Technical Report

UCAM-CL-TR-826
ISSN 1476-2986

Number 826

UNIVERSITY OF
CAMBRIDGE

Computer Laboratory

GREEN IPTV: a resource and energy efficient network for IPTV

Fernando M. V. Ramos

December 2012

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted November 2012 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Clare Hall.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

GREEN IPTV: A Resource and Energy Efficient Network for IPTV
Fernando M. V. Ramos

Abstract

The distribution of television is currently dominated by three technologies: over-the-air broadcast, cable, and satellite. The advent of IP networks and the increased availability of broadband access created a new vehicle for the distribution of TV services. The distribution of digital TV services over IP networks, or IPTV, offers carriers flexibility and added value in the form of additional services. It causes therefore no surprise the rapid roll-out of IPTV services by operators worldwide in the past few years.

IPTV distribution imposes stringent requirements on both performance and reliability. It is therefore challenging for an IPTV operator to guarantee the quality of experience expected by its users, and doing so in an efficient manner. In this dissertation I investigate some of the challenges faced by IPTV distribution network operators, and I propose novel techniques to address these challenges.

First, I address one of the major concerns of IPTV network deployment: channel change delay. This is the latency experienced by users when switching between TV channels. Synchronisation and buffering of video streams can cause channel change delays of several seconds. I perform an empirical analysis of a particular solution to the channel change delay problem, namely, predictive pre-joining of TV channels. In this scheme each Set Top Box simultaneously joins additional multicast groups (TV channels) along with the one requested by the user. If the user switches to any of these channels next, switching latency is virtually eliminated, and user experience is improved. The results show that it is possible to eliminate zapping delay for a significant percentage of channel switching requests with little impact in access network bandwidth cost.

Second, I propose a technique to increase the resource and energy efficiency of IPTV networks. This technique is based on a simple paradigm: avoiding waste. To reduce the inefficiencies of current static multicast distribution schemes, I propose a semi-dynamic scheme where only a selection of TV multicast groups is distributed in the network, instead of all. I perform an empirical evaluation of this method and conclude that its use results in significant bandwidth reductions without compromising service performance. I also demonstrate that these reductions may translate into significant energy savings in the future.

Third, to increase energy efficiency further I propose a novel energy and resource friendly protocol for core optical IPTV networks. The idea is for popular IPTV traffic

to optically bypass the network nodes, avoiding electronic processing. I evaluate this proposal empirically and conclude that the introduction of optical switching techniques results in a significant increase in the energy efficiency of IPTV networks.

All the schemes I present in this dissertation are evaluated by means of trace-driven analyses using a dataset from an operational IPTV service provider. Such thorough and realistic evaluation enables the assessment of the proposed techniques with an increased level of confidence, and is therefore a strength of this dissertation.

List of publications

In the course of my studies I have published the papers and technical reports presented below. Some papers discuss topics covered in this thesis, while others describe distinct research threads. The paper “Channel Smurfing: Minimising Channel Switching Delay in IPTV Distribution Networks” has been given a best paper award.

[2012] H. Kim, J. Crowcroft, and **F. M. V. Ramos**. Efficient channel selection using hierarchical clustering. In *WoWMoM*, San Francisco, CA, June 2012.

[2011] **F. M. V. Ramos**, J. Crowcroft, R. J. Gibbens, P. Rodriguez, and I. H. White. Reducing channel change delay in IPTV by predictive pre-Joining of TV Channels. *Signal Processing: Image Communication*, 26(7):400412, 2011.

[2010] **F. M. V. Ramos**, R. J. Gibbens, F. Song, P. Rodriguez, J. Crowcroft, and I. H. White. Reducing energy consumption in IPTV networks by selective pre-joining of channels. In *SIGCOMM workshop on green networking*, New Delhi, India, Aug. 2010.

[2010] **F. M. V. Ramos**, J. Crowcroft, R. J. Gibbens, P. Rodriguez, and I. H. White. Channel smurfing: Minimising channel switching delay in IPTV distribution networks. In *ICME*, Singapore, July 2010.

[2010] F. Song, H. Zhang, S. Zhang, **F. M. V. Ramos**, and J. Crowcroft. Relative delay estimator for SCTP-based Concurrent Multipath Transfer. In *GLOBECOM*, Miami, FL, Dec. 2010.

[2009] **F. M. V. Ramos**, F. Song, P. Rodriguez, R. Gibbens, J. Crowcroft, and I. H. White. Constructing an IPTV workload model. In *SIGCOMM poster session*, Barcelona, Spain, Aug. 2009.

[2009] **F. M. V. Ramos**, A. Giorgetti, F. Cugini, P. Castoldi, J. Crowcroft, and I. H. White. Power excursion aware routing in GMPLS-based WSONs. In *OFC*, San Diego, CA, Mar. 2009.

[2009] F. Song, H. Zhangy, S. Zhangy, **F. M. V. Ramos**, and J. Crowcroft. Relative delay estimator for multipath transport. In *CoNEXT Student Workshop*, Rome, Italy, Dec. 2009.

[2009] F. Song, H. Zhang, S. Zhang, **F. Ramos**, and J. Crowcroft. An estimator of forward and backward delay for multipath transport. *Computer Laboratory Technical Report UCAM-CL-TR-747*, March 2009.

[2008] **F. Ramos**. Design of a GMPLS system for provisioning optical core and access networks for multicast TV. In *Eurosys doctoral workshop*, Glasgow, UK, Mar. 2008.

Contents

List of Figures

List of Tables

Glossary

ADSL Asymmetric Digital Subscriber Line.

ALR Adaptive Link Rate.

BGP Border Gateway Protocol.

CAGR Compound Annual Growth Rate.

CBT Core Based Trees.

CCN Content Centric Networking.

CDN Content Distribution Network.

CES Consumer Electronics Show.

CMOS Complementary Metal Oxide Semiconductor.

DEMUX Demultiplexer.

DFS Dynamic Frequency Scaling.

DSL Digital Subscriber Line.

DSLAM DSL Access Multiplexer.

DSM Dynamic Spectrum Management.

DVMRP Distance-Vector Multicast Routing Protocol.

DVS Dynamic Voltage Scaling.

EP Energy Proportional.

FEC Forward Error Correction.

FID First I-Frame delay.

FPGA Field Programmable Gate Array.

FTTC Fibre To The Cabinet.

FTTH Fibre To The Home.

FTTx Fibre To The x.

FWM Four Wave Mixing.

GMPLS Generalized MPLS.

GOP Group Of Pictures.

HD High Definition.

HDTV High Definition TV.

I/O Input/Output.

ICT Information and Communications Technology.

IGMP Internet Group Management Protocol.

IP Internet Protocol.

IPTV Internet Protocol Television.

ISP Internet Service Provider.

LAN Local Area Network.

MC-RWA Multicast Routing and Wavelength Assignment.

MEMS Micro Electro Mechanical Systems.

MILP Mixed Integer Linear Programming.

MPLS Multiprotocol Label Switching.

MUX Multiplexer.

NaDa Nano Data Center.

NIC Network Interface Controller.

NTP Network Time Protocol.

OEO Optical-Electrical-Optical.

OSPF Open Shortest Path First.

OSPF-TE OSPF-Traffic Engineering.

OXC Optical Cross Connect.

P2P Peer-to-peer.

PIM Protocol Independent Multicast.

PIM-DM Protocol Independent Multicast - Dense Mode.

PIM-SM Protocol Independent Multicast - Sparse Mode.

QoE Quality of Experience.

RAM Random Access Memory.

RIP Routing Information Protocol.

RP Rendezvous Point.

RSVP Resource Reservation Protocol.

RSVP-TE RSVP-Traffic Engineering.

