPTV presents a number of interesting differences compared to the VoD scenario: most notably, as detailed in Section 3.2.2, content in the IPTV catalog is much more volatile, as videos older than a week are retired from the catalog and a number of new elements are released each day. The usage statistic we used for IPTV are also quite more intensive than their VoD counterparts, with the average American household watching as much as 5 hours of TV per day – a number that has been shown to be on the rise. For this reason, a round of simulation for the IPTV case study is equal to just one day, rather than a week as in the VoD case.

Please note that we denote the usual benchmark unicast scenario with the acronym CS (for Centralized Server) and the CDN-based scenario with MS (for Multiple/Multimedia Servers).

We first evaluate the impact of the number of users in the system on the efficiency of P2P caching. Fig. 4.7 shows the average traffic per user generated by each strategy, with the parameter p denoting the number of active subscribers per PON, varying from 64 to 512. Both the CS and MS approaches are not significantly affected by the number of users, and thus are shown only once. P2P, on the other hand, benefits from the increased number of peer caches: when the number of users in the system is sufficiently high, up to 99% of all video requests can be served locally, without leaving the access segment. As a result, core traffic is almost completely removed.

In Fig. 4.8 we show the average generated traffic for the current HD encoding bitrate of 5 Mbps and an estimated future 4k video definition with an encoding of 15 Mbps. Both simulations were run with $p = 256$ subscribers per PON. Even without increasing the portion of users storage allocated to P2P

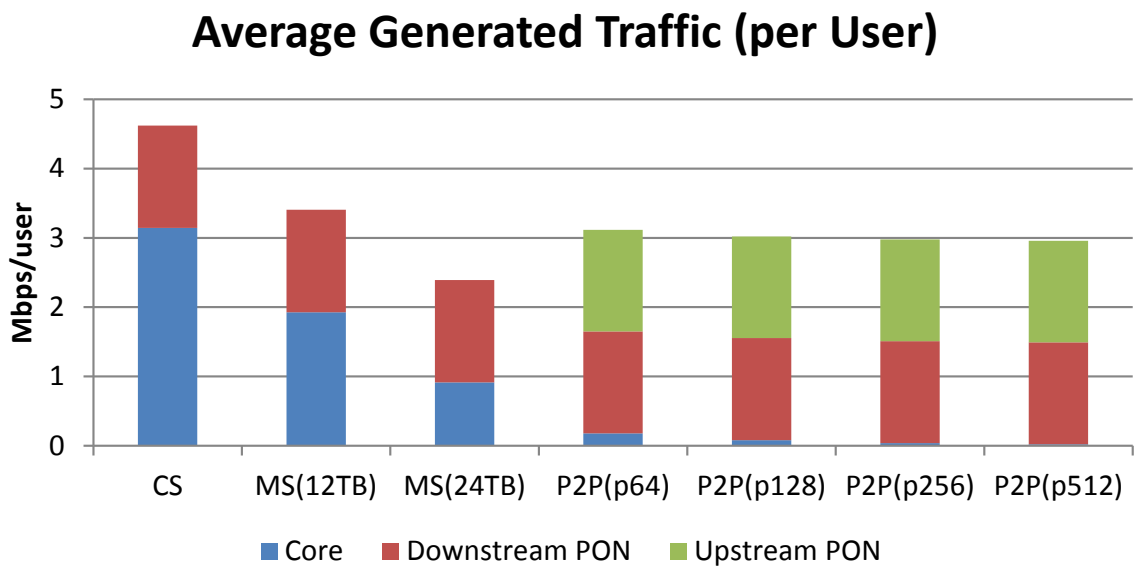

Figure 4.7: Average traffic generated per user by the strategies considered and for increasing values p of P2P users per PON.

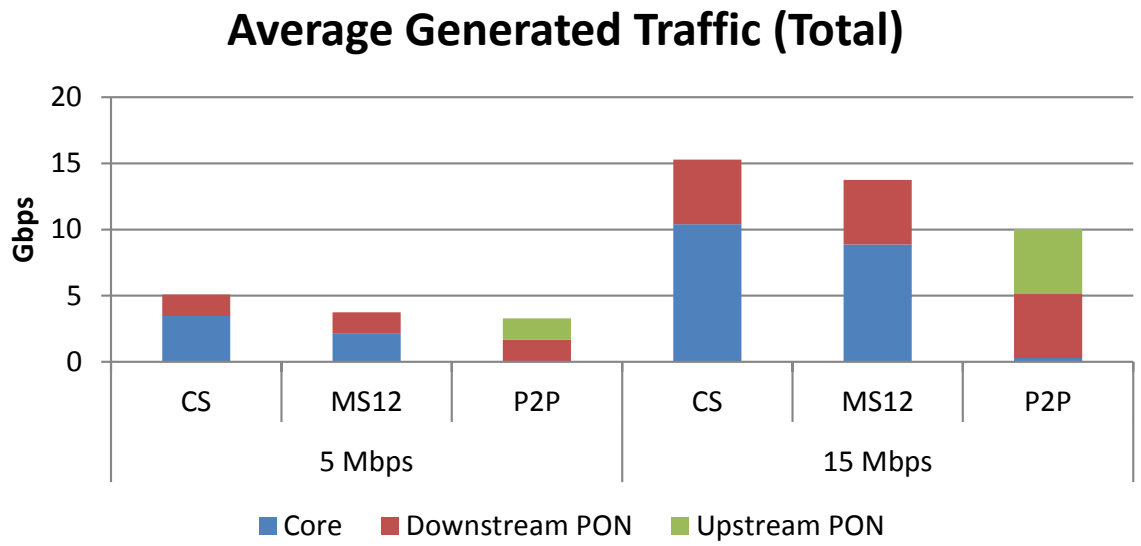

Figure 4.8: Average generated traffic for a current HD bitrate encoding of 5 Mbps and an estimated future 4k encoding bitrate of 15 Mbps.

caching, the reduction in core traffic stays very significant in absolute terms as the encoding bitrate increases. In relative terms, however, the percentage of core traffic generated by P2P compared to the equivalent MS scenario goes from 1.9% with 5 Mbps to 3.3% with 15 Mbps, due to the reduced number of video elements that can be stored in each user cache and the increased effect of storage fragmentation (i.e., many disks with insufficient spare space left to cache an additional item).

MS caching is also a good option to reduce core link utilization compared to a centralized content distribution; our results show that, depending on the size of the MS caches used, it can outperform

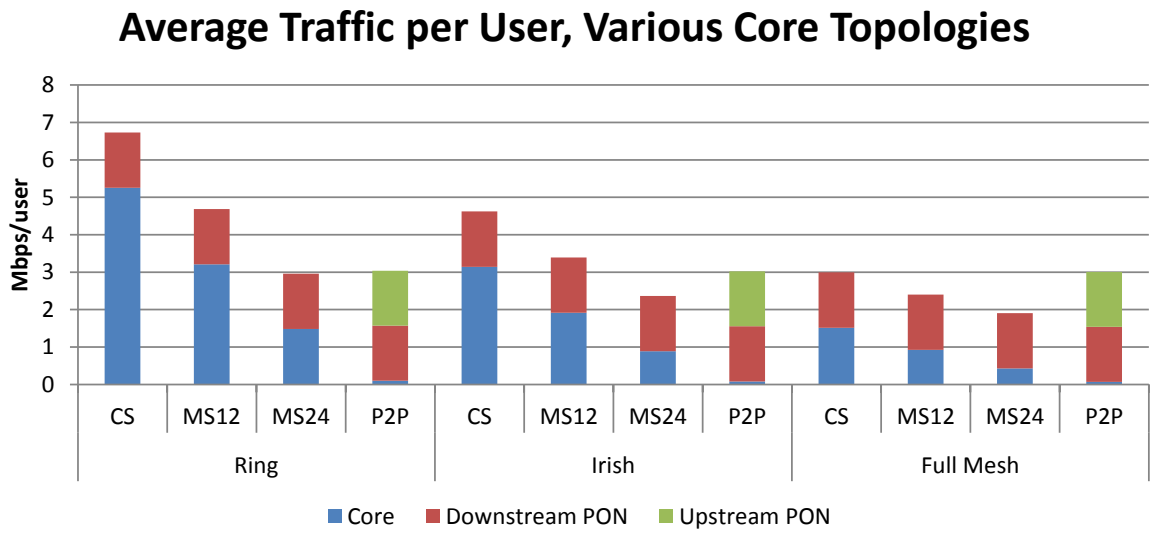

Figure 4.9: Average generated traffic per user over various core topologies, for the IPTV study.

P2P in terms of total generated traffic. However, as indicated in Fig. 4.7, a big portion of this P2P-generated traffic exploits upstream capacity already available in the network, which as such can be considered "free", while the additional core capacity required by MS caching needs to be provisioned, and thus represents a cost for ISPs.

Finally, some simulations over modified versions of the Irish topology were run to investigate the effect of different topologies on the amount of the generated core traffic. Specifically, the two extremes of connectivity were tested, i.e., using a ring topology with 20 bidirectional edges and a full-mesh topology with 380 edges, assuming $p = 128$ users per PON. The average traffic per user generated by the considered content delivery solutions over these topologies is shown in Fig. 4.9. Intuitively, the core traffic generated is inversely proportional to the average hop length from the central server to a generic node, which is a function of the degree distribution of the network (i.e., the number of links departing from each vertex). For this reason, P2P is particularly competitive in sparsely connected networks.

4.1.4 Asymmetric Upstream Capacity

The results presented above mostly focus on systems with symmetric upstream/downstream access capacity; a relevant question could be determining the amount of upstream capacity (or, alternatively, its ratio compared to the downstream capacity available) required for the proposed P2P approach to be effective.

A definitive answer to this question would require extensive simulation campaigns which could be the subject of future work; however there are some basic considerations that can be made depending on the assumptions of the scenario of reference. For example, if we insist on a download model with a single source uploading content items to a requester, than the bare minimum capacity required to avoid re-buffering events is equivalent to the encoding bitrate of the required content. Note however that this

would imply long data transfers during which that particular source would be unable to serve any other request. In such cases it is likely that a P2P system like the one proposed here would suffer, unless steps are taken to mitigate the issue, i.e., adopting chunking and reverting to a traditional model where multiple sources upload different chunks of the requested item in parallel to a customer.

Higher burst rates, on the other hand, would allow us to quickly push content to their destinations, or to multiplex multiple concurrent requests, giving us more leeway to handle a flash crowd scenario. Furthermore, in all of our studies so far we only considered the bandwidth consumption generated by a multimedia streaming service, while in reality such services would compete with the day-to-day Internet usage of the customers, including potential bandwidth intensive applications such as file sharing. Strategies such as throttling file-sharing protocols and prioritizing VoD traffic could be implemented by the ISP to face these issues, but naturally these solutions would affect the customer's perception of the quality of the Internet connection being provided.

4.2 Energy Consumption

In the previous section we have shown that the combination of locality-aware P2P with a pull-based caching strategy is the most bandwidth-efficient strategy for multimedia content delivery. Here we turn our attention to the energy efficiency of the various content delivery strategies considered in our studies. Energy efficiency is a research topic receiving increasing attention in the last decade or so, partly due to the realization that the ICT sector is responsible for a non-negligible percentage of the total CO2 emissions and the increasing effort from governmental bodies and institutions to reduce the impact of man-caused climate change.

More importantly, however, there are economical drivers to reduce the energy consumption of the network. The power consumed by operators to keep their systems running has a direct impact on their operational expenses, in the form of the energy bills that they will need to face. Designing more energy-efficient solutions is hence a way to reduce costs and increase revenues for ISPs.

4.2.1 Steady-State Analysis

As shown in Section 4.1.1, the reduction of core/metro network traffic achieved through locality-aware P2P is counterbalanced by an increase in the access network aggregate traffic, due to the upstream flows originating from the uploading peers. In order to quantify the saving in terms of power consumption that P2P content distribution can achieve, we calculate power consumption values for the worst case scenario – that is, we assume that every active customer is using its entire capacity (e.g., about 20 Mbps for 500 customer sharing a 10 Gbps optical channel) for content distribution purposes, thus saturating access links. The traffic units computed by the stead-state simulator are dimensionless quantities that only give us information on the proportional distribution of traffic across the different sections of the

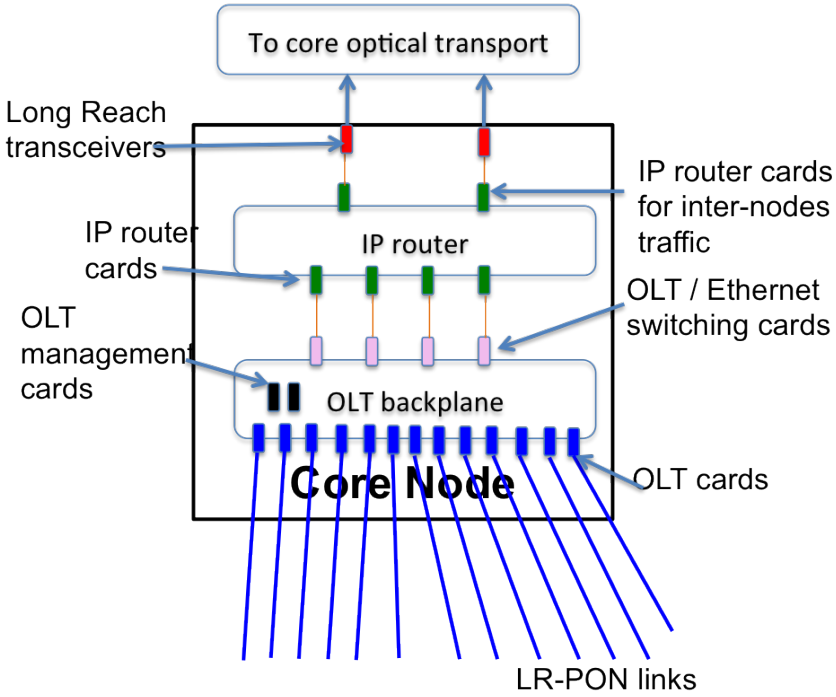

Figure 4.10: Diagram of the network interfaces of a metro/core node.

Table 4.1: Average power consumption of various network components

Component	**Power (W)**
OLT card, 10 Gbps	50
OLT management card (one every 8 OLT cards)	10
OLT switch card (one every 4 OLT cards)	100
Optical Amplifier	10
Long reach transceivers	35
IP line card, 40 Gbps	500
IP router chassis (one every 16 IP line cards)	2 920
IP router fabric (one every 144 IP cards after the first 16)	9 100

considered network topology. Multiplying these generic units by the maximum bandwidth available at each customer gives us an estimate of the maximum traffic imposed on the various portions of the network in each of our target scenarios.

Note that network operators typically use statistical multiplexing and expected average traffic values to dimension their systems; since it is highly unlikely that all of the network customers are going to request their full bandwidth at the same time, the advertised network capacity is typically divided by an "over-subscription" factor before actually allocating resources. However, since we are not interested in accurately modeling ISPs power consumption costs, but rather in comparing the impact of different content delivery techniques on the energy efficiency of the network, applying the same factor of scale to all of these different scenarios would not significantly alter the results presented here.

A power consumption estimation is performed on each of the proposed scenarios. Average power consumption values for various network components were obtained by averaging a number of different

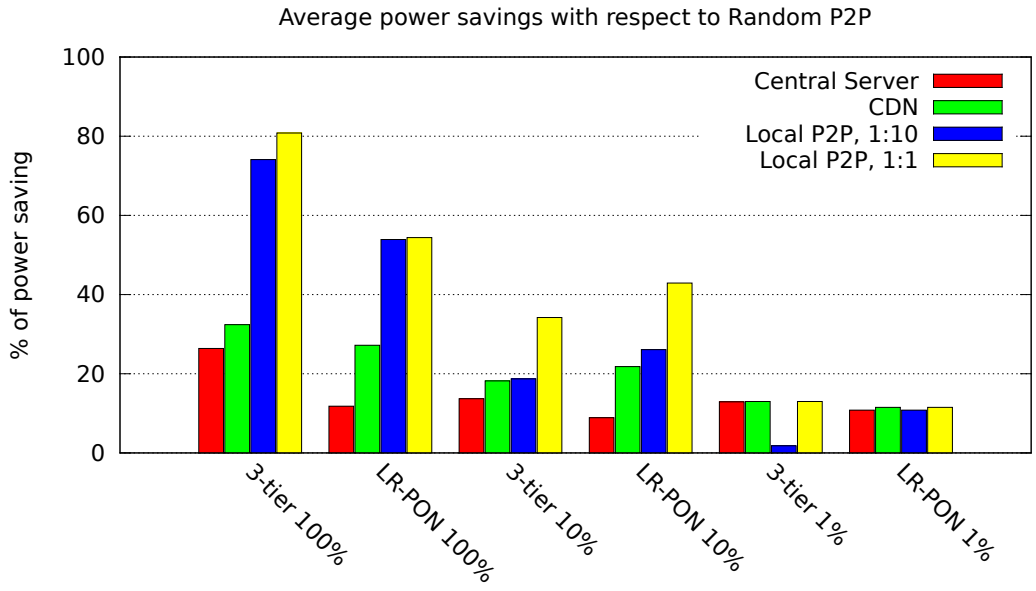

Figure 4.11: Power savings normalized to the consumption figures of a Random P2P scheme in the same scenario, based on the average traffic to be switched at each segment of the network. For each scenario, the topology type (3-tier or LR-PON based) and the percentage of active customers (1%, 10% or 100%) are specified on the horizontal axis.

sources in the literature (Baliga, Ayre, Hinton, Sorin & Tucker (2009), Idzikowski & Wolisz (2009), *IBM Redbooks — BNT Virtual Fabric 10Gb Switch Module for IBM BladeCenter* (2011), *Cisco CRS-1 Carrier Routing System 16-slot Line Card Chassis Specifications* (n.d.)); they are reported in Table 4.1. A diagram showing the network components and interfaces considered for our analysis is shown in Fig. 4.10. We do not consider the costs of back-up links and protection provisioning in our calculations. Furthermore, we assume that metro nodes are able to turn traffic around and to perform IP routing (i.e., as opposed to an architecture in which data would always need to be forwarded to the core node for switching and aggregation purposes).