RTCP Real-time Transport Control Protocol.

RTP Real-time Transport Protocol.

RWA Routing and Wavelength Assignment.

SaD Split and Delivery.

SD Standard Definition.

SDTV Standard Definition TV.

SFCS Synchronisation Frames for Channel Switching.

SPM Self Phase Modulation.

SPT Shortest Path Trees.

STB Set Top Box.

TaC Tap and Continue.

TE Traffic Engineering.

UDP User Datagram Protocol.

UHDTV Ultra High Definition TV.

VM Virtual Machine.

VoD Video on Demand.

WC Wavelength Conversion.

WDM Wavelength Division Multiplexing.

ZA Zapping Accelerator.

Chapter 1

Introduction

In 1884, Paul Nipkow, a German engineering student, proposed and patented the Nipkow disk, "an electric telescope for the electric reproduction of illuminating objects" [209]. Some years later, this mechanical image scanning device became the basis for the essential component of the first television set. The importance of this invention was emphasised by Albert Abramson, an historian of television, who considered this to be "the master television patent" [1]. Some decades later, in 1925, John Logie Baird gave the first public demonstration of television at Selfridges department store in London. These two events mark the beginning of a revolution that continues today. For more than half a century, television has been a dominant and pervasive mass media, experimenting profound changes [42]. From mechanical to fully-electronic television, from black-and-white to colour, from analog to digital, the technological advances have been impressive.

The distribution of television is currently dominated by three technologies: over the air broadcasts, cable, and satellite. The advent of IP networks and the increased availability of broadband access created a new vehicle for the distribution of TV services. The distribution of digital TV services over IP networks, or IPTV, offers much more than traditional broadcast TV. The high visual quality and reliability expectations of traditional broadcast TV can now marry the interactivity, flexibility and rich personalisation enabled by IP technologies [191]. IPTV has even been hyperbolised as the "killer application for the next-generation Internet" [211].

The topic of this dissertation is the distribution of TV over IP networks. In the following pages I investigate some of the challenges faced by IPTV network operators, and I propose and analyse novel techniques to address these challenges.

1.1 What is IPTV?

IPTV is a method of delivering entertainment-quality video using an IP network as the medium, instead of the hitherto predominant cable, free-to-air or satellite broadcasts. Advances in networking technology, digital media and codecs[1] have made it possible for broadband service providers throughout the world to begin streaming live and on-demand television to homes over their high-speed IP networks [141]. IPTV extends the reachability of content to any IP-connected

[1] A codec is a device capable of encoding or decoding a digital media stream.

device, which today means it enables the availability of content to almost anywhere (something users cannot get from traditional services) [29]. As an example of this concept, several companies, such as the BBC and Time Warner Cable Inc., launched very recently applications that allow users to watch live TV and catch up on their favourite TV programmes on their iPads, iPhones, and Android mobile devices [38].

1.2 Motivation

This section outlines the motivation for doing research on IPTV. In the following sections I discuss the value of IPTV and highlight the challenges faced by IPTV operators addressed in this dissertation.

1.2.1 The Value of IPTV

Until the development of IP networks, television was a broadcast medium. Traditional TV networks offered limited freedom of choice and control to its users. Over the years, the number of channels increased from a few free-to-air broadcasts to several hundreds, offering a much wider selection but still effectively delivering the same service. In this sense, IPTV reinvents television [191]. Its integral return channel and its ability to address individual users paves the way for new interactive services.

This bidirectional communication capability also gives more visibility on viewing activities, allowing the service provider to know what the users are watching and when. This raises several issues, such as privacy, but can be a catalyst for the creation of new applications. Television advertisement, for instance, can be reinvented [191]. For telecommunications operators, IPTV offers flexibility and added value in the form of additional services that can be offered to its customers, which improves their profitability and competitive edge [141].

1.2.2 Killer application?

The past few years have witnessed the rapid roll-out of IPTV services. IPTV has been launched by major service providers worldwide — France Telecom, AT&T, Telefonica, China Telecom, Korea Telecom, among others [43] — and its popularity is on the rise [201]. In the United States, for instance, there are already more than 5 million subscribers, and this number is expected to increase to 15.5 million by 2013 [156]. By early 2009, there were more than 25 million IPTV users in the world [138]. IPTV provider managed traffic is expected to grow at a Compound Annual Growth Rate (CAGR) of 53% for the next few years [56].

Contrary to other industries (newspapers, music industry, book publishers) TV is coping well with technological change [71]. An average TV viewer spends 5 hours per day in front of the box, five times more than using the Internet [148]. On February 17th 2010, 106 million Americans watched the Super Bowl — a record for a single program. Tokyo residents spend more time consuming media online (from 6 minutes in 2000 to one hour in 2009), but the time spent in front of the TV is also growing — now to an average of 216 minutes [71]. Television is therefore still supreme at holding the attention of a large number of people for long periods.

In 1996, George Gilder, an American writer, claimed that by the end of the twentieth century television would be extinct due to technological advances. From his book, "Life after Television" [86]:

"All these developments converge in one key fact of life, and death, for telecommunications in the 1990s. Television and telephone systems — optimized for a world in which spectrum or bandwidth was scarce — are utterly unsuited for a world in which bandwidth is abundant."

The facts seem to contradict Gilder's assertion. As with Mark Twain's reports on his own death, Gilder's claims on the death of television seem to be exaggerated.

1.2.3 Challenges

IPTV distribution imposes stringent requirements on both performance and reliability, requiring low latency, a tight control of jitter, and small packet loss in order to guarantee the expected video quality. The offer of this service is therefore challenging for IPTV operators that want to match the level of quality of service that customers are accustomed to from other TV service providers. The problem is that IP networks are "best effort", susceptible to lost or dropped packets as bandwidth becomes scarce and jitter increases. This challenge is partially solved by current IPTV networks being provider-managed services. In their "walled garden" IPTV infrastructures service providers control the load of the network elements and use traffic prioritisation and bandwidth reservation techniques to assure service quality and performance. But other problems exist in this respect. A major concern of IPTV operators is channel change delay. This is the latency experienced by users when switching between channels. Due to bandwidth limitations, in current IPTV networks only one or two TV channels are distributed in the access link that connects the network to the Set Top Box[1](STB). When a user switches to a new channel, the STB has to issue a new channel request towards the network. This is one of the causes of channel change delay. In addition to this network delay, synchronisation and buffering of media streams can cause channel change delays of several seconds. This is the first challenge I address in this dissertation, in Chapter 5.

Video distribution is very resource intensive. High definition TV requires bit rates on the tens of Mbps range, and future ultra high definition formats may increase this figure by orders of magnitude. Efficiency in distribution is therefore a major concern. The emergence of scalable multicast protocols has provided the means for an efficient distribution of TV services. However, current IPTV multicast architectures remain inefficient. They use *static* multicast, distributing *all* TV channels from the source to every access node in the network continuously. As particular channels have no viewers at particular time periods, this method is provably *resource* and *energy* inefficient. These inefficiencies are the second challenge for IPTV network providers I address in this dissertation, in Chapters 6 and 7.

[1] The device that turns the packets received from the network into content which is then displayed on the television screen.

1.3 Issues not covered in this thesis

Research on IPTV covers a broad range of interesting topics. The term IPTV itself has been used in the research community to mean very different things. To try to clarify its precise topic, I provide in this section an explanation of what this dissertation *is not*.