Fig. 4.11 shows the average power savings that can be achieved with respect to the consumption of a random-selection P2P scheme in the same scenario; this was chosen as the base reference because it generates the highest traffic load under all circumstances, thus qualifying as the most power-hungry of the presented techniques. These values were obtained by calculating the number of network interfaces required to sustain the peak traffic generated in each scenario (as computed by the bandwidth efficiency simulations), and then determining the energy consumption of these interfaces. Note that, while higher amounts of traffic translate to higher energy consumption, not all links are created equal: thus it is not only a matter of how much traffic is generated, but also and foremost of where this traffic flows. As shown in the graph, symmetric local-aware P2P consistently guarantees the highest power savings, with its performance improving as the number of available content in the system increases.

There are some results that do not fall exactly into the overall trend due to the nature of our energy consumption model. In particular, the low difference between asymmetric and symmetric locality-aware

Table 4.2: Power Consumption Parameters

Component	Value	Reference
Metro/Core Cache	85W	Orbit 3020 (Edgeware 2013)
STB SSD Drive	0.49W (i), 1.1W (a)	Corsair F120 (StorageReview.com 2013)
ONU	0.6W (i), 5.9W (a)	10G TWDM (Dixit et al. 2012)
OLT	1.25W/Gbps	XGPON (Skubic et al. 2012)
Ethernet	3.8W/Gbps	Van Heddeghem et al. (2012)
IP/MPLS	10W/Gbps	Van Heddeghem et al. (2012)

Note: (i) and (a) denote idle and active consumption respectively

P2P power consumption in the network-managed case (100% active customers) is due to the coarse granularity of the network equipment selected for our analysis; i.e., as the minimum capacity of an IP line card is 40 Gbps, traffic reductions under that threshold do not translate to power consumption savings. More specifically, the difference between switching 28 Mbps (local 1:1 P2P) and 59 Gbps (local 1:10 P2P) in the core section for the LR-PON 100% scenario is of only 675 W per core node, despite a difference in traffic loads of 3 orders of magnitude.

Similarly, the low efficiency of local 1:10 P2P in the 3-tier, 1% scenario is due to the fact that the limited traffic reduction achieved through asymmetric locality is not sufficient to push the average traffic per metro node under the 10 Gbps threshold that would allow the elimination of one redundant OLT card.

4.2.2 Event-Driven Analysis

As an extension of the results for the steady-state analysis shown above, we developed a dynamic model based on the results of the event-driven simulations, in which we distinguish whenever possible between *active* and *idle* energy consumption. We also separate the fraction of the energy consumed paid by the end-user from the fraction paid by the ISP, in order to identify possible cost-saving opportunities for the operator. This updated model is applied to the time-shifted IPTV scenarios detailed in Section 4.1.3.

Model

The components that have been included in the energy consumption analysis are shown in Fig. 4.12, where the MS Cache blocks represent the multimedia servers used in the distributed CDN scenario. The power consumption values used for each component, along with their references, are shown in Table 4.2. Whenever active/idle power consumption values were not available in the literature, an average power consumption figure was used instead. All the values required to calculate the daily energy consumption of a component, such as the average traffic switched at an interface, the time spent in active mode and idle mode for each device etc., are taken from the simulation results.

More specifically, we assume that set-top boxes include a storage device even when P2P is not used; in other words, we envision an integrated TV+Internet STB, such as those distributed by many operators

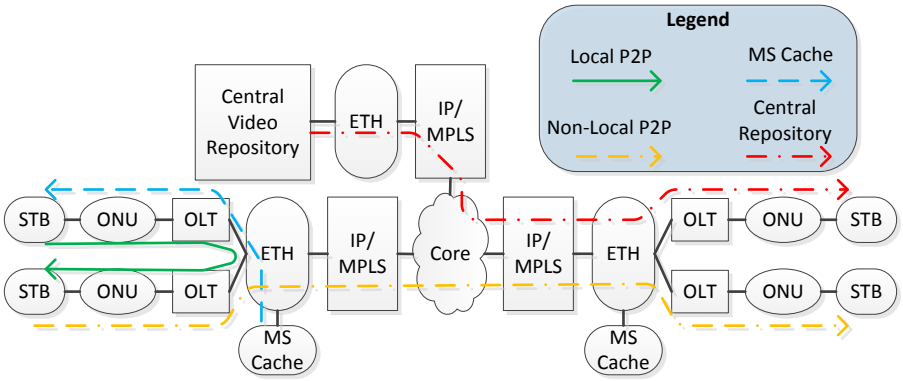

Figure 4.12: Network components affected by each type of data flow in our dynamic energy consumption model.

to support so-called triple play (or lately quadruple play) services. We furthermore assume that the storage device in the set-top box (STB) is in active mode not only when serving or downloading a video element, but also when the user is watching some content. Therefore, the only additional contribution of P2P in terms of power consumption is the time spent in active mode by the set-top boxes uploading contents to other requesters. In our model we make the conservative assumption that this active time does not overlap with the normal video watching time of those uploading users.

The average duration of a video transfer t_{avg} is computed by the simulator based on the effective transmission rates achieved over the network. However, it doesn't take into account transmission delays associated with distance – in other words, data transmission between two endpoints of a link is assumed to be instantaneous and only limited by the available bandwidth (which however is capped to 1 Gbps to account for maximum achievable TCP rates over LR-PON distances).

Based on that value, the time spent in active mode by STBs and optical network units (ONUs) at user premises to download video content ($T_{receive}$) is calculated as

$$T_{receive} = \frac{t_{avg} \cdot R}{U} \tag{4.1}$$

where R is the total number of video requests generated by the users during one day, and U is the total number of active users in the system. In addition to this, for the P2P scenarios we have to take into account the active time required to upload a cached video to a requesting user ($T_{transmit}$). This can be calculated as

$$T_{transmit} = \frac{t_{avg} \cdot R_{P2P}}{U} \tag{4.2}$$

where R_{P2P} is the total number of video requests served by P2P caches in one day. The idle time is just calculated as the total length of a day minus the total computed active time.

For network interfaces, the average traffic values obtained through the simulations are used to compute the capacity to be provisioned at each Metro/Core node. While this hypothesis would not hold in reality, including over-provisioning factors to account for load oscillations wouldn't significantly alter

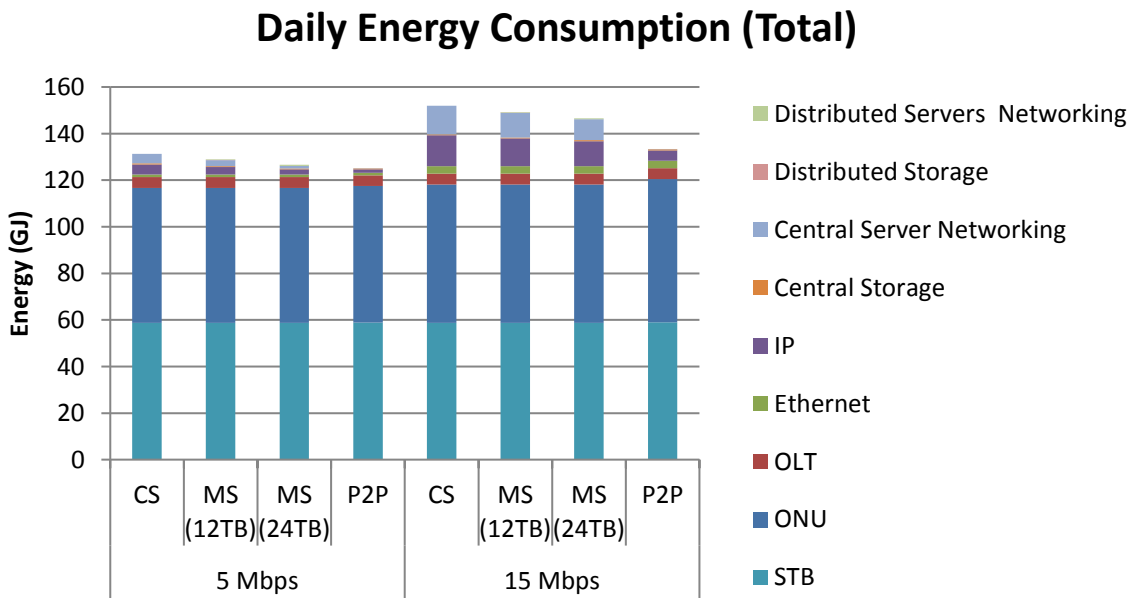

Figure 4.13: Total daily energy consumption, divided by component.

the results as it would equally affect all of the considered distribution strategies. It is also assumed that the capacity on each link must be symmetrical (i.e., full duplex links). Thus, the capacity to be provisioned for each link is equal to the highest value between the downstream and upstream average capacities required on that link, according to the simulations.

To recap, the total daily energy consumption E is calculated as

$$E = (STB_{active} + ONU_{active})(T_{receive} + T_{transmit}) + (STB_{idle} + ONU_{idle})(T_{day} - (T_{receive} + T_{transmit})) + (ETH + IP + OLT + MSC)T_{day} \quad (4.3)$$

where $T_{day} = 86400s$ is the number of seconds in a day, and ETH, IP, OLT, MSC are the power consumption values related respectively to Ethernet switching and IP routing interfaces, optical line terminals (OLTs), and Metro/Core caches, as reported in Table 4.2. Note that MS caches are only used for CDN-based scenarios, and that $T_{transmit} = 0$ for all non-P2P scenarios.

Finally, when determining the cost split between end-users and operators, we assume a Fiber-to-the-Home (FTTH) deployment. Consequently, customers are responsible for the energy consumption costs of the STBs and ONUs, while operators are paying for the remaining part of the electricity bill, including energy consumption of OLTs, network storage devices, routing and switching.

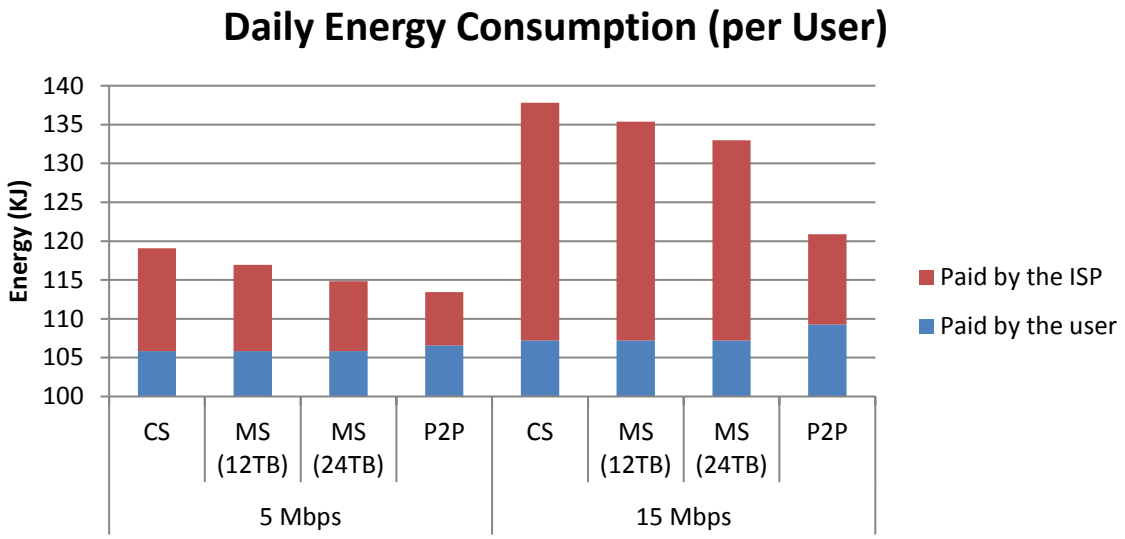

Figure 4.14: Daily energy consumption per user, divided by payee.

Results

Fig. 4.13 shows the total daily energy consumption of the 3 considered strategies for both the 5 Mbps and 15 Mbps scenarios, with 256 subscribers per PON and 5 hours of viewing per day. From the graph it is clear that idle power consumption of STBs and ONUs dominate the total energy figures, mainly due to the large number of devices deployed (one per customer). Active power consumption is largely marginal due to the little time spent in active mode by these devices: e.g., for the 15 Mbps scenario we only have $T_{receive} = 386s$ and $T_{transmit} = 350s$, while the remaining part of the day is spent in idle mode.

Overall the three strategies are closely matched, with P2P gaining some advantage over the others thanks to the much lower data traffic switched from the central video repository into the core network (see Fig. 4.12). This advantage becomes more marked as the encoding bitrate increases. Note that the increased upstream traffic generated by P2P comes almost for free in terms of energy, due to the constraint of full-duplex network interfaces, as downstream traffic dominates upstream traffic. In other words, since in our model all network interfaces are bidirectional and fully symmetric in capacity, the upstream access bandwidth necessary to successfully run these P2P schemes is a free byproduct of the downstream bandwidth required to serve content to users. The only price to pay for P2P is the increased energy consumption of ONUs and STBs during the upload phase, with both values negligible compared to the total energy consumption of these components.

Fig. 4.14 shows the same energy consumption normalized by the number of users in the system, and groups the various components based on who is paying the electricity bill. As discussed above, STBs and ONUs are responsible for the bulk of the total energy consumption, with both devices located in the customers' premises. The increased active time of these devices makes P2P less favorable for end-users;

however the difference is marginal given the preponderance of idle power consumption. On the other hand, thanks to its increased energy-efficiency in terms of routing, switching and network caching, P2P offers interesting cost reduction opportunities for network operators compared to other strategies, with ISP energy consumption for the 15 Mbps P2P scenario equal to just 45% of the equivalent MS strategy and 38% of the CS one.

Finally, simulations on a fully uniform topology (i.e., obtained by equally dividing the total number of users across each of the nodes in the core topology) were conducted in order to assess the impact of user density per Metro/Core node on the energy consumption figures. In this way, the efficiency of P2P is slightly improved, with a 0.25% increase in locality for the uniform scenario and a core traffic decrease per link of less than 0.1%. Intuitively, a uniform distribution of users benefits P2P in the less populated nodes by increasing the pool of potential local sources for any given content; this effect seems to outweigh the corresponding reduction of users in high density areas. For the MS scenarios, a slight increase in locality can be observed for similar reasons. However, core traffic is significantly higher ($\approx 28\%$) compared to the non-uniform case. A similar core traffic increase (29%) can be seen in the CS scenario, due to the higher number of hops that have to be traversed when distributing content from the central server, as users are no longer concentrated in the high density areas close to the server. However, given the very limited impact of traffic switching on the total energy consumption, the effect of these changes is mostly negligible. The most noticeable effect can be observed in the CS scenario, with a mere 0.66% increase in the energy consumed daily per user compared to a uniform user distribution scenario. For this reason, the graph detailing the results of this specific case study was barely distinguishable from Fig. 4.14 and as such was not included here.

5 A Transparent Locality Oracle

Locality-aware policies for content distribution such as those proposed in our work require some sort of oracle service to track content availability at each location and dynamically redirect requests to an optimal source. In this chapter we propose an OpenFlow-based locality oracle, developed as a POX controller module in an attempt to make the redirection mechanism as transparent as possible to the underlying service. We show two possible implementations, using respectively DNS resolution and HTTP redirection, and we analyze the benefits and downsides of each solution. Finally, we evaluate the impact of the redirection process on the latency experienced by the requester.

5.1 Overview

In content delivery systems, *locality-awareness* (or *topology-awareness*) is defined as the ability to choose an optimal data source among a set of potential equivalent sources based on their location, e.g., minimizing the distance between source and destination and/or the cost associated with the transport of the data. This idea has been detailed and analyzed principally in the context of Peer-to-Peer (P2P) systems, to address the problem of inter-AS traffic Dai et al. (2010). Since a typical P2P application does not concern itself with the real network topology on which it operates but only with the virtual mesh of connections between P2P users (the so-called *overlay* network), it can often make poor choices when selecting which peers to contact to retrieve the required data. Locality-aware policies mitigate this issue by informing the peer selection process, e.g., by prioritizing peers that reside in the same AS of the requester.

However, locality-awareness is not specific to P2P systems; in a broader sense, Content Delivery Networks (CDNs) apply the same principle by trying to choose a locally cached copy of the requested content. For example, the Domain Name System (DNS) query for the URL of the host that a customer is trying to contact might be resolved with the IP address of an Edge Server which is closer to the customer's location, based on the IP address of the DNS resolver it is contacting – which is expected to be close to the requester's position.

Content-Centric Networking (CCN) Jacobson et al. (2009) goes a step further by acknowledging that nowadays the Internet is mostly used to request content, rather than to connect to a specific machine:

end users typically want to retrieve a certain web page or video, with little or no care for the specific host from which that content was retrieved. Ideally, we would like to express our interest for a given content element and receive it from the closest (or fastest) possible source; the network should hence use locality-awareness to optimize the resolution of our request.