1.3.1 IPTV, not Internet TV nor P2P TV

The term IPTV is sometimes used in certain contexts to describe WebTV, Internet TV or P2P-based TV. Internet TV and WebTV are normally used to describe the delivery of TV programming over the public Internet, typically to personal computers as streamed or downloadable video content [5]. Broadcasters such as the BBC are already providing this type of service over the Internet [25]. This approach is also referred to as over-the-top (OTT) video, since it essentially uses the Internet as a transport pipe to deliver content. As opposed to PC-based viewing, IPTV services target a TV viewing environment integrated with set-top boxes (STBs), providing cable TV-like experience. The distribution of these services is done in closed, privately-managed IPTV networks. The service provider has full control over content distribution, storage management, and bandwidth provisioning, to ensure end-to-end quality of delivery. In addition, IPTV is a server-centred architecture, in contrast with P2P-based TV. In this dissertation, I target IPTV, not Internet TV nor P2P TV.

1.3.2 Broadcast IPTV, not VoD

IPTV services are usually classified into two main types: broadcast television and Video-on-Demand (VoD). The VoD service model is one-to-one. A single copy of a specific program is unicast to a single subscriber on request. In contrast, IPTV broadcast services require all viewers to watch a program simultaneously, according to a predetermined schedule. This is a one-to-many service model where for efficiency reasons IP multicast is used for distribution. In this dissertation, I target broadcast television services, not VoD.

1.3.3 Single-domain IPTV, not multiple

Routing in the Internet forms a two level hierarchy: inter and intra-domain. Traffic crossing multiple network domains is governed by policy-based border gateway protocol (BGP) [168]. As I referred before, due to its stringent quality of service requirements, IPTV is currently a service managed by a single provider. The design techniques proposed in this dissertation apply therefore to a single independently operated network domain.

1.4 Contributions

In this dissertation I propose and analyse several techniques to assist IPTV providers in the design of novel resource and energy efficient networks. These techniques focus on the technological challenges referred to before: IPTV service's high channel change delay and network efficiency. The main contributions of this dissertation are as follows.

1.4.1 Reducing channel change delay

The first contribution of this dissertation is an empirical analysis of a particular solution to the channel change delay problem, namely, predictive pre-joining of TV channels. In this scheme each Set Top Box simultaneously joins additional multicast groups (TV channels) along with the one requested by the user. If the user switches to any of these channels next, switching latency is virtually eliminated, and user experience is improved. Previous work on this subject used simple mathematical models to perform analytical studies or to generate synthetic data traces to evaluate these pre-joining methods. By analysing IPTV channel switching logs from an IPTV service offered by an operational backbone provider, I demonstrate that these models are conservative in terms of the number of channel switches a user performs during zapping periods. They therefore do not evidence the true potential of predictive pre-joining solutions. To fill this gap I perform a trace-driven analysis using the dataset referred to above (the switching logs) to evaluate the potential of these solutions. The main conclusion of this study is that a simple scheme where the neighbouring channels (i.e., the channels adjacent to the requested one) are pre-joined by the Set Top Box alongside the requested channel, during zapping periods only, eliminates zapping delay for around half of all channel switching requests to the network. Importantly, this result is achieved with a negligible increase of bandwidth utilisation in the access link [163, 164].

1.4.2 Reducing energy by avoiding waste

Current IPTV service providers build *static* multicast trees for the distribution of TV channels. By static multicast I mean that all receivers are known beforehand, and no new group members are allowed to join - it is a *static* set of receivers for all TV content. This means all TV channels are distributed everywhere in the network continuously. This is justified to guarantee the quality of experience required by IPTV customers. By distributing TV channels to as close to the users as possible, network latencies do not add significantly to the already high channel change delay. However, as particular channels have no viewers at particular time periods, this method is provably resource and energy inefficient. To reduce these inefficiencies, I propose a semi-dynamic scheme where only a selection of TV multicast groups is distributed in the network, instead of all. This selection changes with user activity. This method is evaluated empirically by analysing the same dataset as above. I demonstrate that by using the proposed scheme IPTV service providers can save a considerable amount of bandwidth while affecting only a very small number of TV channel switching requests. Furthermore, I show that although today the bandwidth savings would have reduced impact in energy consumption, with the introduction of numerous very high definition channels this impact will become significant [165].

1.4.3 Reducing energy by integrating optical switching

The third contribution of this dissertation is a novel energy friendly protocol for core optical IPTV networks. The objective is to further increase the energy efficiency of IPTV networks. The fundamental concept is to blend electronic routing and optical switching, thus gluing the low-power consumption advantage of circuit-switched all-optical nodes with the superior bandwidth-efficiency of packet-switched IP networks. The main idea is to optically switch popular TV channels. These channels are watched by many, having viewers everywhere in the network at any time. These are long-lived flows in the network, and are therefore perfect targets for this type of slow energy-friendly switching. With the use of this protocol, popular IPTV traffic optically bypasses the network nodes, i.e., this traffic avoids electronic processing. I evaluate this proposal empirically by performing a trace-driven analysis using the IPTV dataset mentioned before. The main conclusion is that the introduction of optical switching techniques results in a quite significant increase in the energy efficiency of IPTV networks.

1.4.4 Evaluation

All the schemes presented in this dissertation are evaluated by means of trace-driven analyses using a dataset from an operational IPTV service provider, Telefonica. This dataset was obtained from measurements collected by Telefonica in its network, from April 2007 to October 2007. The traces recorded user channel change activity from Telefonica's IPTV service, Imagenio. The dataset scales up to 150 TV channels, six months, and 255 thousand users. It is widely accepted that a thorough evaluation using real workloads enables the assessment of future network architectures with an increased level of confidence. This is particularly relevant in a research field that has relied heavily upon hypothetical user models which are different from the reality and can lead to incorrect estimation of system performance. I believe a strength of this dissertation lies in such thorough evaluation using real traces of real IPTV usage.

1.5 Outline

This dissertation is organised as follows. In Chapter 2, I present some background to the distribution of TV in IP networks. Then, in Chapter 3, I describe relevant research related to IPTV with a particular focus on the challenges addressed and the techniques proposed in this dissertation. In Chapter 4, I justify the option for the methodology used and I describe the dataset used for evaluation. In particular, I detail the process of data collection, data cleaning, and how the dataset is validated. Next, I address the problem of channel change delay in IPTV networks, in Chapter 5. In particular, I present an in-depth analysis of predictive pre-joining solutions to this problem. In Chapter 6, I propose a semi-dynamic multicast scheme to increase IPTV network's resource and energy efficiency. To increase energy efficiency further, in Chapter 7, I assess the opportunities for introducing optical bypass in core optical IPTV networks and demonstrate its effectiveness in reducing energy consumption. Finally, in Chapter 8, I summarise the contributions of this work and discuss possible directions for future research.

Chapter 2

Background

The topic of this dissertation is the distribution of TV in IP networks. There are two aspects to this issue: the content and the delivery. In the first section of this chapter, I address the former, explaining why video *content* needs to be coded and how it is coded. I also explain its implications in increasing channel change delay in IPTV networks. Then I cover the *delivery* of TV services using IP multicast. I describe the multicast service model and the most common multicast protocols. Next, I describe a typical IPTV architecture in some detail. Finally, I conclude the chapter with a summary of the challenges faced by IPTV providers which were the motivation for this work.

2.1 Video coding

Video consists of a series of pictures, or frames, taken at regular intervals (typically every 33.3 ms or 40 ms [65]). The data rate of this raw signal is too high for economical transport over telecommunication networks. Uncompressed Standard Definition TeleVision (SDTV), for instance, requires a bit rate of around 200 Mbps, and High Definition TeleVision (HDTV) already demands bit rates close to 1 Gbps [83].

Since network bandwidth is a scarce resource, compression techniques are needed to save transmission capacity (and storage). Efficient media coding schemes have therefore been developed, such as MPEG-2 [109] and MPEG-4 [110]. They make use of the fortunate fact that much audio and video is redundant, containing, effectively, repeated or less useful data, which leads to a high correlation between adjacent frames in a typical video sequence. Hence, dependencies between neighbouring frames can be exploited to increase coding efficiency.

In these coding schemes the video streams are divided into segments, each commonly termed a Group of Pictures (GOP), as in Figure 2.1. The video is coded, with three types of frames defined: I, P and B-frames. A GOP is composed of all the predicted frames (P and B) between two I-frames, together with the starting I-frame. The GOP size is thus defined as the time between I-frames. Each type of frame explores a different redundancy pattern existing in video sequences and therefore results in different compression efficiencies and in different functionality.

I-frames[1] are encoded with image compression techniques that exploit the spatial correlation of pixels within the frame without using information from any other frames. Since they are not dependent on any other frame, they are used as a decoding reference for other frames, and can serve as access points where decoding can begin. P and B-frames are predicted based on one or more surrounding frames, using the estimated motion of objects of the frame it refers to. Therefore, they cannot be decoded alone. P-frames use motion prediction from a past reference frame and B-frames use a prediction based on references from the past, future, or a combination of both.

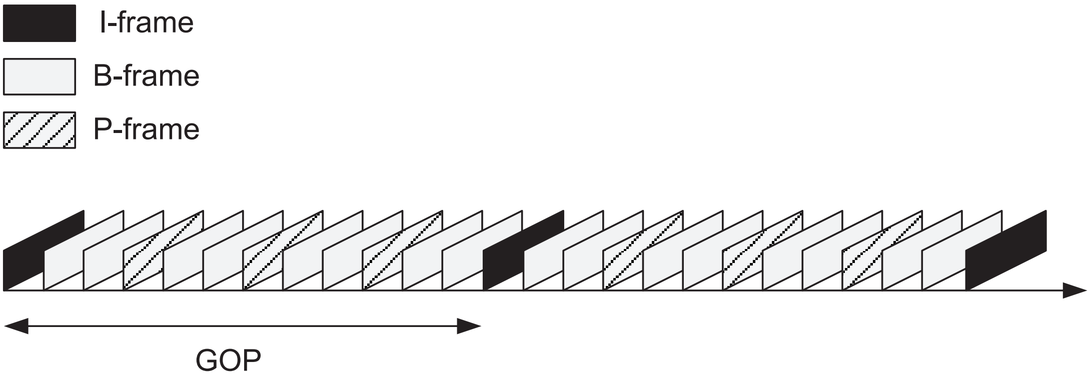


Figure 2.1: Typical frame structure

To decode a video stream the decoder will therefore need an I-frame as a first reference frame, which can be decoded without further information. To speed up play out time, it would be advantageous to transmit I-frames very frequently. This way decoding and play out could start sooner, reducing inconvenient delays. The problem is that I-frames are significantly larger than P or B-frames, requiring higher storage space and higher bit rates to be transported. Depending on the content, this difference can be of one order of magnitude [182]. There is thus a trade-off between compression efficiency on one side and play out performance on the other. In practise, GOP duration is typically in the range of 1 to 2 seconds [65, 182]. More advanced codecs, however, have longer GOPs to gain from the encoding efficiency, at the cost of higher latencies.

2.1.1 IPTV channel change delay

In traditional analogue TV broadcast and cable technology, channel change is almost instantaneous since it only involves the TV receiver tuning to a specific carrier frequency, demodulating the content and displaying it on the TV screen. The zapping delay in these systems is typically less than 200 ms [28]. TV viewers thus consider zapping times to be virtually instant, and have become used to this surfing (through channels) experience. With the digitisation and compression of content, zapping times have increased significantly. Users already experience this today

[1] These frames are also called anchor-frames or key-frames, albeit in sometimes different contexts. For instance, an I-frame is always a key-frame in MPEG-2, but this is not a sufficient condition in MPEG-4. In the following, however, I will consider the MPEG-2 standard and use only the term I-frame.

in digital cable TV networks, but it is an even more severe problem in IPTV, since zapping times are also affected by network delay. Users zapping in IPTV usually experience a couple of seconds' delay or more [176].

Kooij *et al.* [128] presented a study recently where they conclude that to achieve an acceptable quality of service, channel change time needs to be below 0.43 s. Although the study is limited in terms of the size of the test subject population, it is clear that IPTV high zapping delay degrades the quality of experience perceived by customers and is a major obstacle for IPTV services wide adoption. To understand how to mitigate this problem, it is important to understand the components of channel change time.

Consider Figure 2.2, where the main contributors to zapping delay are depicted. The user starts by issuing a channel change request using the remote control. The request reaches the Set Top Box (STB) after an estimate 5-10 ms delay [199] (step 1 in the figure).

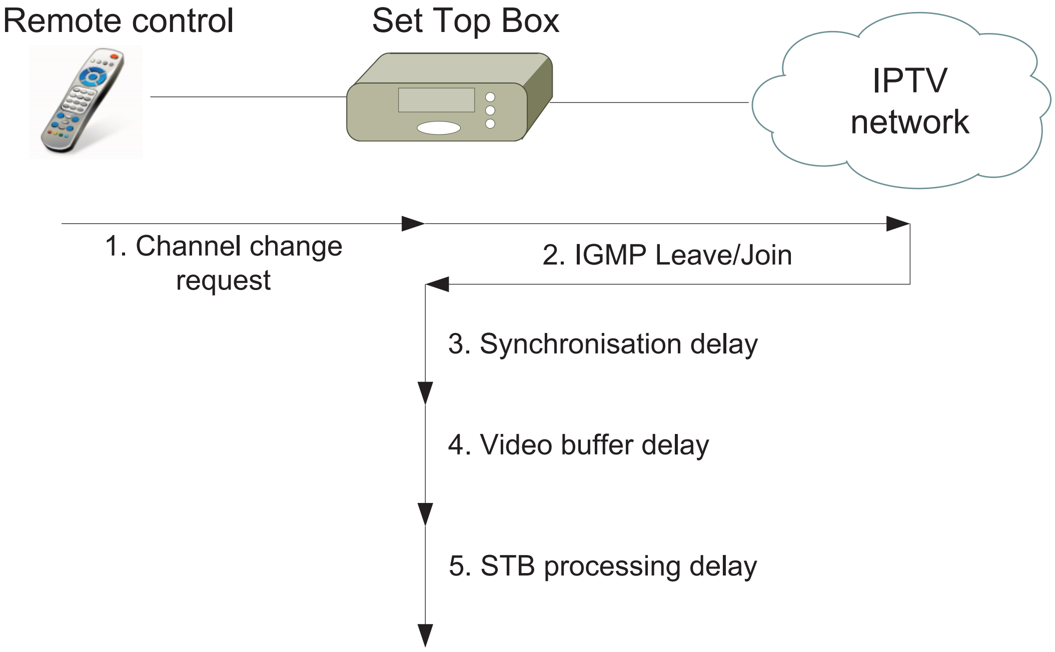


Figure 2.2: Main components of channel change time

In IPTV video delivery, in contrast to cable networks, for instance, typically only the channel the user is watching is delivered to the STB at any one time. This is due to bandwidth limitations in the access network. When a user switches to a new channel, the STB has to issue a new channel request towards the network (step 2). Since video distribution is done via multicasting, this is translated into leave and join multicast requests. In typical systems, these operations are handled by a group management protocol, usually IGMP (Internet Group Management Protocol) [39]. This protocol is analysed later in this chapter. In short, the STB sends a leave request from the current multicast session (the TV channel the user is currently watching) and a join request to the new multicast group (the TV channel the user is switching to). This channel change request reaches the first upstream network node that has the channel available and the routing infrastructure sets up the multicast forwarding state to deliver the packets of the multicast session to the STB. To minimise this component of the delay, *all* TV channels are distributed very close to the user, commonly to the DSLAM[1] (in DSL networks[2]) or to the local router. In addition to this *signalling delay* we need to add the *propagation delay* experienced in the access link. The sum of these types of delay, which I jointly call *network delay*, is usually below 100-200 ms [27, 28, 80, 182, 189]. This is therefore a relatively unimportant contributor to the overall delay, as is made clear in the following paragraphs.