Regardless of the specific context in which they operate, locality-awareness techniques are typically based on the following two basic primitives:

- tracking the location of the replicas of the data elements in our set of interest (e.g., a Video on Demand service catalog or the directory of an FTP server);
- redirecting requests for content in this set to the most appropriate source, based on some proximity and/or cost metric.

Here we present a possible implementation scheme of a Locality Oracle (LO), a service whose purpose is to implement the two primitives described above. The oracle was developed for the P2P-based multimedia delivery system described in the previous chapters, but it could be used more generically in any system where HTTP requests for content can be served by multiple caches.

Our implementation is based on OpenFlow (OF) (McKeown et al. 2008); more specifically, it has been developed as a POX (McCauley n.d.) controller module. OpenFlow is a key element of our reference network architecture, as described in the DISCUS project (Ruffini et al. 2014). In our vision, the P2P-based content delivery system supported by the LO is owned by the Internet Service Provider (ISP) itself; developing the oracle on top of the OF controller ensures that the source selection process can be integrated in the control plane of the network. For example, if a certain segment of the network is experiencing congestion, the oracle might be able to redirect requests to a less congested but more distant source. Since the LO can be instructed to intercept requests destined to a dynamic set of end hosts configurable at run-time, this strategy could be adopted on-the-fly whenever required (e.g., because of a flash crowd after the release of some popular content) and disabled when it is no longer needed. However, in this work we do not define the policies to be used for the source selection process, and instead we focus on describing the OpenFlow mechanics that can be exploited to implement content request redirection.

5.2 Design Considerations

The Locality Oracle (LO) acts as a proxy for all user requests for multimedia content in the service catalog. As explained in Section 3.2.3, in our simulator we use a single centralized oracle for the entire network. In a real system deployment, however, one might prefer to adopt a distributed approach to avoid a single point of failure and increase the scalability and responsiveness of the service.

In that case, since the main advantage of locality for operators is to reduce the load in the network core, it makes sense to assign an oracle to each metro/core node. When a local source is not available in the originating access section, requests could simply be forwarded to a higher-level coordinator oracle, e.g., responsible for a group of nodes interconnected in one of the all-optical islands envisioned in the DISCUS architecture (Ruffini et al. 2013). If no viable P2P source can be found even in this larger group, the content request should reach the video head-end server to be served by a unicast flow.

Note that the coordinator oracle does not need to keep track of the individual content of each user cache in its domain; for each element of the multimedia catalog, it is sufficient to record the metro/core nodes that advertise at least one cached copy of it, so that future requests can be redirected to the relevant oracle to obtain a viable source address. This helps keeping the system scalable with the number of users and the size of the network.

As mentioned repeatedly, in our work we focus on an *opportunistic caching* scheme, in which users are only assumed to cache content that they required for their own consumption; in other words, we don't consider pre-caching of content in set-top boxes. Whenever a video element is downloaded and cached by a peer, the locality oracle of that metro/core node must hence be notified; similarly, the LO must be informed of any cache replacement operation in order to keep a coherent view of the user cache content. This can be done for example through update messages from the set-top box to the local LO and from the LO to the coordinator, similarly to what is described by Pavlou et al. (2012).

Fig. 5.1 shows possible sequence diagrams for typical interactions between peers and LOs. In Fig. 5.1a, user u1 of metro/core node n1 requests a copy of content c1, which is available locally. The locality oracle replies with the address of a peer source for content c1, e.g. user u2, based on its load balancing policies. User u1 then requests content c1 directly from user u2, which proceeds with the upload. Once the transfer has been completed, user u1 sends an update message to the locality oracle to inform it that content c1 has now been cached in its disk, and that content c3 and c5 had to be overwritten to make space for c1. The locality oracle updates its internal tables, and notices that now there is no other peer in node n1 with a copy of content c5. It then proceeds to inform the oracle coordinator of the removal of content c5 with an additional update message.

In Fig. 5.1b a non-local P2P transfer is shown. When the locality oracle of metro/core node n1 receives a request for content c5, it notices that there is no local copy available, so it forwards the request (along with an identifier of the user who requested it) to the oracle coordinator. The latter identifies a metro/core node which has a cached copy of content c5 and forwards the request message to the relevant locality oracle. From this point on, the communication flow follows the same pattern as the example above. For brevity's sake, we omit the diagram describing what happens when there is no P2P source available; in this case, the coordinator oracle might simply reply with the address of a CDN server with a copy of the requested content.

In this work, we consider a single peer source for each video upload. However, the same mechanism could be used in a scenario with multiple simultaneous peers uploading in parallel. In such a case, the LO

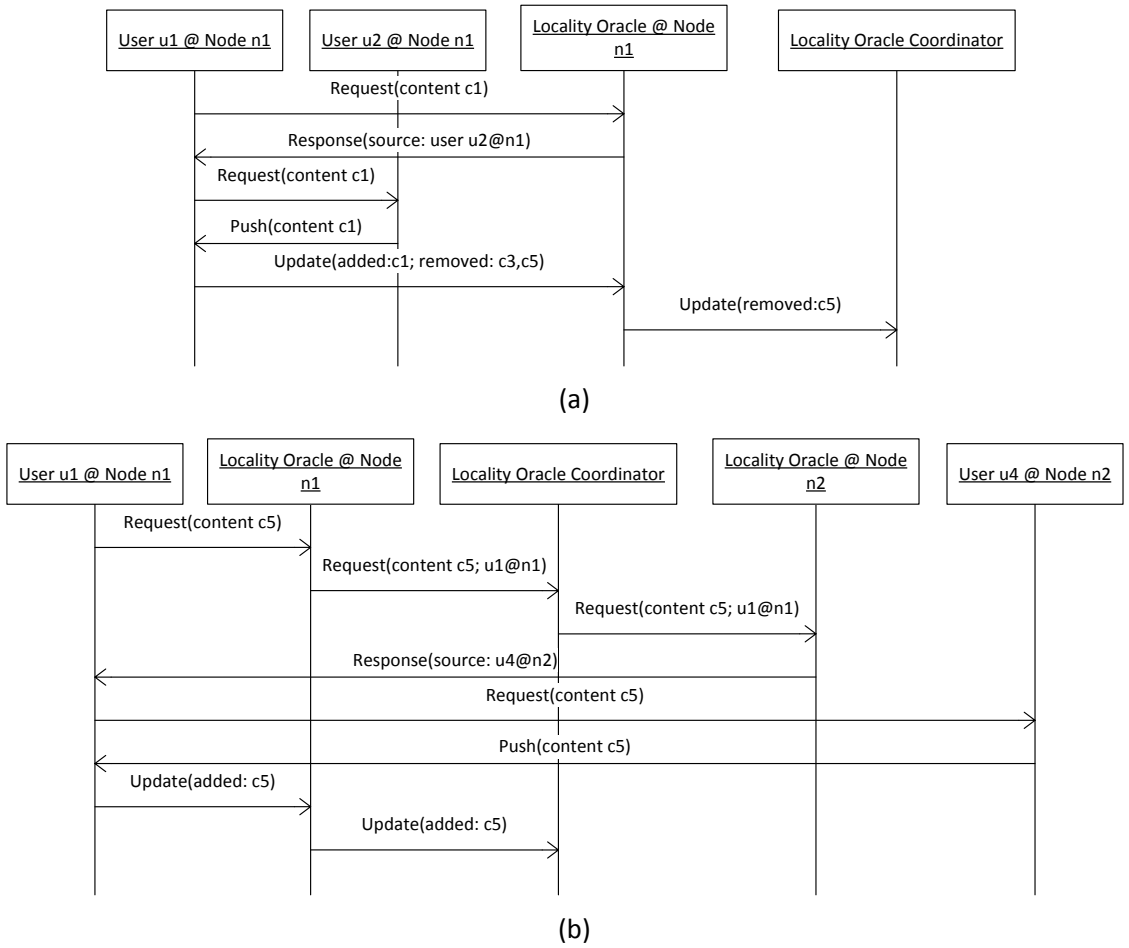

Figure 5.1: Sequence diagrams for a local Peer-to-Peer data flow (a) and a non-local one, i.e. between users connected to different metro/core nodes (b).

would return a list of available sources, possibly ordered with respect to some preference criteria (e.g., load balancing or proximity considerations). The requester would then negotiate with these sources the download of multiple portions (or *chunks*) of the desired video.

5.3 Related Work

In the context of distributed P2P systems, the seminal papers on locality-awareness are arguably those detailing P4P (Xie et al. 2008) and ONO (Choffnes & Bustamante 2008). The original aim of these works was to keep P2P exchanges confined to the Autonomous System of the requester whenever possible, in order to minimize the amount of (costly) inter-AS traffic generated by the P2P applications.

The ideas described in the P4P paper led to the creation of an IETF working group and the eventual definition of the ALTO protocol (Alimi et al. 2013). ALTO details possible coordination mechanisms between ISPs and P2P applications to enable the exchange of locality information and to optimize the P2P source selection process in a mutually beneficial way. Unfortunately, ALTO failed to gain

popularity in real world deployments due to the natural mistrust between P2P developers and network operators, with both parties reluctant to share what they consider sensitive information (Stiemerling & Kiesel 2013). By contrast, in our design the LO is operated directly by the ISP, either in support of content delivery services also owned by the operator itself, or in coordination with a service provider or CDN vendor.

In the context of Content Centric Networking, Nguyen et al. (2013) describe a possible way to implement CCNx (the reference CCN implementation) through a wrapper working on top of an OpenFlow controller. Only the design principles are detailed, but the paper shows that a proof of concept implementation has only a very limited impact on the switching performance of an OF-enabled Pronto switch; this is promising, as their proposal is significantly more complex than our LO implementation.

Similarly, Chanda & Westphal (2013) extend the Software Defined Network model to implement named-based routing and off-path caching on top of OpenFlow. While our aims are somewhat different and more specific than theirs, our approach is based on similar ideas. In our work, however, we attempt to avoid modifications to the OpenFlow protocol and the introduction of a proxy server between the requester and the end host; for this reason, we put the controller module directly in charge of the request redirection process.

5.4 Implementation

In this section we detail the two proposed implementations of the locality oracle (LO) concept, and describe the benefits and limitations of each. While the technical details of the two methods differ, there are common aspects to both. Specifically, we assume that content is requested through HTTP GET messages directed to an end host associated to the service. We also assume the presence of an OpenFlow switch between the access point of the customer and the machine that hosts the service (this is the case, in example, for the DISCUS architecture (Ruffini et al. 2014) that we use as a reference). For the DNS approach, it is further required that the default DNS resolver for home customers is located downstream with respect to the OpenFlow switch – i.e., that DNS requests sent by the customers have to traverse the OF switch to reach the resolver.

All the code required for these implementations was written in Python, and it is freely available at `https://sites.google.com/a/tcd.ie/edipascale/software`.

5.4.1 DNS-based Redirection

The first approach we implemented is based on DNS resolution. This method requires explicit collaboration from the requester, as the name of the desired content must be pre-pended to the base domain name associated to the service. In other words, if the user wanted to retrieve the content named “first” from the host *mydomain.com*, it would attempt to connect to the virtual host *first.mydomain.com*. This

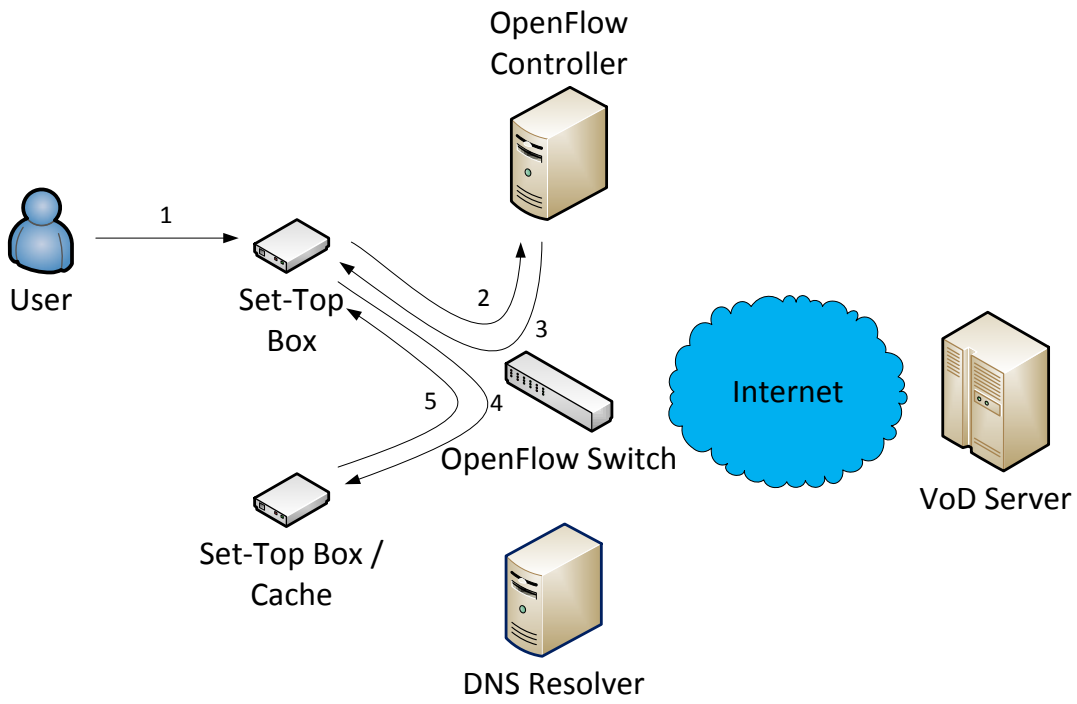

Figure 5.2: Flow diagram of the DNS-based redirection method.

is required because only the host-name is used for the name resolution process (and not, for example, the rest of the URL specifying the exact resource name).

From the controller, our DNS oracle service sets up a flow rule matching all UDP datagrams on port 53 (the default DNS port) and instructs the switches to forward those packets to the controller itself. These DNS requests are then inspected to check whether the base host-name matches one of the hosts associated to the services for which the oracle is responsible; e.g., in the previous example, *mydomain.com*. If that is the case, the oracle extracts the content name from the virtual host-name of the DNS request and uses it to retrieve a list of locally available sources from its internal database. The oracle then chooses one of the sources based on its configured policies, e.g., to ensure load-balancing or minimize usage of expensive connections for the operator, and it returns the chosen source IP address in a DNS response resolving the virtual host-name. The requester can then proceed to contact the chosen source for content retrieval, as if it was the original host machine it was trying to reach.

A diagram detailing a possible use case of this method for our P2P-based multimedia content delivery system is shown in Fig. 5.2. The following steps are depicted:

1. the user requests the desired content;

2. the STB attempts to resolve the virtual host-name associated with the required content by sending a DNS request, which is intercepted by the controller;

3. the controller responds with the IP address of the selected cache, according to some locality policy;
4. the STB establishes a connection with the specified cache and sends a HTTP GET request for the content;
5. the cache pushes the requested content to the STB.

The advantage of this approach is that it is straightforward and does not add significant delay to the request, as the only additional step required compared to a traditional direct download from the end host is the selection of the source in the controller, which is expected to be negligible. Indeed, resolving the DNS request in the controller can even be faster than contacting the designated DNS server, depending on the latency between the requesting client, the OF controller and the default DNS resolver.

On the other hand, this solution requires explicit coordination with the requester, which needs to craft a virtual host DNS request to explicitly indicate the content it is trying to access, thus breaking the transparency of the redirection mechanism. Furthermore, as a DNS query can only return an IP address, it is not possible to specify a target TCP port, which can be a problem when the target of the redirection is a P2P client (e.g., because of firewalls blocking the default port or concurrent applications already using the default port on the target host).

5.4.2 HTTP-based Redirection

The HTTP method was designed to address the lack of transparency and flexibility of the DNS approach described above. The idea is that the user should request a content as if there was no oracle involved – i.e., under our assumptions, through a HTTP GET request to the host-name associated to the service.

Similarly as for the previous example, a flow rule is created in each OpenFlow switch matching TCP packets on port 80 (or any other required port) destined to the IP address associated with the host-name to be contacted for the locality-aware service we are deploying (e.g., the VoD server). These packets are going to be forwarded to the controller and inspected by the oracle to look for HTTP GET requests. If such a request is intercepted, the packet is dropped at the OF switch, and an appropriate HTTP response is crafted by the controller, disguising itself as the host machine that the client was trying to contact. The response will include a redirection message (i.e. code 307 - *Temporary Redirect*) and a Location tag with the IP address and TCP port of the chosen source, selected among the available ones in the same way described for the DNS case. The client will then proceed to contact what it believes to be the correct address for the original request.