After the STB receives the first packets from the recently joined multicast group, there is still a time lag before it can start consuming the audiovisual data because the STB must wait for the next I-frame before it can start decoding the content, as explained in the previous section. I refer to this as the *synchronisation delay* (step 3). The maximum synchronisation delay is equal to the duration of the GOP, which occurs when the STB just misses the start of an I-frame and thus has to wait for the next. On average, this delay is half the GOP duration. Synchronisation delay therefore comprises a substantial portion of the channel change time (recall that GOP duration is typically in the range of 1 to 2 seconds). Many video services also employ content encryption, so the encryption keys must be acquired and provided to the decryption engine for decrypting the content, and this also adds to synchronisation delay.

In general, multicast-based video applications use an unreliable underlying transport protocol such as UDP [158] to distribute IPTV content. For this reason, packet loss may occur and loss-repair techniques need to be included in the system. For example, a local repair server can be included at the network edge for retransmitting lost packets or, if the retransmission cost is high, Forward Error Correction (FEC) techniques[3] may be used to provide reliability. Regardless of the type of loss-repair method, buffering will be required at the receiver side for these operations to be performed (step 4). Buffering reduces the system sensitivity to short term fluctuations in the data arrival rate by absorbing variations in end-to-end delay and allowing margins for retransmissions when packets are lost.

There are three other reasons to buffer incoming packets before forwarding them to the decoder: to avoid under-run (starvation) and to compensate for network jitter and packet-reordering delay. Starvation results from the different frames being encoded at different data rates, and thus the video encoding process resulting in a variable bit rate stream. Since the network flow is typically constant bitrate (or capped variable), this mismatch between the input and the output of the encoder is solved with the inclusion of a smoothing buffer. Network jitter and packet-reordering are caused by cross-traffic in network equipment: in an IP network traffic is asynchronous, so packets have to wait in buffers. Also, different packets may follow different paths, and this results in a variable and unpredictable delay between packets. While the amount of protection offered by a buffer grows with its size, so does the latency it introduces. Typical decoder buffer requirements range from 1 to 2 seconds [28].

[1] The DSL Access Multiplexer is a layer-2 aggregation switch that connects multiple customer DSL interfaces to the network.

[2] In this type of access network the digital data is transmitted over twisted-pair copper wires of a local telephone network, using separate frequency bands from the telephone signals.

[3] With FEC the sender adds redundant data to its messages allowing the receiver to detect and correct errors, at the cost of higher bandwidth requirements.

The final source of delay is end-system delay (step 5). This is the processing delay in the STB and display device. This can occur at a number of system layers, and is generally a trade off between terminal resources (memory and processor speed) and cost. The processing time depends very much on the STB, but 150 ms is a typical figure [182].

Figure 2.3 pictorially summarises the contribution of each component of channel change time (not to scale). As can be observed, the main contributors to IPTV channel change latency are stream synchronisation and buffering (steps 3 and 4 in Figure 2.2), adding up to around 2 seconds on average [80, 182, 189]. The main concern in the industry and in the research community has been, in fact, to try to improve the performance on these two aspects [30]. Chapter 5 of this dissertation is also devoted to this problem.

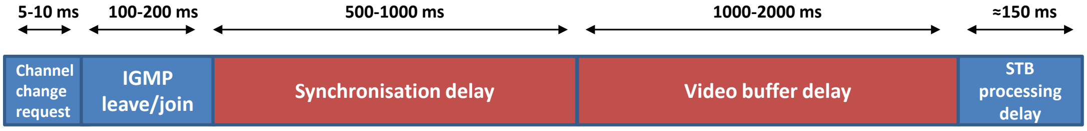


Figure 2.3: Contribution of each component of channel change time (not to scale)

2.2 IP multicast

This section covers the second aspect concerning the distribution of TV services in an IP network: the delivery. I briefly explain IP multicast and some of the most important multicast protocols developed over the years. In particular, I focus on the limitations of the original multicast protocols and how a new class of protocols enabled the emergence of large scale IPTV networks.

2.2.1 Why multicast?

TV broadcasting requires all viewers watch a program simultaneously, according to a predetermined schedule. In an IP network, the simplest method to send data to many receivers simultaneously is to send them multiple times from the source. This method has, however, several drawbacks. First, it is very expensive to the sender. Second, it is very inefficient as an excessive number of duplicate packets can be carried in the network links. Third, sending multiple unicasts requires the sender to know the address of each and every single receiver. For all these reasons, this simple technique does not scale.

Instead of sender replication, a better option is for the responsibility of replication to move to the network. The simplest such scheme is broadcast: network nodes replicate all broadcast packets (with some restrictions to avoid loops), thus ensuring packets are delivered to all devices on the network. This solution removes the burden from the sources but it is still very inefficient and does not scale. All nodes in the network receive the data, including those not interested.

IP networks offer an alternative solution: multicast. This mechanism provides an efficient many-to-many distribution of data making it the ideal solution to distribute TV services. The original work on IP multicast routing was by Steve Deering [62]. In his PhD thesis [64] he presented a new service model for multicast (which he called the Host Group Model) and a set of multicast routing algorithms to support that service model. Since then, the multicast problem has been extensively studied and several protocols proposed. The IP multicast model can be summarised in the following three points [58]:

- Senders send to a multicast address.
- Receivers express interest in a multicast group address.
- Routers conspire to deliver traffic from the senders to the receivers and optimise (for some definition of "optimise") packet replication.

In the next subsections I address each of these.

2.2.2 Multicast addresses

IP unicast packets are transmitted with a source and destination address, which enables routers to find a path from sender to receiver. To send multicast traffic the destination address has different semantics to a unicast address: it does not represent a particular destination. Instead, it is a group address (i.e., a logical address) that represents the set of receivers.

2.2.3 Group management

To receive multicast traffic an interested host has to inform its local router of its interest. This is done by means of a group management protocol, typically the Internet Group Management Protocol (IGMP). When a host wants to join a multicast group it programs its Ethernet interface to accept the relevant traffic, and sends an IGMP join message on its local network. This informs any local router that there is a receiver for this group now on this subnet. The local routers then arrange for the traffic destined to this address to be delivered on the subnet.

The routers periodically send an IGMP query to this multicast group to understand if there are still hosts interested in receiving multicast traffic from this group. If the host is still a member, it replies with a join message (unless any other host in the subnet does it first). In its original version [62], when a host wanted to leave a multicast group it would need to reprogramme its Ethernet interface to reject the traffic, but packets would still be sent to the subnet until the next IGMP query was sent by the local router (for which no-one would respond). Joining a multicast group was therefore quick, but leaving could be slow. IGMPv2 [74] improves over IGMPv1 by adding the ability for a host to signal desire to leave a multicast group, by means of an explicit leave message. This also avoids the need for the local router to send the periodic IGMP queries referred to above. IGMPv3 [39, 103] improves over IGMPv2 mainly by adding the ability to listen to multicast originating from a specific set of source IP addresses only. A network designed to deliver an IPTV multicast service using IGMP typically uses the basic architecture presented in Figure 2.4.

It is important to note that IGMP is utilised between the client computer and a local multicast router. A multicast routing protocol, typically PIM-SM as I explain later, is then used in the IP network to direct multicast traffic from the IPTV server to its multicast clients

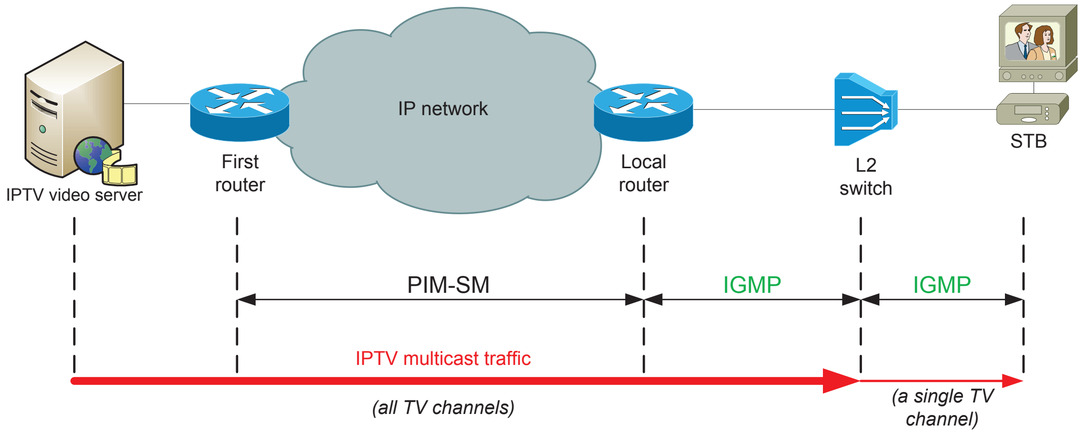


Figure 2.4: IGMP basic network architecture

(the STBs). A final detail also worth mentioning is the IGMP snooping capability some layer-2 switches possess. As its name implies, IGMP snooping [53] is the process of listening to IGMP network traffic. By using this technique the switch can maintain a map of which links need which IP multicast streams, meaning traffic can be filtered. This prevents hosts on a local network from receiving traffic for a multicast group they have not explicitly joined, all at layer 2.

This technique is especially useful for bandwidth-intensive IP multicast applications such as IPTV. Current IPTV systems distribute *all* TV channels to *all* local routers. All this traffic reaches a layer-2 aggregation switch (a DSLAM[1] in DSL networks) for channels to be distributed to as close to the user as possible, in order to reduce channel change latency. If this switch has IGMP snooping filtering capabilities, as is usually the case, it is thus possible to distribute to the Set Top Box only the TV channel the user has switched to, filtering all others. This allows the IPTV system to overcome the bandwidth limitations of access networks[2].

2.2.4 Multicast routing protocols

For multicast traffic to be delivered the routers have to build distribution trees from the senders to all receivers of each multicast group. As the senders do not know who the receivers are (they just send their data) and the receivers do not know who the senders are (they just ask for the multicast traffic) the routers have to build these trees without help from the hosts. Two generic solutions have been proposed to this problem. In the first, *flood and prune*, the senders flood their data to all possible receivers and have the routers for networks where there are no receivers to prune off their branches from the tree. In the second, *centre-based trees*, explicit distribution trees are built centred around a particular router.

2.2.4.1 Flood and prune protocols

In these protocols the sender floods traffic throughout the network. A router may receive the same traffic in different interfaces, rejecting any packet that arrives at any interface other than

[1] DSL Access Multiplexer.

[2] An aspect discussed later in this chapter.

the one it would use to send a unicast packet back to the source, a technique known as *Reverse Path Forwarding*. The router then sends a copy of each non-rejected packet out of each interface other than the one back to the source. In this way the data are received by all routers in the network. This includes those that have no hosts interested in receiving this traffic. As those routers know they have no receivers (via IGMP) they then send prune messages back towards the source to stop unnecessary traffic from flowing. The final distribution tree is what would be formed by the union of shortest paths from each receiver to the sender, i.e., a reverse shortest-path tree.

Two well-known protocols fall in this category: the Distance-Vector Multicast Routing Protocol (DVMRP) [203], a multicast extension to the Routing Internet Protocol (RIP) [139]; and the Protocol Independent Multicast, Dense Mode (PIM-DM) [2]. The main difference between these two protocols is that DVMRP computes its own unicast routing table while PIM-DM uses that of the underlying unicast routing protocol (the reason for being called *independent*).

Multicast protocols based on the flood and prune technique build efficient trees, but have problems. Sending traffic everywhere and requiring routers not on the delivery tree to store prune state is not a scalable mechanism. But for groups where most routers actually do have receivers (where receivers are *densely* distributed), this type of protocol is a good option.

2.2.4.2 Centre-based trees protocols

Rather than flooding the data everywhere, algorithms in the centre-based tree category map the multicast group address to a particular unicast address of a router. Then, explicit distribution trees centred in this router are built.

The earliest such protocol was Core-Based Trees (CBT) [20, 21], which works as follows. To join a multicast group a CBT router sends a join message towards the core router for the group. At each router on the way to the core, forwarding state is instantiated for the group and an acknowledgement is sent back to the previous router. This procedure builds the multicast tree. CBT builds *bidirectional shared* trees. Routing state is *bidirectional* as packets can flow both up the tree towards the core or down the tree away from the core, depending on the location of the source. In addition, the tree is *shared* by all sources of the group.

The main advantage of CBT is the state routers need to keep. Only routers in the distribution tree for a group keep forwarding state for that group. This protocol is therefore highly scalable, and is especially suited for sparse groups where only a small proportion of subnets have members. The main problem of CBT is core placement. Without good core placement the trees constructed can be quite inefficient.

After CBT, several related protocols were proposed that took advantage of the good scalability offered by centre-based protocols while simultaneously trying to avoid the dependency of the core and reduce the efficiency concerns associated. The most successful was undoubtedly [178] the Protocol Independent Multicast - Sparse Mode (PIM-SM) [73]. The important insight of this protocol was to realise that the problem of discovering the senders could be separated from building efficient trees.

In a similar manner to CBT, in PIM-SM when a receiver joins a group its local router sends a join message to the core (in PIM-SM, the core router is called the Rendezvous Point, or RP), instantiating forwarding state for the group. Contrary to CBT, however, this state is unidirectional. It can only be used by packets flowing from the RP towards the receiver. When a sender starts sending data it encapsulates each packet in another IP packet and unicasts it directly to the RP. These data are de-encapsulated and then flow down the shared tree to all receivers.

As in CBT, these unidirectional trees may not be good distribution trees, but at least serve the purpose of starting data flowing from the senders to the receivers. Once these data are flowing, a receiver's local router can initiate a transfer from the shared tree to a shortest-path tree by sending a source-specific join message towards the source (as the receiver now knows who the source is after receiving its data). When data starts to arrive along the shortest-path tree, a prune message is sent back up the shared tree to avoid receiving redundant traffic. The trigger to move from the shared tree to the shortest-path tree is adjustable, allowing a good compromise between tree efficiency and router state scalability. For example, it may be preferable to switch high-bandwidth multicast traffic to the shortest-path tree, as efficiency is very important in such scenario. For low-bandwidth traffic tree efficiency is less relevant and thus reducing router state with a shared tree may be preferable. Because PIM-SM can optimise its distribution tree in such way it is less critically dependent on core location.

2.3 A typical IPTV network

Unlike Internet video which runs on top of the best-effort Internet, IPTV is a provider-managed service with strict quality-of-service requirements [29]. The service provider has full control over content distribution, storage management, and bandwidth provisioning, to ensure end-to-end quality of delivery. Incumbent operator's IPTV networks are therefore "walled gardens", well provisioned to guarantee the user experience required by TV viewers [42]. In this section I present the architecture of a typical IPTV network.