Fig. 5.3 describes the application of the HTTP redirection method to our reference P2P system. The steps depicted are the following:

1. the user requests the desired content (the eventual DNS resolution step is not shown);

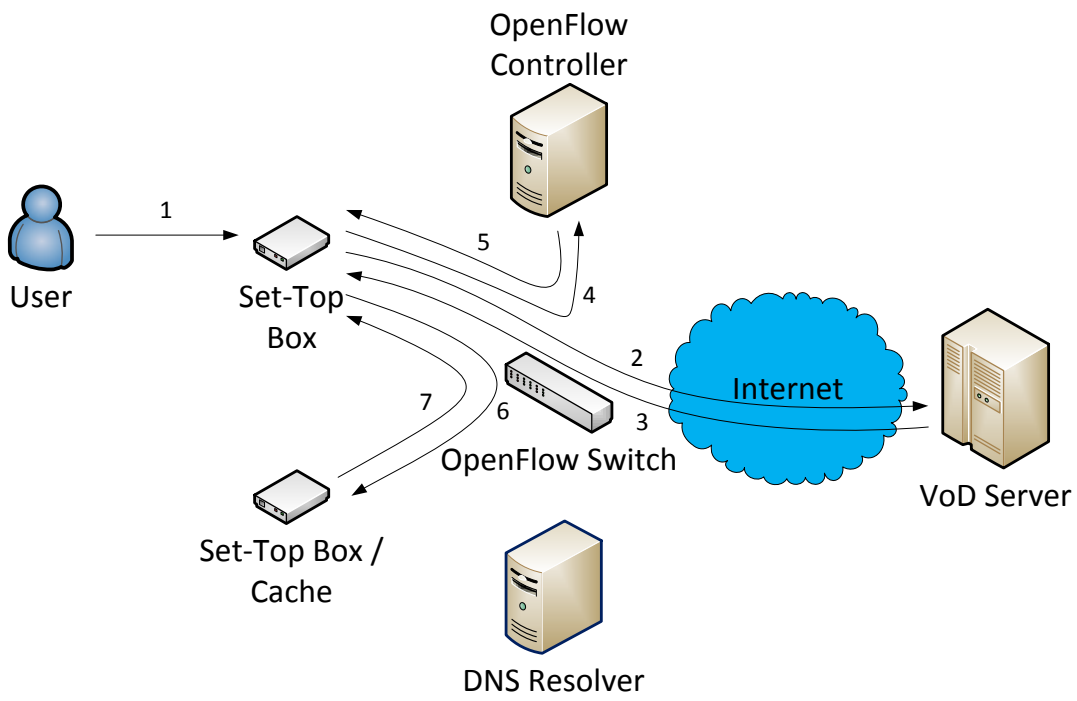

Figure 5.3: Flow diagram of the HTTP-based redirection method. The DNS resolution of the VoD server host-name is not shown for brevity.

2. a SYN packet is sent to the VoD server to establish a TCP session;
3. the VoD server replies with an ACK+SYN packet;
4. the STB completes the 3-way handshake and sends the HTTP GET request;
5. the controller intercepts this request and replies with a HTTP 307 redirect to the cache address;
6. the STB creates a new connection with the cache and requests the content;
7. the content is pushed to the requesting STB by the selected cache.

The transparency of the mechanism allows us to adopt this approach with any kind of HTTP-based content retrieval service, even if they were not designed to be locality-aware. The price to pay is an increase in delay for the requesting client compared to a direct retrieval from the cache, as a full TCP connection has to be established with the end host before the controller can intercept the HTTP GET request and determine what is the content being requested. This also means that resources are briefly allocated on the end host to serve these incoming connections, even though no actual data transfer will take place. Indeed, the server will be waiting for the first ACK of the client in order to complete the three-way handshake, but it will never receive it since it has been meanwhile intercepted by the oracle, thus leaving the TCP connection in an *half-open* state. Until this unusual condition is recognized, the TCP

port assigned by the end host for this particular flow will be unusable by other incoming connections, and memory will be allocated to support the TCP flow (i.e., buffers, session look-up table entry etc.). This issue can be alleviated by explicitly terminating the TCP connection on the VoD server side by sending an RST-flagged packet from the oracle on behalf of the original requester. Furthermore, many techniques have been developed to protect a server from an excessive number of half-open connections, especially since the popularization of Distributed Denial of Service (DDOS) attacks exploiting this idea; for more information on this topic, please refer to (Eddy 2007).

5.4.3 Content Tracking

Half of the responsibilities of a locality oracle revolve around content request redirection, which is what we have just described; the other half is related to content tracking. In order to chose a viable source, the LO needs to be notified of the location of all the cached copies of the contents that are part of the supported catalog.

The simplest way to implement this would be through explicit communications between each of these caches and the LO, as described in Section 5.2. These notification messages would be forwarded to the controller whenever there is a change in the composition of the content that a cache is storing. In our proof of concept implementation this can be done either by directly contacting the LO, if its IP address is known, or with messages sent to the host-name associated with the service, which would then be intercepted by the LO component in the controller in the same way described for content requests. Direct communication is simple and effective, and it is definitely the most efficient way of proceeding when dealing with ISP-managed cache servers. On the other hand, it requires explicit coordination between the caches and the oracle, thus breaking the transparency we were striving for.

We have also implemented an alternative approach that can be used for our locality-aware P2P-based content distribution system, where it is expected that requesters will cache the content they downloaded for their own consumption. In this approach, we exploit the flow-based nature of traffic switching in OpenFlow networks. Specifically, when the oracle suggests a local source for a given content, it can track the data exchange between the suggested source and the requester. When the flow rule created in the OF switch to support this data exchange expires because of an idle timeout (i.e., no data matching the flow is recorded for a configurable amount of time), a *FlowRemoved* event is sent to the controller. On intercepting such an event, the oracle can assume that the exchange has been completed, and hence the requester can be added to the list of viable sources for the given content.

The downside of this solution is that there is no sure way to know if the transfer has actually been successful just by observing the flow expiration: if any connection issue arises, the controller will still eventually register a *FlowRemoved* event and interpret that as an indication of a completed transfer. A better prediction can be achieved by comparing the size of the requested content with the number of bytes transferred over the OF flow, as reported in the *FlowRemoved* event: if the two numbers are

roughly equal (with a small difference accounting for possible re-transmissions of the TCP protocol), it could be read as a sign of a successful download. A more rigorous approach might resolve to attempting to fetch the newly cached content from the user to verify the integrity of the download, but this would obviously impose a more significant overhead on the system.

Finally, an extension to the OpenFlow protocol could be proposed in order to explicitly notify the controller of a successfully completed TCP transaction, i.e., in order to immediately dispose of the relative flow entry in the switching table, rather than waiting for a timeout. While this method would give us reliable information on the progress of a file transfer, it is arguably not sufficiently important to justify a modification to the standard. However, as new features are continuously added to the OF specifications to address new use cases, it is possible that in the future we will have a better way of transparently tracking the status of a TCP connection.

5.4.4 Traffic Tunneling

So far in this thesis we did not really consider the issue of tunneled traffic, as we assumed that the VoD service was being run directly by the ISP or in collaboration with it. Given this assumption, it is reasonable to also assume that the caches in the STBs can be accessed regardless of the specific traffic aggregation strategies applied in the metro/core nodes. However, when considering the transparent application of locality awareness policies to legacy multimedia services (as discussed in this chapter), tunneling becomes once again relevant.

While it is probably safe to assume that any MPLS tunneling or optical aggregation performed by the network operator would only take place after the content request is intercepted by the first OpenFlow switch, any form of tunneling applied directly at the user (such as SSH tunneling to circumvent local firewalls or VPN tunnels) would indeed interfere with the transparent HTTP redirection process described above.

The DNS method would not necessarily be affected, as the user will presumably still attempt to resolve the URL of the resource it is trying to access before encapsulating the request for its transport over the tunnel; however, as explained above, this solution requires explicit cooperation by the end user and thus cannot be considered fully transparent.

5.5 Evaluation

To evaluate the impact of the proposed solutions on the delay perceived by end-users, we deployed the oracle on a virtual topology using Mininet Lantz et al. (2010). Our topology includes a client host (h1) and a P2P cache (h2) residing in the same access section and connected by an OpenFlow-enabled switch (s1) with a latency of 2ms. A second switch (s2) is connected to the first by a link with 20ms of latency,

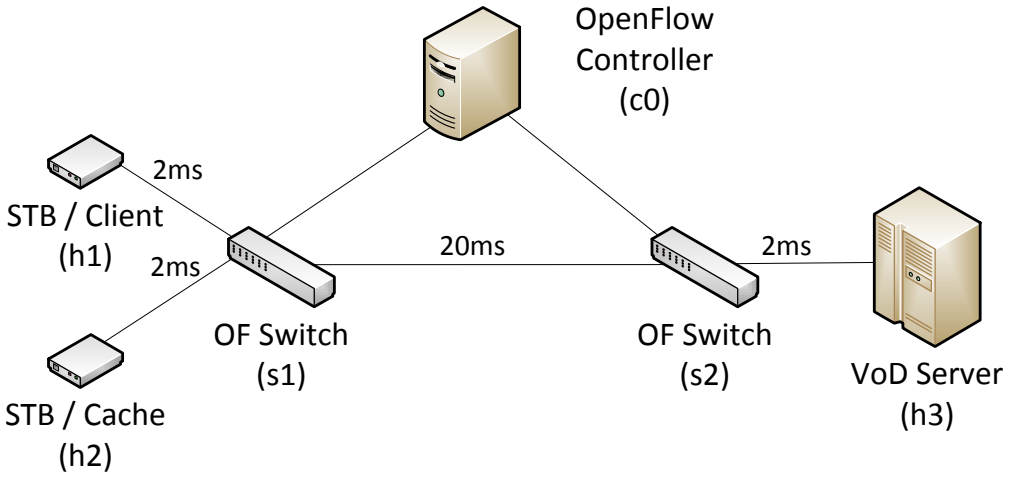

Figure 5.4: Mininet virtual topology used for the evaluation.

to emulate a longer distance from the access; finally, the VoD server (h3) is attached to s2 with a low latency of 2ms (see Fig. 5.4).

A simple HTTP server is running on both the VoD server and the P2P cache, while a python client issues content requests from the client host in h1. Timestamps are taken at each of the following steps:

- immediately before attempting to establish a connection at the client;
- after the HTTP request has been sent by the client;
- when the HTTP request is received by the server;
- when a redirection message is issued at the controller;
- after receiving a redirection message at the client;
- and finally, when a *200 OK* response has been received by the client, signaling the start of the actual download.

A request for a content was issued 50 times and measurement were averaged across these attempts to minimize the impact of fluctuations due to the emulation environment. While the exact delays measured are obviously topology-dependent and thus not particularly significant by themselves, by running these experiments with and without the LO we can assess the impact of redirecting content requests to the controller for inspection. The averaged results are presented in Table 5.1; they focus on the HTTP redirection mechanism, as with the DNS method the VoD server is never contacted and thus the time measurements are not comparable.

Table 5.1: Latency Measurements (in ms)

Step	**HTTP Oracle**	**Direct**
Idle	0	0
GET request sent by client	120.52	111.28
Redirect issued by controller	127.52	.
Redirect received by client	166.28	.
New GET request sent by client	222.20	.
GET received by server/cache	248.08	189.04
200 OK response received by client	291.60	273.00

As can be seen from the results, establishing a TCP connection with the VoD server from the client host (i.e., reaching the state where it's possible to send the actual HTTP GET request) takes approximately 9ms more when the HTTP redirecting LO is active. This time is spent forwarding the SYN packet to the controller for inspection, taking the appropriate forwarding decision (i.e., route it normally) and sending the appropriate instruction back to the switch. Naturally, this delay could increase in topologies where the controller is located further away from the switches.

On top of that, another 7ms elapse before the controller is able to intercept the content request and issue the appropriate redirection message to the requester. Furthermore, additional 39ms are required for the redirection message to be received by the requester, allowing it to establish a new connection with the local cache. In total, the redirection overhead in our experiments totals about 55ms $(39 + 9 + 7)$. In practice, however, the additional amount of time required to initiate the file transfer in the LO scenario amounted to a little less than 19ms on average $(291.60 - 273.00)$, due to the proximity of the local cache compared to the VoD server. In other words, part of the delay introduced by the oracle redirection is compensated by the lower end-to-end latency existing between the P2P source and the client (4ms), compared to latency between the end host and the client (24ms). Please note that we did not take into account the actual data transfer times, which would naturally further favor the redirection scenario for this very same reason.

6 Caching Storage Optimization

In this chapter we present an optimization algorithm to minimize caching storage occupation in a peer-to-peer locality-aware multimedia distribution service. Our proposed solution uses observed historical data on video requests to predict future popularity of the elements in the catalog; the expected request rates computed in this way are then used to guide the caching decisions of the optimization algorithm. Simulation results show that our solution is able to achieve local hit rates that are within 0.5% of those of a traditional Least Frequently Used caching policy, but with a fraction of the storage occupation.

6.1 Overview

In the previous chapters we proposed a peer-to-peer (P2P) based, locality-aware multimedia distribution system for time-shifted IPTV over a next-generation broadband network. We have shown that such a system is able to greatly reduce core network traffic and energy consumption by serving requests locally to the Access Section (AS) of the requesting customer. This is achieved by letting customers cache whatever content they downloaded for their personal consumption (in what we call *opportunistic caching*) and using these local copies to serve future requests. Locality awareness is implemented through a Locality Oracle (LO), a service that keeps track of the content of each user cache in order to match video requests with local sources whenever possible.

In this chapter we aim to keep similarly high percentages of locally-served requests while minimizing the caching storage usage. With the plethora of emerging entertainment services, each often coming with its own Set-Top Box (STB), a war for the control of the customers' living room space has started. Old and new actors are trying to win this competition by presenting themselves as a one-stop shop for all sorts of entertainment services. For example, the latest models of videogaming consoles offer much more than just gaming, acting simultaneously as gateways to video streaming services and videoconferencing programs, or interacting directly with satellite and cable TV STBs. If this trend is to continue, it is likely that several services will have to compete, among other things, for the available storage space on a shared device; in such a scenario, it is advantageous to make the most of the limited resources available.

To achieve this goal, we define an optimization problem that attempts to determine the best caching policy based on the expected rates of requests for the elements of the catalog. Consistently with our

opportunistic caching approach, we do not attempt to pre-fetch content to populate the caches, but rather rely on the natural pattern of user requests. The problem becomes then deciding which of the videos requested by a given user should be kept, and which should instead be deleted. The optimization algorithm is run every time a request for a video element has been completed, i.e., when a new video has been downloaded by a user, and it only affects the content of that user's cache.

Our assumption of a purely pull-based cache greatly simplifies our optimization problem compared to a typical CDN placement scenario, as the set of items that can be cached at each user is limited to the small subset of objects requested by it. We use simple historical aggregate data on content requests to estimate popularity of future content. We do not divide video elements into segments; note however that *chunking* could easily be implemented without significantly affecting the results shown here. In principle, it would be sufficient to treat each content chunk as an independent element of the catalog, with its own popularity estimation. To the best of our knowledge, our work is the first attempt to minimize caching storage space usage in a peer-to-peer based multimedia distribution system.

The rest of the chapter is structured as follows: in Section 6.2 we formalize the optimization problem and define the method used to estimate the popularity of video elements. Section 6.3 describes the simulation experiments we used to evaluate our solution; Section 6.4 details an algorithm to solve our optimization problem and analyzes its computational complexity. Finally, the results are presented in Section 6.5.

6.2 Optimization Model

In this section we are going to illustrate the optimization problem that is solved at each caching decision instance, i.e., each time we need to decide whether or not to cache a content item that has just been downloaded. We first define the variables and parameters that characterize our system, and then we describe the equations that define the constraint optimization problem.

6.2.1 Variables Definition

We index video elements by a unique progressive integer identifier i. The size of element i, i.e., the space it occupies in the storage when cached, is defined as s_i. For each STB b we define:

- $X^b = \{x_i^b \in \{0,1\} : x_i^b \neq 0\}$ where x_i^b is a binary variable indicating whether we are caching content i at STB b. These are the only decision variables of the problem. Note that we say that $i \in X^b \iff x_i^b \in X^b$ or, equivalently, that $i \in X^b \iff x_i^b \neq 0$; in other words, we only store in X^b the non-zero decision variables related to the video elements which are present in the cache at the beginning of the optimization process.

- $Y^b = \{y_i^b \in \mathbb{N}_0 : i \in X^b\}$ where y_i^b is an integer variable stating how many copies of content i are currently being uploaded from STB b to other requesters. Naturally, this is only possible if element i is cached at STB b.

- $|Y^b| = \sum_{i \in X^b} y_i^b$ is the total number of concurrent uploads for all content elements cached at STB b.

- S_{max} is the storage capacity of the STB cache, assumed to be equal for all STBs.

- N_{up} is the maximum number of concurrent uploads that we can support at each STB, due to the upstream bandwidth constraints of the network. Note that this is a simplifying assumption, as in reality, due to the way upstream bandwidth is shared on a PON, this is not a constant and depends instead on the number of active users on each PON and their respective load.

- R_i is the expected number of peak concurrent requests for content i in the current hour in the Access Section (AS) of STB b. Subsection 6.2.3 describes how these request rates are computed; in the context of the optimization problem formulation, they can be seen as input parameters.

6.2.2 Problem Formulation

At each completed request for video content j from Set-Top Box (STB) b, the Locality Oracle (LO) determines if j should be cached and whether some other previously cached contents should be erased. This is achieved by attempting to solve the following optimization problem:

$$min\, f(X^b, j) = \sum_{i \in X^b \cup \{j\}} x_i^b s_i \tag{6.1}$$

subject to:

$$x_i^b \in \{0,1\}; \quad y_i^b \in \mathbb{N}_0 \qquad \forall i \in X^b \cup \{j\} \tag{6.2}$$

$$\sum_{i \in X^b \cup \{j\}} x_i^b s_i \leq S_{max} \tag{6.3}$$

$$y_i^b \leq N_{up}\, x_i^b \qquad \forall i \in X^b \tag{6.4}$$

$$x_i^b(N_{up} - |Y^b| + y_i^b) + \sum_{\substack{c: i \in X^c \\ c \neq b}} (N_{up} - |Y^c| + y_i^c) \geq R_i \qquad \forall i \in X^b \cup \{j\} \tag{6.5}$$

$$|Y^b| = \sum_{i \in X^b} y_i^b \leq N_{up} \tag{6.6}$$

The objective function in Eq. 6.1 simply tries to minimize the total cache storage occupancy, i.e., the sum of the products of the boolean caching variable x_i by the size s_i of the respective content elements.