2.3.1 Network topology

A traditional "walled garden" IPTV network can be split logically into three main domains — the access network, the metropolitan network, and the IP network. The IP network usually has a two-level, hierarchical structure [79]: the regional network (sometimes called the edge or gateway) and the core (or backbone). This is shown in simplified form in Figure 2.5[1].

In an IPTV system, live TV streams are encoded in a series of IP packets and delivered through an IP network to the residential broadband access network. The IPTV head-end, the primary source of television content, digitally encodes video streams received externally (e.g., via satellite) and transmits them through a high-speed IP network. The core network comprises a small number of large routers in major population centres. The core routers of any one

[1] This figure will be used as the *IPTV reference architecture* throughout this dissertation.

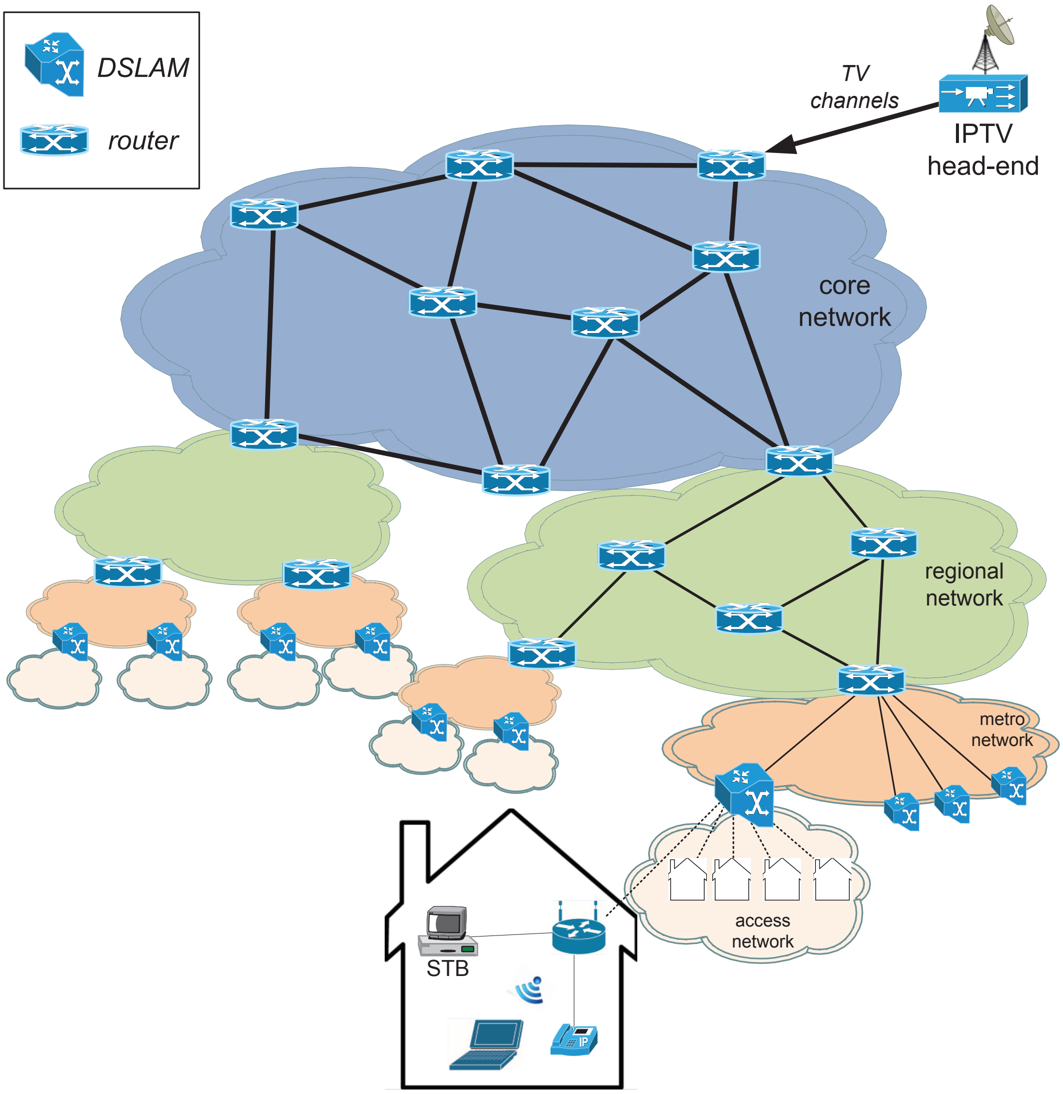


Figure 2.5: IPTV reference architecture

network are often highly meshed, with high-capacity WDM fibre links interconnecting them. The topology of the core typically consists of a set of nodes connected by high bandwidth 10 Gbps and 40 Gbps links [14]. In the regional network routers are normally lower-end routers with high port density, where IP customers get attached to the network. These routers aggregate the customer traffic and forward it toward the core routers [79].

The metro network serves as the interface between the regional network and the access network. Metro (typically Ethernet) switches concentrate traffic from a large number of access nodes and uplink to two or more regional routers (to provide redundancy). The access network connects each home to one of the edge switches in the provider's network. There is a wide variety of access technologies: from ADSL (Asymmetric Digital Subscriber Line[1]) to fibre-based solutions (FTTx[2]) to wireless options. The bandwidth of each access link is limited, and it

[1] ADSL is a type of DSL technology that offers higher bit rates toward the customer premises (downstream) than the reverse (upstream).

[2] This is a generic term for any broadband network architecture using optical fibre to replace all or part of the

varies with the technology: around 20 Mbps for ADSL[1], but increasing to the hundreds of Mbps as optics comes closer to the home. IPTV is currently being rolled out predominantly by incumbent operators [201], so ADSL has been the main access network used to distribute content to customer premises. Since IPTV is naturally agnostic to the layers below IP, IPTV deployments from other providers are expected in the future [191]. In ADSL, the copper pairs originally installed to deliver a fixed-line telephone service are now used to also deliver a broadband service [45]. These copper pair-based access technologies are limited in capacity by usable bandwidth and reach, so typically only the TV channel the user is watching is delivered to the STB (Set Top Box) at any one time. This is the main reason why channel zapping delay is high, as is explained in Chapter 5. For these access technologies, the terminal unit (DSLAM in ADSL networks) commonly takes the form of a layer 2 switch with IGMP snooping capability, as mentioned before, with line cards appropriate to the access technology facing the subscriber.

Finally, inside a household, a residential gateway connects to a modem and one or more STBs, receiving and forwarding all data, including live TV streams, STB control traffic, VoIP and Internet data traffic. Finally, each STB connects to a TV.

In this dissertation I use the terminology shown in Figure 2.5 to determine events at different aggregation levels. Namely, a DSLAM serves multiple STBs, a regional-metro router serves multiple DSLAMs, a regional-core router serves multiple regional routers, and, lastly, the IPTV head-end serves content to all core routers.

2.3.2 Network protocols

As explained in the previous section, in IPTV systems the TV head-end injects live TV streams encoded as IP packets to the IP network core. The TV channels are distributed from the TV head-end to edge nodes (DSLAMs in Figure 2.5) through bandwidth-provisioned multicast trees, for efficient distribution. By far, the most common multicast routing protocol [178] is PIM-SM [73]. Current networks use *static* IP multicast within a single network domain. By static multicast I mean all receivers are known beforehand, and no new group members are allowed to join — we have a *static set of receivers* for all TV content. Again referring to Figure 2.5, this means *all* DSLAMs join *all* multicast groups (thus receive content from *all* TV channels). This is despite the fact that particular channels may have no viewers at particular time periods. The only section of the network which is not static in this sense is between the DSLAM (or local router, in case the layer 2 switch has no snooping capabilities) and the STB, due to the limited bandwidth resources of access networks referred above. Therefore, when a user switches to a new channel, the STB issues a new channel request towards the network.