The remaining equations describe the constraints of our optimization problem. Eq. 6.2 defines the boolean type of the caching decision variables and the integer type of the upload parameters. Eq. 6.3 ensures that the total size of the elements cached (including the requested content j, if we decide to cache it) should not exceed the total capacity of the cache.

Eq. 6.4 states that an element i can be uploaded by STB b to some other user only if it has been previously cached at b, and vice-versa, that an element i cannot be erased from b if it's currently being uploaded to some other customer.

Eq. 6.5 represents the main set of constraints of the problem, whose purpose is to counter the tendency to delete all the elements from the cache in order to minimize the objective function. It states that the total upload slots for content i should be greater or equal to the expected number of peak concurrent requests R_i. The available slots are computed by summing, over all the STBs c that reside in the same access segment of b and which hold a copy of i, the unused upload slots $N_{up} - |Y^c|$, and then discounting those slots that are already being used to meet our target rate for content i (i.e., y_i^c). Naturally the term related to STB b will only be counted if we decide to keep content i in the cache, i.e., if we pick $x_i^b = 1$. This constraint is applied to all video elements present in the cache at the beginning of the optimization problem, and to the additional content j that was just downloaded.

Finally, Eq. 6.6 is purely descriptive, as it does not operate on the decision variables – it simply states that the total number of uploads from an STB cannot exceed its bandwidth capacity. While the values of the parameters y_i^b do affect our optimization model through Eq. 6.4 and 6.5, they are determined by the peer selection algorithm of the Locality Oracle, and as such they are not modifiable by our caching decision process.

6.2.3 Popularity Estimation

From the description of the optimization problem, it should be evident that the correct computation of the appropriate request rates R_i is crucial to the efficiency of our algorithm. Underestimating the number of concurrent requests would lead to an excessive zeal in deleting redundant content, which in turn could saturate the bandwidth capacity of the few remaining sources, leading to lower local hit rates. Overestimating the peak concurrent requests, on the other hand, would impair the ability of our algorithm to reduce the cache occupancy and prune unnecessary replicas.

There are many possible ways to estimate these rates, depending on the assumptions that we are willing to make about our knowledge of the system. For example, it is well known that requests for video elements in a catalog follow a Zipf-like distribution, with the most popular elements receiving the vast majority of the requests (Choi et al. 2012). One could assume that the service provider might have enough historical information to derive the parameters that better fit the popularity curve of the elements in its catalog. Past history of related elements such as previous episodes of a TV show, or the type of video being considered (e.g., movie, documentary, news segment etc.) might be used to increase

the accuracy of the estimation.

However, in this work we decided to only use data which is necessarily available to the Locality Oracle (LO) – that is, the past history of requests for video elements in the catalog. Since every request has to pass through the LO to be matched with an available source, the oracle has a perfect knowledge of the current popularity of video elements as it evolves over time. Videos are kept in a ranking table which moves dynamically as requests are recorded. At the end of each day, the total number of requests for elements of a given rank and a given "age" (in terms of days elapsed since its release) is averaged with the known values for past elements with the same rank and age. For example, if the observed number of daily requests on the n^{th} day of simulation for a content of rank j and released d days before (with $0 \leq d \leq 6$) was $req^n_{j,d}$, then the new estimate for future daily requests of elements of rank j and age d will be

$$daily^n_{j,d} = \frac{req^n_{j,d} + daily^{n-1}_{j,d} \cdot (n-1)}{n} \tag{6.7}$$

However, for the purposes of our algorithm we are interested in calculating the peak rate of concurrent requests that we are going to receive for a given content element i in a given Access Section (AS) in the hour being considered, that is, R_i. If j, d are respectively the rank and age of the content i we are attempting to optimize, then multiplying $daily_{j,d}$ by the percentage of the total requests expected in the current hour h, $hourPctg_h$ (based on typical recorded user activity patterns), and then by the percentage of customers belonging to the AS of the STB b we are considering, $ASPctg_b$, will give us the fraction of daily requests that we expect to see locally in the current hour. Multiplying this value by the average length in hours of a video element, $avgLength$, and dividing it by the length of the interval of time we are considering, i.e., $1h$, will give us the average number of concurrent local requests in the next hour. Finally, to get the peak number of concurrent requests we need to multiply the obtained average value by the expected peak-to-average ratio of concurrent requests, $peakRatio$. Putting all these things together, we obtain the formula:

$$R_i = daily_{j,d} \cdot hourPctg_h \cdot ASPctg_b \cdot avgLength \cdot peakRatio \tag{6.8}$$

Note that on the first day, having no previous information to rely on, we do not attempt to optimize cache content; instead, we cache everything, resorting to traditional eviction policies such as Least Frequently Used (LFU) when necessary. Also note that in our simulations we enforce $R_i \geq 1 \quad \forall i$ both to counter possible approximations to 0 and to ensure that at least one copy of each content is kept in each AS if possible. Furthermore, the effect of using different values for the $peakRatio$ parameter (which we sometime shorten to k for brevity) is described in section 6.5.

6.3 Simulation Methodology

To test the effectiveness of our proposed approach we have used a modified version of our custom event-driven simulator PLACeS. We use the same core topology described in Section 3.2.3 for time-shifted IPTV, based on a real dataset of Irish broadband customers and on the work on optimal metro/core node placement presented by Ruffini et al. (2012). The number p of users per PON used in these simulation experiments takes values in the set $p \in \{64, 128, 256\}$.

The IPTV catalog is modeled exactly as in the time-shifted IPTV experiments described in Section 3.2.2. Note that the maximum number of concurrent uploads per user, N_{up}, is obtained by dividing the total capacity of an LR-PON, i.e., 10Gbps, by the number p of users per PON times the encoding bitrate of the multimedia content; in other words, $N_{up} = 10240/5p$. Note that the total effective available rate on the PON might be lower than 10Gbps when accounting for overhead; however, as mentioned in Section 6.2.2, this fixed capacity model is a simplification which assumes constant maximum usage from all the other users of the PON, and as such it represents a lower bound of the actual capacity a user would see at any given time.

Requests are still generated using the Zipf and Zipf-Mandelbrot distributions described for the IPTV analysis (see Section 2.2.3). However, we now let the parameters of these distributions vary between subsequent days. Specifically, the shift parameter takes integer values in the interval $[0, 50]$, and the exponent parameters take real values in the interval $[0.4, 1]$, based on typical values derived from studies in the literature (Hefeeda & Saleh 2008, Yu et al. 2006). These variation were introduced to make sure that using historical data to estimate future request rates would not be trivialized by having request traces generated from the same distributions every day.

To evaluate the efficiency of the algorithm, we measure the time-weighted average occupancy of the cache at the end of each simulated day. We also measure the percentage of requests that could be served locally to the AS of the requester, as we did in Section 4.1.3. Ideally, we would like to achieve high locality percentages (and consequently low values of core traffic) while keeping cache occupancy to a minimum.

An instance of the optimization problem formulated in Section 6.2.2 is run every time a content has been successfully downloaded by a customer. We invoke CPLEX from our simulator to attempt to solve each optimization instance. Given the small size of the optimization space, the computational complexity is quite low. As a reference, our most computationally intensive test case, i.e., the simulation of a full week of time-shifted IPTV service for the $p = 256$ users per PON scenario (corresponding to over 65.5 million video requests), took a little over 10 hours on a quad-core Intel i7 desktop machine with 8 GB of RAM.

It is important to note that some instances of the optimization problem might not have a feasible solution. For example, we might not have enough residual bandwidth capacity in the system to satisfy

an expected request rate R_i for a content i which has not been sufficiently replicated (e.g., because it has only been released recently). In these cases, the optimization algorithm will fail, and the simulator will default to a positive caching decision, if necessary by using standard replacement policies such as Least Frequently Used (LFU) to make space for element i. In our experiments, we keep track of the percentage of optimization instances that are successful, as shown in the results section.

As a reminder, LFU was chosen over Least Recently Used (LRU) as the benchmark of our optimization algorithm because of the volatility of video elements in time-shifted IPTV services. Since elements older than a week will automatically be purged from all caches when they are erased from the catalog, the main drawback of LFU policies – i.e., old and no longer popular content polluting the cache – is not a real concern here.

6.4 Caching Optimization Algorithm

The model that we have described in Section 6.2 defines a zero-one linear programming problem, as the only decision variables are the binary variables x_i^s and all the constraints are linear. We have found however that a simple greedy algorithm is able to find the optimal solution to the problem in polynomial time, if a feasible solution exists.

The simple idea behind the algorithm is to identify the elements that need to be kept in the cache, either because they are currently being uploaded or because removing them would make us unable to satisfy the expected request rate for that item; everything else can (and should) be deleted to minimize the total cache occupancy. Because each caching decision is taken separately, irrespective of the remaining content of the cache, the complexity of the problem is greatly reduced compared to an exhaustive binary search. The algorithm, which is implemented at the Locality Oracle and run every time a content has been downloaded at an STB, is described in detail below.

Let there be $n-1$ elements in the cache X^b; we have just completed the download of element n, and we need to determine what to cache and what can be evicted. The first for loop iterates over the $n-1$ elements already in the cache, and permanently marks them as elements to be kept if setting the corresponding variable to 0 violates constraint (6.4), i.e., if they are currently being uploaded. Let's say that f_1 variables are permanently set after this loop; in the algorithm, their corresponding indexes are moved in the set F.

The second for loop iterates over all the elements in the cache which have not been set in the previous loop, plus the additional item n which has just been downloaded. For each of these objects, the corresponding decision variable is tentatively set to 0, and constraint (6.5) is evaluated. If the constraint is satisfied, then we do not need to cache that item, and we can fix the corresponding decision variable in F; let's say that f_2 variables are fixed in such way. If, however, constraint (6.5) is not satisfied, we tentatively set the relevant decision variable to 1 and check the constraint again. On

Algorithm 1 Greedy Solution

Require: $X^b = \{1, 2, \ldots, n-1\}$, n downloaded, $F = \emptyset$
  for all $i \in X^b$ **do**
    $x_i^b \leftarrow 0$
    if constraint (6.4) is not satisfied **then**
      $x_i^b \leftarrow 1$
      $F \leftarrow F \cup \{i\}$
    end if
  end for
  for all $i \in X^b \cup \{n\}, i \notin F$ **do**
    $x_i^b \leftarrow 0$
    if constraint (6.5) is satisfied **then**
      $F \leftarrow F \cup \{i\}$
    else
      $x_i^b \leftarrow 1$
      if constraint (6.5) is satisfied **then**
        $F \leftarrow F \cup \{i\}$
      else
        Problem is unfeasible, **exit**
      end if
    end if
  end for
Ensure: $|F| = n$
  if constraint (6.3) is not satisfied **then**
    Problem is unfeasible, **exit**
  end if

a success, the variable has been permanently set and can be added to F; on a failure, there is no way that this particular constraint can be satisfied and hence the problem is unfeasible.

At the end of this second for loop, all the remaining variables should have been set to either 0 or 1, and thus the cardinality of the set F should be equal to n. The only thing left to check is whether the elements that we have identified as mandatory to satisfy constraints (6.4,6.5) actually fit into our limited storage. For this reason, we check constraint (6.3) and, on a success, we can consider the problem solved.

Assuming a feasible solution exists, this algorithm will perform $n-1$ checks of constraint (6.4), after which f_1 variables will be set; $n - f_1$ checks of constraint (6.5), after which f_2 additional variables will be set; and $n - f_1 - f_2$ additional checks of constraint (6.5), which will set all the remaining variables. Finally, a single check of constraint (6.3) will be performed. Assuming that each check has a unitary computational cost and that the assignment operations are negligible, the complexity of this algorithm can be computed as

$$(n-1) + (n-f_1) + (n-f_1-f_2) + 1 = 3n - 2f_1 - f_2 = O(n) \tag{6.9}$$

In the worst case scenario, we have that $f_1 = f_2 = 0$, i.e., no item in the cache is currently being uploaded but all of them (including the element just downloaded) are required to satisfy the expected request rates. In this case, the algorithm performs exactly $3n$ constraint checks to compute the optimal

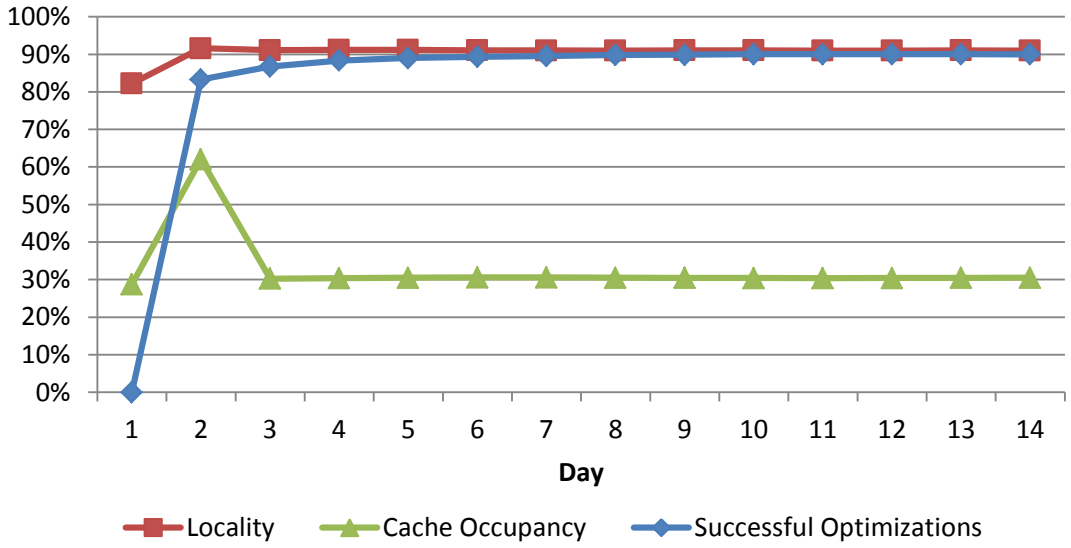

Figure 6.1: Percentage of locally served requests, cache occupancy and successful optimizations over the course of two weeks for a $p = 64, k = 100$ simulation.

solution. On the other hand, in the best case scenario we have $f_1 = n - 1, f_2 = 1$, i.e., all the items in the cache are currently being uploaded and the latest downloaded element is not required to be cached; in this case the algorithm performs $n + 1$ constraint checks in total.

6.5 Results

The first question we had to answer was whether our system would converge to a stable state after a certain amount of time. For this reason we ran some preliminary experiments over two weeks of simulated time, with $p = 64$ users per PON and a value of $k = 100$ for the *peakRatio* parameter introduced in Eq. 6.8. Note that caches are assumed to be empty at the beginning of the simulations, and that on the first day of simulation we do not attempt to optimize caching.

The results are shown in Fig. 6.1. It can be seen that the cache occupancy stabilizes around 30% from day 3 onwards; the peak in day 2 is due to the presence of an almost full cache at the end of the first day, as a result of the simple LFU policy applied at the beginning of the simulation. Note that cache occupancy is measured as a time-weighted average over the whole day; since most of the requests are received during the peak hours of the evening, the average occupancy of day 1 is still relatively low despite the policy of caching every requested content. The other metrics examined also appear to converge after the second day, and the second simulated week adds little or no information to the overall picture; this result also holds for different choices of p and k. For this reason, in all subsequent experiments we limit the length of the simulation to seven days. Furthermore, besides calculating the average cache occupancy across all the days in which the optimization was used, we will also compute

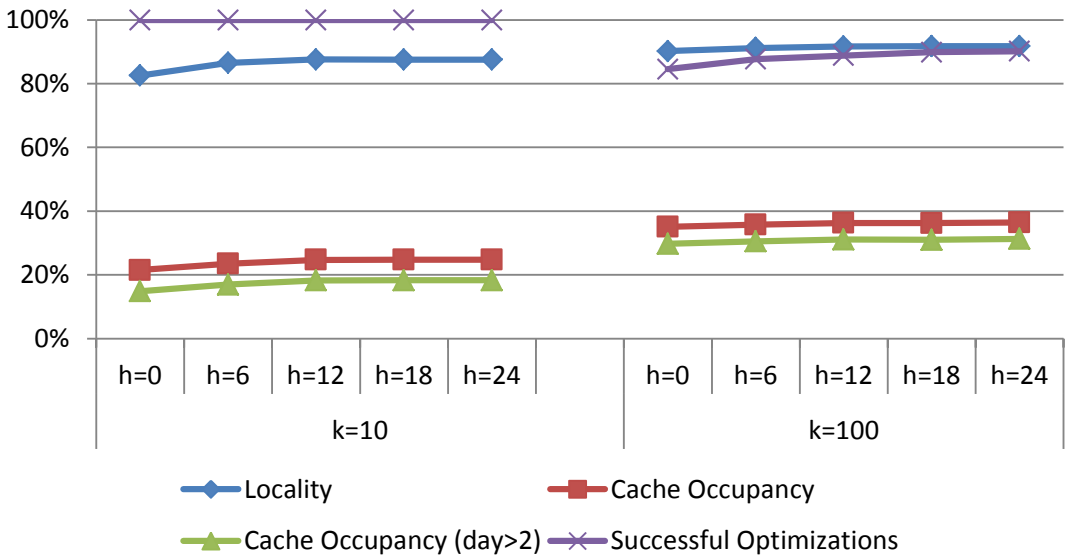

Figure 6.2: Average percentage of locally served requests, cache occupancy and successful optimizations for different values of the protection threshold h for new videos.

the average cache occupancy obtained by excluding the transient peak of the second day, as this metric is closer to the actual occupancy values at regime.

Another question is related to the correctness of the popularity estimation of newly released video elements. Inevitably there is going to be a window of time in which the difference in the number of hits of new videos will be very small and ranking estimation errors will occur. To mitigate this issue we introduce a protection threshold h, i.e., a number of hours that need to pass before a newly released element is allowed to be erased from a cache. In other words, we act conservatively by trying to keep potentially popular content in the cache until we have enough data to take an informed eviction decision. Fig. 6.2 shows the effect of using different values of h, from 0 to 24 hours, with different values of the *peakRatio* parameter k. We can see that the highest increment in locality ($\approx +4\%$) is achieved when going from 0 to 6 hours in the $k = 10$ scenario; obviously this comes at the price of a slightly higher cache occupancy ($\approx +2\%$). Aside from that, the increase in locality brought by higher values of h is typically negligible ($< 1\%$). Results for the $k = 100$ scenario are similar, with higher values of h showing a rather negligible positive effect on locality and a progressive moderate increase of the cache occupancy. In the rest of the experiments we will use $h = 6$.

Finally, in Fig. 6.3 we compare the performance of the optimization algorithm with our non-optimized LFU benchmark, and we evaluate the impact of an increasing number of active customers per PON p on the efficiency of our solution. The graph shows that the optimization is able to achieve locality percentages comparable with those obtained with traditional caching policies, but with a fraction of the cache occupancy.

More specifically, with a *peakRatio* value of $k = 10$ we are able to successfully solve most of the

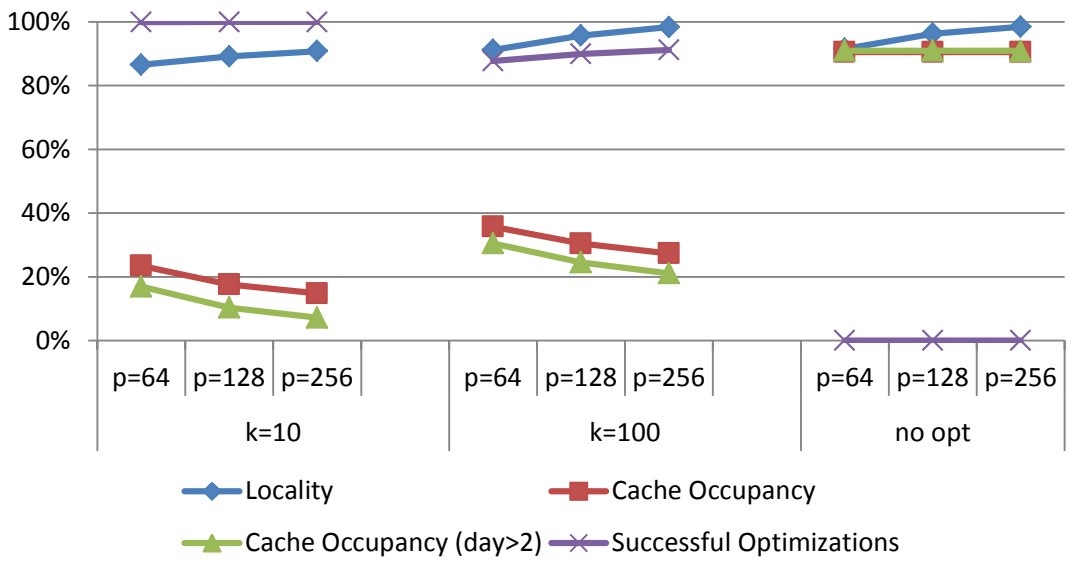

Figure 6.3: Average percentage of locally served requests, cache occupancy and successful optimizations for different values p of users per PON and different *peakRatio* values k.

Table 6.1: Caching Optimization Results

		Locality	**Cache Occupancy**	**Cache Occupancy** ($day > 2$)	**Successful Optimizations**
k=10	p=64	86.54%	23.49%	16.97%	99.94%
	p=128	89.15%	17.63%	10.37%	99.98%
	p=256	90.82%	14.81%	7.19%	99.99%
k=100	p=64	91.17%	35.75%	30.50%	87.70%
	p=128	95.70%	30.46%	24.56%	89.92%
	p=256	98.36%	27.37%	21.09%	91.23%
no opt	p=64	91.58%	90.55%	90.96%	–
	p=128	96.32%	90.51%	90.93%	–
	p=256	98.44%	90.51%	90.91%	–

optimization instances and we achieve a reduction of the cache occupancy between 81% and 92% compared to the non-optimized case, depending on the number of active users. However, the percentage of requests served locally is also between 5% and 8% lower than the values of our benchmark scenario. With a *peakRatio* value of $k = 100$, about a tenth of the optimization instances will fail, resulting in the enforcing of a traditional LFU policy; however, we can achieve locality percentages that differ by our benchmark values by no more than 0.6% and with a reduction of the average cache occupancy ranging between 66% and 77%. More in general, the gap between the locality rates achieved by our optimization algorithm and those of the LFU benchmark decreases with the number of users for $k = 100$.

A recap of the results achieved is presented in Table 6.1.

It is worth discussing further the implications of the *peakRatio* parameter. In essence, it tells us what is the ratio of concurrent requests at peak time, compared to the daily average. High values of this parameter indicate a large concentration of requests during the busiest hours of the day; however, even

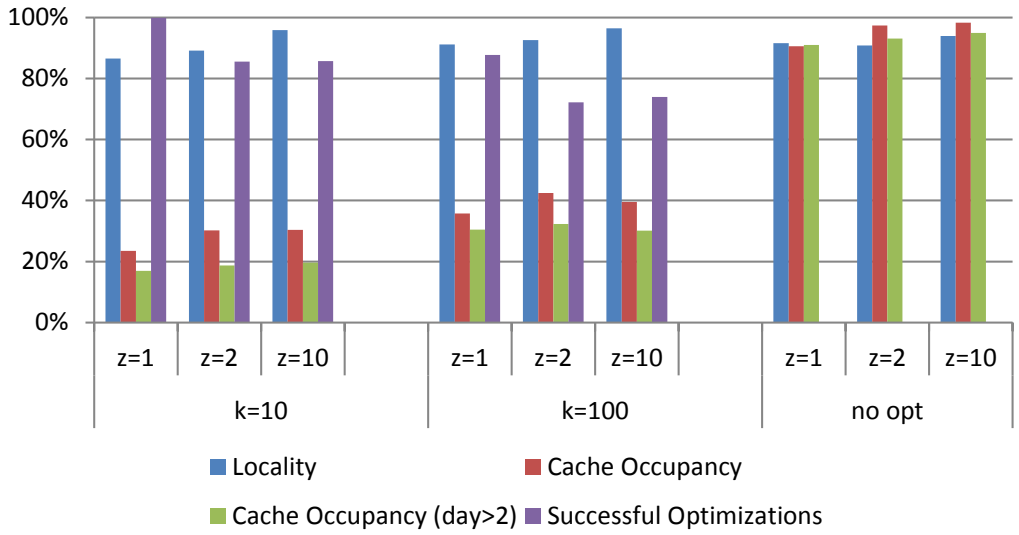

Figure 6.4: Average percentage of locally served requests, cache occupancy and successful optimizations for different values z of the Zipf distribution exponent and different *peakRatio* values k.

considering the observed increase in video requests during the evening hours, a 100x increase compared to the average seems rather steep. Another aspect to be kept in consideration is that in our experiments there did not seem to be much difference between a *peakRatio* value of 10 or 50, but a significant one between 10 and 100, as reported above. This might have to do with the way the rate of requests R_i is calculated (i.e., because of decimal approximations in the simulator).

Finally, we asked ourselves whether the performance of the caching algorithm would deteriorate with stronger variations in the popularity patterns, i.e., by increasing the range of the interval from which we extract the exponent of the Zipf and Zipf-Mandelbrot distributions (see Section 2.2.3). In Fig. 6.4 we show the results obtained for different maximum values z of the exponent; note that we switched to a histogram layout to reduce the clutter of overlapping points from different series. The results show that the performance of the algorithm in terms of locality actually increase if we allow higher exponents to be chosen; this is not too surprising, as a higher value of z means that a higher number of requests will be directed to the most popular items, increasing the efficiency of caching. Indeed, our algorithm achieves even higher levels of locality than the non-optimized LFU case, with a maximum increase of 2.65% for $k = 100, z = 10$. As for storage usage, it seems to increase with higher values of z, with a maximum increase of +29.38% for the $k = 10, z = 10$ case; however findings are inconclusive, as the trend is not monotonic and the cache occupancy after day 2 is showing a much smaller variation.

7 Conclusions and Future Work

In this final chapter, we give a brief recap of the main contributions of our thesis, discuss the limitations of our study, and propose further avenues of investigation for the future.

7.1 Contributions of this Thesis

7.1.1 Development of an Open-Source Simulator

Chapter 3 details the custom simulation tools that we created to evaluate various content delivery strategies. More specifically, after a preliminary steady-state analysis, we developed PLACeS, an event-driven, flow-based simulator for locality-aware peer-to-peer content delivery over next-generation networks. The simulator, which has been released as open source software, is highly customizable in terms of the multimedia popularity model and its related parameters, the network topology used, and customer placement and behavior. A simplified centralized bandwidth sharing algorithm for data flows allows the simulator to scale to very large number of users (in the order of magnitude of millions) in reasonable simulation times (in the order of magnitude of hours). Results are collected by the simulator on both a per-round and aggregate basis, and printed to screen and to file. Furthermore, it is possible to export graphml snapshots detailing the state of the network at a given instant in time, as an additional way to visualize the results of the simulations.

7.1.2 Analysis of the Efficiency of Edge Caching

Bandwidth Efficiency

Extensive simulation campaigns showed that locality-aware peer-to-peer strategies are more efficient than CDN or unicast solutions in terms of core bandwidth utilization, thanks to their ability to keep most of the traffic exchanges local to the access node of the requesting customer. This result is valid not only for the steady-state analysis presented in Section 4.1.1, but also for the event-driven Video on Demand study in Section 4.1.2 and for the time-shifted IPTV study presented in Section 4.1.3. As could be expected, in all these studies the efficiency of our proposed scheme is shown to be proportional

to the fraction of users taking part in the distributed P2P cache; however, significant savings can be achieved even in the most conservative cases.

We showed that using a CDN cache layer in addition to the pull-based P2P caches does not provide significant benefits for the VoD scenario in the absence of some cooperative mechanism, which is required to ensure that the CDN caches are not merely replicating the content already available on the customers' Set Top Boxes. We also investigated the effects of higher encoding bitrates for time-shifted IPTV, showing that similar core bandwidth savings can be achieved even when moving to an envisioned 4k ultra high definition service.

Finally, we investigated the impact on our findings of different degrees of connectivity in the network topology; locality-aware P2P is shown to be more beneficial over sparse networks, where a local cache miss implies fetching the content from a CDN server potentially several hops away from its intended destination.

Energy Efficiency

In Sections 4.2.1 and 4.2.2 we evaluated the energy efficiency of locality-aware P2P compared to other traditional content delivery strategies, such as unicast and CDN. We showed that the bandwidth savings reported in the previous section can translate to energy savings, due to the reduced number of network interfaces required to serve the multimedia traffic. The increased usage of upstream access bandwidth brought by P2P, on the other hand, represents a fixed cost and does not have much effect on the total energy balance.

Besides reducing the overall energy consumption, locality-aware P2P is also particularly effective at reducing the energy quota paid by the ISP, thus potentially enabling cost-saving opportunities for network operators in terms of reduced energy bills.

7.1.3 Design of a Locality Oracle

In Chapter 5 we presented an OpenFlow-based implementation of a Locality Oracle. In our approach, requests for multimedia contents are intercepted by a POX controller module and redirected to a desired local source, either by resolving specially crafted DNS requests or transparently by issuing HTTP redirects in response to GET requests. The DNS approach does not introduce any additional delay and is more efficient in terms of network resources management, but it requires the explicit collaboration of the requesting user. The HTTP solution is completely transparent, but it introduces additional delay in the processing of the request, as a TCP session with the end server must be established before the content request can be intercepted. We have also detailed a mechanism to track content transfers based on the expiration of OpenFlow flows; while this method can only make educate guesses, it does provide a transparent way to monitor the availability of local sources as a result of content transfers. Finally, we

have deployed these oracles on a virtual topology based on Mininet, and evaluated the latency overhead generated by the HTTP redirection method compared to a direct connection with the VoD server.

Besides showcasing the full potential of SDN beyond simple control plane management, developing the proposed strategies in OpenFlow gives us additional flexibility; in example, an ISP could dynamically deploy redirection in response to changes in the network (e.g., to handle a flash crowd) even if the HTTP-based service was never designed to support redirection.

7.1.4 Optimization of Caching Storage

In Chapter 6 we define an optimization algorithm to minimize the cache occupancy in a peer-to-peer based locality-aware multimedia content distribution system. By using simple historical data on content popularity we can estimate future peak request rates for new video elements; these rates are then used to determine which of the contents requested at each STB should be cached. Our algorithm is able to achieve the same high levels of locality of traditional caching policies at a fraction of the storage occupancy, with reductions of the storage space usage as high as 77%. Even higher reductions of up to 92% are possible if we are willing to sacrifice between 5% and 8% of locally served requests. These storage savings make for a compelling case for the implementation of P2P-based opportunistic caching strategies for multimedia content even in those scenarios where storage resources are limited or shared among several services.

7.2 Future Work

7.2.1 Chunking

Throughout our work, we have made the simplifying assumption that multimedia elements are requested and downloaded as a whole. This is, however, not true for most real world VoD services, where each content is divided into multiple chunks of roughly the same size. The purpose of this policy is to achieve a better usage of network resources, as only a limited number of chunks of the current item of interest will be fetched in advance; ideally, the size of the pre-fetching window should be sufficiently large to prevent interruptions in the streaming during temporary network congestion events, but small enough to minimize the overhead of fetching segments that are not going to be needed. While the high access capacity made available by next-generation networks could possibly allow us to relax this conservative requirement on the use of bandwidth, it is true that chunking is implemented in most (if not all) current multimedia services and its inclusion in our model would certainly affect the traffic patterns generated by video requests.

In order to investigate the impact of these aspects on our findings, we are currently in the process of modifying our event-driven simulator, PLACeS, to support chunking. Users will request videos as

in the current implementation, but they will only fetch a number of chunks sufficient to fill a limited buffer. If the popularity model dictates a new content request before the completion of the previous one (i.e., due to a zapping event) no more chunks of the original video will be fetched. It is expected that these changes will reduce the overall amount of traffic transiting the network and extend the active time of data transfers; however, it is likely that the efficiency of caching will not be heavily impaired, as the pull-based caching strategies will ensure that the most popular chunks will still be widely available locally. The results of this study will be published in an appropriate journal as an extension of the work presented in this thesis.

7.2.2 Adaptive Streaming

Another aspect of video streaming that is not properly captured by our simulations is the so-called adaptive streaming De Cicco & Mascolo (2014), sometimes referred to with the acronym DASH (Dynamic Adaptive Streaming over HTTP). Adopted by many existing multimedia services, the idea behind this technique is to dynamically increase or decrease the video quality of the streamed content (i.e., by switching to a different encoding bitrate) in response to changes in network conditions, such as high packet loss or increase in jitter. Naturally, chunking is a pre-requisite for the adoption of DASH, so that subsequent pieces of a video can be streamed at different encoding rates. In our simulations access bandwidth is abundant and always sufficient to stream content at least at the bitrate at which it was encoded, as shown in Section 4.1.2; however, as multimedia operators transition to higher encoding rates and in scenarios where the available bandwidth is limited by external factors (i.e., congestion due to other concurrent services, or network faults), DASH strategies could become again more relevant.

7.2.3 Network-Managed Locality Oracle

The oracle implementation described in Chapter 5 is but a proof of concept of what can be achieved by combining OpenFlow with locality-awareness. Future work should focus on defining load balancing policies that could optimize the cache selection process based on the network status information available at the controller. Furthermore, appropriate mechanisms for AAA (Authentication, Authorization and Accounting) must be defined to ensure that the redirection process does not violates access policies and to guarantee the security of the involved parties. A more robust transparent mechanism to track changes in the composition of user caches should also be investigated.

7.2.4 Improving Caching Optimization

With regards to the storage optimization algorithm, future work should focus on testing its efficiency with traces extracted from real world deployments of video on demand services, to ensure that the artificial generation of request patterns does not affect the validity of the results obtained here. It might

also be interesting to investigate the effects of jointly optimizing the caching decision process and the peer selection process, which would bring the upload parameters y_i^b in the scope of our decision algorithm. Finally, it could be possible to modify the formulation of the problem so that it computes priorities (or weights) for the cached video elements; these values would then be used to guide the eviction of cached videos even in those instances for which there is no feasible solution to the constrained optimization problem.

7.3 Final Remarks

Given the reported benefits of locality-aware P2P for multimedia distribution, both in terms of bandwidth usage, energy consumption and operational expenditures for operators, one might wonder why P2P has typically such a poor reputation in the industry.

One common objection is that P2P is considered to simply not be reliable enough, since users own and control components that are fundamental for the correct functioning of the system. As we discussed in Chapter 1, while P2P will never offer the tight control levels of a CDN infrastructure, the inclusion of caching storage in the STBs, the fact that the locality oracle and its redirection policies are controlled by the network operator itself, and the large number of users involved in the distributed caching process make this approach fairly resilient to churn and faults.

Another commonly perceived issue with P2P is the poor support from both users and service providers. More specifically, some customers might be unwilling to give up part of their storage and upstream capacity in order to support the P2P system. However, the storage requirement would be minimal – even more so thanks to the optimization algorithm presented in Chapter 6 – and in any case a limited and specific portion of the STB storage could be explicitly reserved for this purpose in the lease contract of the STBs, so that it would not even be advertised as available to the end user. Similarly, since we are discussing a network-managed P2P system, it could be designed to prioritize sources that have excess upstream capacities not currently being used; this traffic could be discounted from any data cap calculation from the ISP, so to ensure that it would not represent a cost for the user. Indeed, one could go a step further and consider implementing new business models where customers are incentivised for participating in the distributed caching scheme – after all, this would be convenient for the ISP as long as the cost associated to these incentives is lower than the savings awarded by P2P.

Similarly, on the service provider side there are often resistances to allowing users to store local copies of multimedia content due to fears of copyright infringements and piracy. In our opinion, however, this is a fairly irrational stance, since in many cases users are already allowed to record broadcasted shows to watch them at a later time, and traditional DRM solutions can be used to deter customers from pirating protected stored content. Of course these strategies are not perfect – indeed even in today's streaming services based on CDN or unicast it's possible for a malicious user to circumvent DRM protections and

capture the video stream for its further dissemination – but we do not see how storing replicas of the content locally would significantly increase the threat of piracy.

We hope that this thesis made a good job of highlighting the significant benefits that a network-managed P2P solution would bring to all the actors of a multimedia delivery service. While inevitably there are some remaining issues and concerns to be addressed, we do not believe that they represent insurmountable obstacles, and we are confident that in the future we shall see a greater interest from commercial service providers towards similar distributed caching strategies.

Acronyms

AAA	Authentication, Authorization and Accounting
ALM	Application Layer Multicast
ALTO	Application Layer Traffic Optimization
API	Application Programming Interface
AS	Autonomous System; Access Segment
ASE	Amplified Spontaneous Emission
CapEx	Capital Expenditures
CCN	Content-Centric Networking
CDN	Content Delivery Network
CO	Central Office
CS	Centralized Server
DASH	Dynamic Adaptive Streaming over HTTP
DBA	Dynamic Bandwidth Allocation
DDOS	Distributed Denial Of Service
DHT	Distributed Hash Table
DISCUS	DIStributed Core for unlimited bandwidth supply for all Users and Services
DNS	Domain Name System
DRM	Digital Right Management
DSL	Digital Subscriber Line
EDFA	Erbium-Doped Fiber Amplifier
EPON	Ethernet PON
FC	Fiber Channel
FEC	Forward Error Correction
FTP	File Transfer Protocol
FTTC	Fiber-To-The-Curb/Cabinet
FTTH	Fiber-To-The-Home
FTTP	Fiber-To-The-Premises
FTTX	Fiber-To-The-X
GEM	GPON Encapsulation Mode
GPON	Gigabit-capable PON
HD	High Definition
HTTP	Hyper-Text Transfer Protocol
ICN	Information-Centric Networking
ICT	Information and Communication Technology

IETF	Internet Engineering Task Force
IP	Internet Protocol
IPTV	IP Television
ISP	Internet Service Provider
LFU	Least Frequently Used
LO	Locality Oracle
LR-PON	Long-Reach Passive Optical Network
LRU	Least Recently Used
MAC	Media Access Control
MC	Metro/Core
MPCP	MultiPoint Control Protocol
MS	Multimedia Server
OF	OpenFlow
OLT	Optical Line Terminal
ONU	Optical Network Unit
OpEx	Operational Expenditures
OTN	Optical Transport Network
P2P	Peer-to-Peer
PLACeS	P2P Locality-Aware Content dElivery Simulator
PON	Passive Optical Network
QoE	Quality of Experience
QoS	Quality of Service
RTT	Round-Trip Time
SDH	Synchronous Digital Hierarchy
SNR	Signal-to-Noise Ratio
SOA	Semiconductor Optical Amplifier
SONET	Synchronous Optical Networking
SP	Service Provider
STB	Set Top Box
SSD	Solid State Drive
TCP	Transport Control Protocol
TDM	Time Division Multiplexing
UDP	User Datagram Protocol
UGC	User Generated Content
URL	Uniform Resource Locator
VDSL	Very-high-bit-rate DSL
VOD	Video On Demand
WDM	Wavelength Division Multiplexing

Bibliography

Abeywickrama, S. & Wong, E. (2013), 'Delivery of video-on-demand services using local storages within passive optical networks.', *Optics express* **21**(2), 2083–96.
URL: *http://www.ncbi.nlm.nih.gov/pubmed/23389189*

Abrahamsson, H. & Nordmark, M. (2012), Program popularity and viewer behaviour in a large TV-on-demand system, *in* 'Proceedings of the 2012 ACM conference on Internet measurement conference - IMC '12', ACM Press, New York, New York, USA, p. 199.
URL: *http://dl.acm.org/citation.cfm?id=2398776.2398798*

Aggarwal, V., Feldmann, A. & Scheideler, C. (2007), 'Can ISPS and P2P users cooperate for improved performance?', *ACM SIGCOMM Computer Communication Review* **37**(3), 29.
URL: *http://portal.acm.org/citation.cfm?doid=1273445.1273449*

Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D. & Ohlman, B. (2012), 'A survey of information-centric networking', *IEEE Communications Magazine* **50**(7), 26–36.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6231276*

Aiyarak, P., Saket, A. S. & Sinclair, M. C. (1997), Genetic programming approaches for minimum cost topology optimisation of optical telecommunication networks, *in* 'Genetic Algorithms in Engineering Systems: Innovations and Applications', number 446, pp. 415–420.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=681062*

Alimi, R., Penno, R. & Yang, Y. (2013), 'ALTO Protocol, draft 20'.
URL: *http://tools.ietf.org/html/draft-ietf-alto-protocol-20*

Aperjis, C. & Johari, R. (2011), 'Bilateral and Multilateral Exchanges for Peer-Assisted Content Distribution', *Networking, IEEE/ACM* **19**(5), 1290–1303.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5729356*

Applegate, D., Archer, A., Gopalakrishnan, V., Lee, S. & Ramakrishnan, K. K. (2010), Optimal content placement for a large-scale VoD system, *in* 'Proceedings of the 6th International COnference on - Co-NEXT '10', ACM Press, New York, New York, USA, p. 1.
URL: *http://dl.acm.org/citation.cfm?id=1921168.1921174*

Avramova, Z., De Vleeschauwer, D., Wittevrongel, S. & Bruneel, H. (2010), 'Performance analysis of a caching algorithm for a catch-up television service', *Multimedia Systems* **17**(1), 5–18.
URL: *http://www.springerlink.com/index/10.1007/s00530-010-0201-1*

Avramova, Z., Wittevrongel, S., Bruneel, H. & Vleeschauwer, D. D. (2009), Analysis and Modeling of Video Popularity Evolution in Various Online Video Content Systems: Power-Law versus Exponential Decay, *in* '2009 First International Conference on Evolving Internet', IEEE, pp. 95–100.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5277844*

Baliga, J., Ayre, R., Hinton, K., Sorin, W. & Tucker, R. (2009), 'Energy Consumption in Optical IP Networks', *Journal of Lightwave Technology* **27**(13), 2391–2403.
URL: *http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4815495*

Baliga, J., Ayre, R., Hinton, K. & Tucker, R. S. (2009), Architectures for Energy-Efficient IPTV Networks, *in* 'Optical Fiber Communication Conference and National Fiber Optic Engineers Conference', OSA, Washington, D.C., p. OThQ5.
URL: *http://www.opticsinfobase.org/abstract.cfm?URI=OFC-2009-OThQ5*

Bikfalvi, A., García-Reinoso, J., Vidal, I., Valera, F. & Azcorra, A. (2011), 'P2P vs. IP multicast: Comparing approaches to IPTV streaming based on TV channel popularity', *Computer Networks* **55**(6), 1310–1325.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S1389128610003877*

Borghol, Y., Mitra, S., Ardon, S., Carlsson, N., Eager, D. & Mahanti, A. (2011), 'Characterizing and modelling popularity of user-generated videos', *Performance Evaluation* **68**(11), 1037–1055.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S016653161100099X*

Borst, S., Gupta, V. & Walid, A. (2010), Distributed Caching Algorithms for Content Distribution Networks, *in* '2010 Proceedings IEEE INFOCOM', IEEE, pp. 1–9.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5461964*

Bruneau-Queyreix, J., Negru, D. & Batalla, J. M. (2014), Home-Box based collaborative caching strategy: An asset for Content Delivery Networks, *in* '2014 International Conference on Telecommunications and Multimedia (TEMU)', IEEE, pp. 58–63.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6917736*

Carta, A., Mellia, M., Meo, M. & Traverso, S. (2010), Efficient Uplink Bandwidth Utilization in P2P-TV Streaming Systems, *in* '2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)', IEEE, pp. 1–6.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5683745*

Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.-Y. & Moon, S. (2009), 'Analyzing the Video Popularity Characteristics of Large-Scale User Generated Content Systems', *IEEE/ACM Transactions on Networking* **17**(5), 1357–1370.
URL: *http://dl.acm.org/citation.cfm?id=1665839*

Cha, M., Rodriguez, P., Moon, S. & Crowcroft, J. (2008), On next-generation telco-managed P2P TV architectures, *in* 'Proceedings of the 7th international conference on Peer-to-peer systems', USENIX Association, pp. 5–5.
URL: *http://portal.acm.org/citation.cfm?id=1855646*

Chan, C., Wong, E. & Nirmalathas, A. (2011), Energy Savings Dependency of IPTV Caching Systems on Similarity in User Behavior, *in* 'ECOC 2011', Vol. 1, pp. 7–9.
URL: *http://www.opticsinfobase.org/abstract.cfm?URI=ECOC-2011-We.10.P1.101*

Chan, S.-H. G. & Xu, Z. (2013), 'LP-SR: Approaching Optimal Storage and Retrieval for Video-on-Demand', *IEEE Transactions on Multimedia* **15**(8), 2125–2136.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6589138*

Chanda, A. & Westphal, C. (2013), ContentFlow : Adding Content Primitives to Software Defined Networks, *in* 'Global Communications Conference (GLOBECOM), 2013 IEEE', pp. 2154–2160.

Chellouche, S. A., Negru, D., Chen, Y. & Sidibe, M. (2012), Home-Box-assisted content delivery network for Internet Video-on-Demand services, *in* '2012 IEEE Symposium on Computers and Communications (ISCC)', IEEE, pp. 544–550.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6249353*

Chen, Y.-F. R., Jana, R., Stern, D., Wei, B., Yang, M. & Sun, H. (2009), Zebroid: using IPTV data to support peer-assisted VoD content delivery, *in* 'Proceedings of the 18th international workshop on Network and operating systems support for digital audio and video - NOSSDAV '09', ACM Press, New York, New York, USA, p. 115.
URL: *http://dl.acm.org/citation.cfm?id=1542245.1542272*

Cho, K., Jung, H., Lee, M., Ko, D., Kwon, T. & Choi, Y. (2011), 'How can an ISP merge with a CDN?', *IEEE Communications Magazine* **49**(10), 156–162.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6035830*

Cho, K., Lee, M., Park, K., Kwon, T. T. & Choi, Y. (2012), WAVE: Popularity-based and collaborative in-network caching for content-oriented networks, *in* '2012 Proceedings IEEE INFOCOM Workshops', IEEE, pp. 316–321.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6193512*

Choffnes, D. & Bustamante, F. (2008), 'Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems', *ACM SIGCOMM Computer Communication Review* **38**(4), 363–374.
URL: *http://portal.acm.org/citation.cfm?id=1403000*

Choi, J., Reaz, A. S. & Mukherjee, B. (2012), 'A Survey of User Behavior in VoD Service and Bandwidth-Saving Multicast Streaming Schemes', *IEEE Communications Surveys & Tutorials* **14**(1), 156–169.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5742775*

Cisco CRS-1 Carrier Routing System 16-slot Line Card Chassis Specifications (n.d.).
URL: *http://www.cisco.com/en/US/docs/routers/crs/crs1/16_slot_lc/system_description/reference/guide/sysdsc_a.html*

Claeys, M., Tuncer, D., Famaey, J., Charalambides, M., Latr, S., Turck, F. D. & Pavlou, G. (2014), Towards Multi-Tenant Cache Management for ISP Networks, *in* 'Networks and Communications (EuCNC), 2014 European Conference on', pp. 1–5.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6882692&tag=1*

Dai, J., Liu, F. & Li, B. (2010), 'The disparity between P2P overlays and ISP underlays: issues, existing solutions, and challenges', *IEEE Network* **24**(6), 36–41.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5634441*

De Cicco, L. & Mascolo, S. (2014), 'An Adaptive Video Streaming Control System: Modeling, Validation, and Performance Evaluation', *IEEE/ACM Transactions on Networking* **22**(2), 526–539.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6493502*

De Vleeschauwer, D. & Laevens, K. (2009), 'Performance of Caching Algorithms for IPTV On-Demand Services', *IEEE Transactions on Broadcasting* **55**(2), 491–501.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4837567*

Di Pascale, E., Payne, D. B. & Ruffini, M. (2012), Bandwidth and energy savings of locality-aware P2P Content Distribution in next-generation PONs, *in* '2012 16th International Conference on Optical Network Design and Modelling (ONDM)', IEEE, pp. 1–6.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6210215*

Dixit, A., Lannoo, B., Lambert, S., Colle, D., Pickavet, M. & Demeester, P. (2012), Evaluation of ONU Power Saving Modes in Next Generation Optical Access Networks, *in* 'European Conference and Exhibition on Optical Communication', Optical Society of America, p. Mo.2.B.5.
URL: *http://www.opticsinfobase.org/abstract.cfm?URI=ECEOC-2012-Mo.2.B.5*

Eddy, W. (2007), RFC 4987 - TCP SYN Flooding Attacks and Common Mitigations, Technical report.
URL: *https://tools.ietf.org/html/rfc4987*

Edgeware (2013), Orbit 3020 Video Delivery Server, Technical report.

Effenberger, F., Kani, J.-i. & Maeda, Y. (2010), 'Standardization trends and prospective views on the next generation of broadband optical access systems', *Selected Areas in Communications, IEEE Journal on* **28**(6), 773–780.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5514390*

Famaey, J., Iterbeke, F., Wauters, T. & De Turck, F. (2013), 'Towards a predictive cache replacement strategy for multimedia content', *Journal of Network and Computer Applications* **36**(1), 219–227.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S1084804512001919*

Famaey, J., Wauters, T. & De Turck, F. (2011), On the merits of popularity prediction in multimedia content caching, *in* '12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops', IEEE, pp. 17–24.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5990669*

Fayazbakhsh, S. K., Lin, Y., Tootoonchian, A., Ghodsi, A., Koponen, T., Maggs, B., Ng, K., Sekar, V. & Shenker, S. (2013), Less Pain, Most of the Gain: Incrementally Deployable ICN, *in* 'Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM - SIGCOMM '13', ACM Press, New York, New York, USA, pp. 147–158.
URL: *http://dl.acm.org/citation.cfm?id=2486001.2486023*

Feldmann, A., Gladisch, A., Kind, M., Lange, C., Smaragdakis, G. & Westphal, F.-J. (2010), Energy trade-offs among content delivery architectures, *in* '2010 9th Conference of Telecommunication, Media and Internet', IEEE, pp. 1–6.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5557700*

Figueiredo, F., Benevenuto, F. & Almeida, J. M. (2011), The tube over time: characterizing popularity growth of youtube videos, *in* 'Proceedings of the fourth ACM international conference on Web search and data mining - WSDM '11', ACM Press, New York, New York, USA, p. 745.
URL: *http://portal.acm.org/citation.cfm?doid=1935826.1935925*

Fiorani, M., Aleksic, S., Monti, P., Chen, J., Casoni, M. & Wosinska, L. (2014), 'Energy Efficiency of an Integrated Intra-Data-Center and Core Network With Edge Caching', *Journal of Optical Communications and Networking* **6**(4), 421–432.
URL: *http://www.opticsinfobase.org/abstract.cfm?uri=jocn-6-4-421*

Fratini, R., Savi, M., Verticale, G., Tornatore, M., Elettronica, D., Bioingegneria, I., Milano, P. & Leonardo, P. (2014), Using Replicated Video Servers for VoD Traffic Offloading in Integrated Metro / Access Networks, *in* 'Communications (ICC), 2014 IEEE International Conference on', pp. 3438–3443.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6883853&tag=1*

Gallo, D., Miers, C., Coroama, V., Carvalho, T., Souza, V. & Karlsson, P. (2009), A Multimedia Delivery Architecture for IPTV with P2P-Based Time-Shift Support, *in* '2009 6th IEEE Consumer Communications and Networking Conference', IEEE, pp. 1–2.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4784884*

García, G., Beben, A. & Ramón, F. (2011), COMET: Content mediator architecture for content-aware networks, *in* 'Future Network & ...', pp. 1–8.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6095215*

Garcia-Reinoso, J., Bikfalvi, A., Vidal, I. & Valera, F. (2009), FLaCoSt: A Novel Peer to Peer Architecture for Video Streaming in a Next Generation Network, *in* '2009 First International Conference on Advances in P2P Systems', IEEE, pp. 186–191.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5358975*

Ghodsi, A., Shenker, S. & Koponen, T. (2011), Information-centric networking: seeing the forest for the trees, *in* '... Hot Topics in Networks', ACM, Cambridge, MA, pp. 1–6.
URL: *http://dl.acm.org/citation.cfm?id=2070563*

Gramatikov, S., Jaureguizar, F., Cabrera, J. & García, N. (2013), 'Stochastic modelling of peer-assisted VoD streaming in managed networks', *Computer Networks* **57**(9), 2058–2074.
URL: *http://dx.doi.org/10.1016/j.comnet.2013.04.006*

Guan, K., Atkinson, G., Kilper, D. C. & Gulsen, E. (2011), On the Energy Efficiency of Content Delivery Architectures, *in* '2011 IEEE International Conference on Communications Workshops (ICC)', IEEE, pp. 1–6.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5963557*

Gummadi, K. P. K., Dunn, R. R. J., Saroiu, S., Gribble, S. S. D., Levy, H. M. H. & Zahorjan, J. (2003), Measurement, modeling, and analysis of a peer-to-peer file-sharing workload, *in* 'Proceedings of the nineteenth ACM symposium on Operating systems principles', ACM, pp. 314–329.
URL: *http://portal.acm.org/citation.cfm?id=945475*

Han, D., Andersen, D., Kaminsky, M., Papagiannaki, D. & Seshan, S. (2011), Hulu in the neighborhood, *in* '2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011)', IEEE, pp. 1–10.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5716501*

Hefeeda, M. & Saleh, O. (2008), 'Traffic Modeling and Proportional Partial Caching for Peer-to-Peer Systems', *IEEE/ACM Transactions on Networking* **16**(6), 1447–1460.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4460521*

Huang, C., Li, J. & Ross, K. W. (2007), 'Can internet video-on-demand be profitable?', *ACM SIGCOMM Computer Communication Review* **37**(4), 133.
URL: *http://portal.acm.org/citation.cfm?doid=1282427.1282396*

Hwang, I.-s. & Liem, A. (2012), 'A Hybrid Scalable Peer-to-Peer IP-Based Multimedia Services Architecture in Passive Optical Networks', *Journal of Lightwave Technology* (c), 1–1.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6353867*

IBM Redbooks — BNT Virtual Fabric 10Gb Switch Module for IBM BladeCenter (2011).
URL: *http://www.redbooks.ibm.com/abstracts/tips0708.html*

Idzikowski, F. & Wolisz, A. (2009), Power consumption of network elements in IP over WDM networks, Technical Report July.
URL: *www2.tkn.tu-berlin.de/publications/papers/powerNumbers_final.pdf*

IEEE (2012), IEEE Std 802.3 Standard for Ethernet, Technical Report June, IEEE.

Internet Connection Speed Recommendations - Netflix Help Center (n.d.).
URL: *https://help.netflix.com/en/node/306*

ITU-T (2008), G.984.1 Gigabit-capable passive optical networks (GPON): General characteristics, Technical report, ITU-T.

ITU-T (2010), G.987.1 10-Gigabit-capable passive optical networks (XG-PON): General requirements, Technical report, ITU-T.

ITU-T (2013), G.989.1 40-Gigabit-capable passive optical networks (NG-PON2): General requirements, Technical report, ITU-T.

Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H. & Braynard, R. L. (2009), Networking named content, *in* 'Proceedings of the 5th international conference on Emerging networking experiments and technologies - CoNEXT '09', ACM Press, New York, New York, USA, pp. 1–12.
URL: *http://dl.acm.org/citation.cfm?id=1658939.1658941*

Janardhan, V. & Schulzrinne, H. (2007), Peer assisted VoD for set-top box based IP network, *in* 'Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV - P2P-TV '07', ACM Press, New York, New York, USA, p. 335.
URL: *http://portal.acm.org/citation.cfm?doid=1326320.1326327*

Jayasundara, C. & Nirmalathas, A. (2011), Energy efficient content distribution for VoD services, *in* 'Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2011 and the National Fiber Optic Engineers Conference', pp. 3–5.
URL: *http://www.opticsinfobase.org/abstract.cfm?URI=OFC-2011-OWR3*

Jayasundara, C., Nirmalathas, A. & Wong, E. (2011), Localized P2P VoD Delivery Scheme with Pre-Fetching for Broadband Access Networks, *in* '2011 IEEE Global Telecommunications Conference - GLOBECOM 2011', IEEE, pp. 1–5.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6133940*

Jayasundara, C., Nirmalathas, A., Wong, E. & Chan, C. A. (2011), 'Improving Energy Efficiency of Video on Demand Services', *Journal of Optical Communications and Networking* **3**(11), 870.
URL: *http://www.opticsinfobase.org/abstract.cfm?URI=jocn-3-11-870*

Jayasundara, C., Nirmalathas, A., Wong, E. & Nadarajah, N. (2010), 'Popularity-Aware Caching Algorithm for Video-on-Demand Delivery over Broadband Access Networks', *2010 IEEE Global Telecommunications Conference GLOBECOM 2010* pp. 1–5.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5683976*

Jiang, W., Ioannidis, S., Massoulié, L. & Picconi, F. (2012), Orchestrating massively distributed CDNs, *in* 'Proceedings of the 8th international conference on Emerging networking experiments and technologies - CoNEXT '12', ACM Press, New York, New York, USA, p. 133.
URL: *http://dl.acm.org/citation.cfm?id=2413176.2413193*

Jin, X. & Kwok, Y.-K. (2011), Network aware P2P multimedia streaming: Capacity or locality?, *in* '2011 IEEE International Conference on Peer-to-Peer Computing', IEEE, pp. 54–63.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6038661*

Kerpez, K., Luo, Y. & Effenberger, F. J. (2010), 'Bandwidth Reduction via Localized Peer-to-Peer (P2P) Video', *International Journal of Digital Multimedia Broadcasting* **2010**, 1–10.
URL: *http://www.hindawi.com/journals/ijdmb/2010/562832/*

Lantz, B., Heller, B. & Mckeown, N. (2010), A Network in a Laptop : Rapid Prototyping for Software-Defined Networks, *in* '9th ACM Workshop on Hot Topics in Networks', pp. 1–6.

Laoutaris, N. & Rodriguez, P. (2008), 'ECHOS: edge capacity hosting overlays of nano data centers', *ACM SIGCOMM Computer* **38**(1), 51–54.
URL: *http://portal.acm.org/citation.cfm?id=1341442*

Li, Z. & Simon, G. (2011), Time-Shifted TV in Content Centric Networks: The Case for Cooperative In-Network Caching, *in* '2011 IEEE International Conference on Communications (ICC)', IEEE, pp. 1–6.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5963380*

Ling, Q., Xu, L., Yan, J. & Zhang, Y. (2014), 'An adaptive caching algorithm suitable for time-varying user accesses in VOD systems', *Multimedia Tools and Applications* .
URL: *http://link.springer.com/10.1007/s11042-014-2220-y*

Liu, Y. & Simon, G. (2011), Peer-Assisted Time-Shifted Streaming Systems: Design and Promises, *in* '2011 IEEE International Conference on Communications (ICC)', IEEE, pp. 1–5.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5963371*

Mandal, U., Habib, M. F., Zhang, S., Lange, C., Gladisch, A. & Mukherjee, B. (2014), 'Adopting Hybrid CDN-P2P in IP-Over-WDM Networks: An Energy-Efficiency Perspective', *Journal of Optical Communications and Networking* **6**(3), 303.
URL: *http://jocn.osa.org/abstract.cfm?URI=jocn-6-3-303*

McCauley, M. (n.d.), 'About POX'.
URL: *http://www.noxrepo.org/pox/about-pox/*

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. & Turner, J. (2008), 'OpenFlow: enabling innovation in campus networks', *ACM SIGCOMM Computer Communication Review* **38**(2), 69–74.
URL: *http://portal.acm.org/citation.cfm?id=1355746*

Mu Mu, Ishmael, J., Knowles, W., Rouncefield, M., Race, N., Stuart, M. & Wright, G. (2012), 'P2P-Based IPTV Services: Design, Deployment, and QoE Measurement', *IEEE Transactions on Multimedia* **14**(6), 1515–1527.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6295665*

Nguyen, X. N., Saucez, D. & Turletti, T. (2013), Efficient caching in content-centric networks using OpenFlow, *in* '2013 Proceedings IEEE INFOCOM', IEEE, pp. 1–2.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6566716*

Parvez, K. N., Williamson, C., Mahanti, a. & Carlsson, N. (2012), 'Insights on Media Streaming Progress Using BitTorrent-Like Protocols for On-Demand Streaming', *IEEE/ACM Transactions on Networking* **20**(3), 637–650.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6015506*

Pavlou, G., Wang, N., Chai, W. K. & Psaras, I. (2012), 'Internet-scale content mediation in information-centric networks', *Annals of Telecommunications - Annales Des Télécommunications* **68**(3-4), 167–177.
URL: *http://link.springer.com/10.1007/s12243-012-0333-8*

Payne, D. (2009), FTTP deployment options and economic challenges, *in* 'IEEE ECOC'09', Vienna, Austria, pp. 1–34.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5287047*

Peterson, R. & Sirer, E. (2009), Antfarm: efficient content distribution with managed swarms, *in* 'Proceedings of the 6th USENIX symposium on Networked Systems Design and Implementation', USENIX Association Berkeley, pp. 107–122.
URL: *http://portal.acm.org/citation.cfm?id=1558985*

Podlipnig, S. & Böszörmenyi, L. (2003), 'A survey of Web cache replacement strategies', *ACM Computing Surveys* **35**(4), 374–398.
URL: *http://dl.acm.org/citation.cfm?id=954339.954341*

Qiu, L. & Padmanabhan, V. (2001), On the placement of web server replicas, *in* 'IEEE INFOCOM 2001', Vol. 3, Ieee, pp. 1587–1596.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=916655*

Rao, A., Legout, A., Lim, Y.-s., Towsley, D., Barakat, C. & Dabbous, W. (2011), Network characteristics of video streaming traffic, *in* 'Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies on - CoNEXT '11', ACM Press, New York, New York, USA, pp. 1–12.
URL: *http://dl.acm.org/citation.cfm?id=2079296.2079321*

Ruffini, M., Doran, N., Achouche, M., Parsons, N., Pfeiffer, T., Yin, X., Rohde, H., Schiano, M., Ossieur, P., O'Sullivan, B., Wessaly, R., Wosinska, L., Montalvo, J. & Payne, D. B. (2013), DISCUS: End-to-end network design for ubiquitous high speed broadband services, *in* '2013 15th International Conference on Transparent Optical Networks (ICTON)', IEEE, pp. 1–5.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6602948*

Ruffini, M., Mehta, D., O'Sullivan, B., Quesada, L., Doyle, L. & Payne, D. B. (2012), 'Deployment Strategies for Protected Long-Reach PON', *Journal of Optical Communications and Networking* **4**(2), 118–129.
URL: *http://www.opticsinfobase.org/abstract.cfm?URI=jocn-4-2-118*

Ruffini, M., Wosinska, L., Achouche, M., Chen, J., Doran, N., Farjady, F., Montalvo, J., Ossieur, P., O'Sullivan, B., Parsons, N., Pfeiffer, T., Qiu, X.-Z., Raack, C., Rohde, H., Schiano, M., Townsend, P., Wessaly, R., Yin, X. & Payne, D. (2014), 'DISCUS: an end-to-end solution for ubiquitous broadband optical access', *IEEE Communications Magazine* **52**(2), S24–S32.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6736741*

Saleh, O. & Hefeeda, M. (2006), Modeling and Caching of Peer-to-Peer Traffic, *in* 'Proceedings of the 2006 IEEE International Conference on Network Protocols', IEEE, pp. 249–258.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4110297*

SANDVINE (2013), Global Internet Phenomena Report: 1H 2013, Technical report.
URL: *https://www.sandvine.com/downloads/general/global-internet-phenomena/2013/sandvine-global-internet-phenomena-report-1h-2013.pdf*

Savi, M., Verticale, G., Tornatore, M. & Pattavina, A. (2014), Energy-efficient VoD content delivery and replication in integrated metro/access networks, *in* '2014 IEEE Latin-America Conference on Communications (LATINCOM)', pp. 1–6.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7041840*

Seedorf, J., Kiesel, S. & Stiemerling, M. (2009), 'Traffic localization for P2P-applications: The ALTO approach', *2009 IEEE Ninth International Conference on Peer-to-Peer Computing* pp. 171–177.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5284511*

Skubic, B., In de Betou, E., Ayhan, T., Dahlfort, S. & In, O. (2012), 'Energy-efficient next-generation optical access networks', *IEEE Communications Magazine* **50**(1), 122–127.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6122542*

Song, H., Kim, B.-w. & Mukherjee, B. (2010), 'Long-Reach Optical Access Networks : A Survey of Bandwidth Assignment Mechanisms', *Communications Surveys and Tutorials, IEEE* **12**(1), 112–123.

Spagna, S., Liebsch, M., Baldessari, R., Niccolini, S., Schmid, S., Garroppo, R., Ozawa, K. & Awano, J. (2013), 'Design principles of an operator-owned highly distributed content delivery network', *IEEE Communications Magazine* **51**(4), 132–140.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6495772*

Stiemerling, M. & Kiesel, S. (2013), 'ALTO Deployment Considerations'.
URL: *http://tools.ietf.org/html/draft-ietf-alto-deployments-08*

StorageReview.com (2013), 'Corsair Force F120 SSD Review'.
URL: *http://www.storagereview.com/corsair_force_f120_ssd_review*

Thouin, F. & Coates, M. (2007), 'Video-on-Demand Networks: Design Approaches and Future Challenges', *IEEE Network* **21**(2), 42–48.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4133935*

Van Heddeghem, W., Idzikowski, F., Vereecken, W., Colle, D., Pickavet, M. & Demeester, P. (2012), 'Power consumption modeling in optical multilayer networks', *Photonic Network Communications* **24**(2), 86–102.
URL: *http://www.springerlink.com/index/10.1007/s11107-011-0370-7*

Wikipedia - Erasure Code (n.d.).
URL: *http://en.wikipedia.org/wiki/Erasure_code*

Wolman, A., Voelker, M., Sharma, N., Cardwell, N., Karlin, A. & Levy, H. M. (1999), 'On the scale and performance of cooperative Web proxy caching', *ACM SIGOPS Operating Systems Review* **33**(5), 16–31.
URL: *http://portal.acm.org/citation.cfm?doid=319344.319153*

Wu, C., Li, B., Member, S. & Zhao, S. (2011), 'On Dynamic Server Provisioning in Multi-channel P2P Live Streaming', *Networking, IEEE/ACM Transactions on* **19**(5), 1–1.
URL: *http://iqua.ece.toronto.edu/ bli/papers/chuanwu-tnet11.pdf*

Wu, W. & Lui, J. C. (2012), 'Exploring the Optimal Replication Strategy in P2P-VoD Systems: Characterization and Evaluation', *IEEE Transactions on Parallel and Distributed Systems* **23**(8), 1492–1503.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6095536*

Xie, H., Krishnamurthy, A., Silberschatz, A. & Yang, Y. (2007), P4P: explicit communications for cooperative control between P2P and network providers.
URL: *https://bscw.fing.edu.uy/fingpub/bscw.cgi/d419493/p4p.pdf*

Xie, H., Yang, Y. R., Krishnamurthy, A., Liu, Y. G. & Silberschatz, A. (2008), P4P: provider portal for applications, *in* 'Proceedings of the ACM SIGCOMM 2008 conference on Data communication - SIGCOMM '08', ACM Press, New York, New York, USA, p. 351.
URL: *http://portal.acm.org/citation.cfm?doid=1402958.1402999*

Xu Cheng, Jiangchuan Liu, Haiyang Wang & Chonggang Wang (2012), 'Coordinate Live Streaming and Storage Sharing for Social Media Content Distribution', *IEEE Transactions on Multimedia* **14**(6), 1558–1565.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6304932*

Yang, Y., Chow, A. L. H., Golubchik, L. & Bragg, D. (2010), Improving QoS in BitTorrent-like VoD Systems, *in* '2010 Proceedings IEEE INFOCOM', IEEE, pp. 1–9.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5462029*

Yu, H., Zheng, D., Zhao, B. Y. & Zheng, W. (2006), 'Understanding user behavior in large-scale video-on-demand systems', *ACM SIGOPS Operating Systems Review* **40**(4), 333.
URL: *http://dl.acm.org/citation.cfm?id=1218063.1217968*

Zhang, C., Dhungel, P. & Di Wu, K. (2010), 'Unraveling the bittorrent ecosystem', *IEEE Transactions on Parallel and Distributed Systems* **22**(7), 1164–1177.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5482574*

Zhe Li & Simon, G. (2013), 'In a Telco-CDN, Pushing Content Makes Sense', *IEEE Transactions on Network and Service Management* **10**(3), 300–311.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6514997*

Zhou, Y., Fu, T. Z. J. & Chiu, D. M. (2011), Statistical modeling and analysis of P2P replication to support VoD service, *in* '2011 Proceedings IEEE INFOCOM', IEEE, pp. 945–953.
URL: *http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5935322*