Distributing unwanted traffic (the TV channels for which there are no viewers) in the network may seem strange as it represents an inefficient use of the network's resources, with plausible energetic and monetary costs. But there are good reasons to do so. First, IPTV providers want

usual copper local loop used for last mile telecommunications. Examples includes FTTH (Fibre To The Home) and FTTC (Fibre to the Cabinet).

[1] A figure that varies with the quality of the twisted-pair local loop and its length (i.e., the distance from the household to the local exchange).

to guarantee that no control traffic clogs their networks. Second, they want to be assured that propagation delays for join requests — when users switch to a new TV channel — are modest, in order to minimise channel change delay. So TV channels have to be distributed to as close to the users as possible. But it is anyway possible to increase the resource (and associated energy) efficiency of the network without jeopardising service quality, as I explain in Chapter 6.

2.3.3 IPTV services

Alongside conventional TV, current IPTV providers often support additional features, some of which are not offered by traditional TV services. For example, many add sophisticated Electronic Program Guides and Set Top Boxes with extra functionality, such as recording capabilities. Depending on the provider, IPTV users can also enjoy many advanced features such as on-line gaming, chatting, and other web services on their TVs.

In terms of the TV content itself, most systems today distribute Standard Definition (SD) TV channels using MPEG-2, requiring 4 Mbps guaranteed bit rate per channel [5]. As optical fibre comes closer to the customer premises, higher capacities are becoming available in the access link and several TV broadcasters are now offering High Definition (HD), requiring around 20 Mbps per channel [29]. In the future, ultra high definition systems are also expected. Panasonic, for instance, presented recently at the Consumer Eletronics Show (CES) a 150-inch plasma TV set with 4k resolution [210] (Figure 2.6). In the next decade, as prices plunge, this type of device may become common in our living rooms, creating new market opportunities. These very high resolutions require hundreds of Mbps per TV channel [83], significantly increasing the bandwidth demands on the network and increasing its operational complexity. This may make current static multicast systems prohibitive and justify the use of resource and energy-efficient distribution schemes, as the ones proposed in Chapters 6 and 7 of this dissertation.

2.4 Challenges for IPTV providers

In a fiercely competitive market as that of the telecommunications sector, IPTV service providers have several challenges to address. These include financing difficulties, particularly in a time of economic crisis, the choice of the best business plan to supplant competitors, and how to keep up with the recent technological advances and hurdles. In this dissertation, I address the latter.

IP multicast offers the point-to-multipoint delivery mechanism necessary for the efficient distribution of TV services. However, the original service model has some issues that for some time have stalled the widespread use of multicast. In a paper published a decade ago, Diot *et al.* [67] identified some of these issues:

- The multicast service model does not consider group management. This includes authorisation for group creation and for transmission, billing policy and address discovery.

- Security is also a problem. Authentication is not mandatory, and scalable key management for encryption and data integrity is still an issue.

Figure 2.6: Panasonic's 150" plasma TV presented at CES 2008 [210]

- Distributed multicast address allocation is another concern. Because the current multicast address space is unregulated, nothing prevents applications to sending data to any multicast address.

- Finally, there is no robust support for network management.

These problems still exist today but, as explained in the previous section, current IPTV networks are provider-managed services. Being closed networks under the control of a single entity eliminates the four problems identified by Diot *et al.* Still, other issues persist. For example, IPTV offerings should match the level of quality of service that customers are accustomed to from other TV service providers. Customers would not tolerate poor quality of picture and sound. But the delivery of video is challenging. In order to be successfully decoded in the Set Top Box, the video stream has to arrive at a known and constant bit rate, in sequence, with minimal jitter or delay. The problem is that IP networks are "best effort", susceptible to lost or dropped packets as bandwidth becomes scarce and jitter increases. In addition, video streaming requires high data rates, so efficiency in distribution is a major concern. Also, the access network has been a bottleneck until recently. These problems have been mostly solved:

1. The use of traffic prioritisation techniques and bandwidth reservation for IPTV traffic assures that quality of service requirements are guaranteed in these closed networks.

2. The emergence of scalable multicast protocols — in particular centre-based ones, notably PIM-SM — provide the means for an efficient distribution of TV services.

3. The last mile bandwidth bottleneck has been broken in most developed countries, with enhancements to DSL technology, and with the recent trend to bring fibre closer to the home [201]. This, coupled with the progress of video codecs, such as MPEG-2 and MPEG-4, to compress video content, and with the increased processing power and storage capacity of (cheap) STBs, enables the distribution of TV on current IP networks.

The fact that multicast IPTV has been fully deployed with success by several telecom companies — examples include AT&T, Telefonica, France Telecom, China Telecom, etc. — is evidence that these technological advances made it cost effective to deploy and manage a multicast network.

But IPTV service providers still face important technological challenges. A relevant one is still related to the provision of a level of service at least as good as its competitors. This certainly includes offering new, added-value services, but it is also fundamental to guarantee the quality of experience conventional TV users expect. IPTV service's high channel change delay, covered in Section 2.1.1, is still a thorn in IPTV provider's side in this respect. This is the first problem I explore in this dissertation, in Chapter 5. Another technological challenge is to maintain an operationally cost and energy efficient network in face of the evolution of IPTV services. This is the second problem I address. As explained in Section 2.3, static multicast is inefficient. A dynamic multicast solution also brings issues, such as network scalability and service quality, with more signalling messages on the network, frequent router state changes requiring additional processing, and an increase in channel switching delay (in certain periods some TV channels may not be distributed close to the users requesting them). Chapters 6 and 7 are devoted to this compromise between network efficiency and service guarantees.

Chapter 3

State of the art

The distribution of TV over IP networks provides an attractive business opportunity for telecommunications service providers. The emergence of major players in this market is an evidence of this fact. Early IPTV deployments have demonstrated that significant technical challenges must be overcome to ensure the service is compelling to users and competitive with other provider's offerings. As a consequence, IPTV has posed interesting research questions that were the subject of several papers over the past few years. In this chapter I present research on IPTV and also on topics closely related to the IPTV challenges identified in the previous chapter.

The chapter opens with research on IPTV network measurements. With the recent deployment of IPTV networks a number of papers measuring and characterising IPTV traffic have been published. The analysis of real IPTV workloads led to a clearer understanding of how people watch TV and how this impacts the network. The findings from these studies offered clues that led to some of the techniques I investigate in Chapters 5, 6, and 7. The next three sections focus on the particular set of problems addressed in this dissertation, namely, IPTV channel change delay and resource and energy-efficiency of IPTV networks. First, I present a short survey on the research done to date to mitigate the high channel change delays that occur in IPTV systems. Afterwards, I devote a section to energy efficiency on networks, a topic that usually goes under the label "green networking". This chapter closes with a summary of work on the integration of optical switching with electronic routing in IP networks. This is a technique I explore in Chapter 7 with the objective of increasing the energy efficiency of IPTV networks.

3.1 Measurement of IPTV systems

The traditional methods used to assess TV viewing habits, employed by companies such as Nielson Media Research [147][1], are based on monitoring of a sample of representative users in order to extrapolate their behaviours to the entire population. The bidirectional communication in IPTV systems offers new possibilities in this space, giving more visibility to viewers activities across an entire network. The availability of IPTV workloads from large-scale IPTV systems

[1] Nielsen Media Research [147] measures media audiences, providing a wide set of statistics on TV viewing and program ratings based on a sample of population.

can therefore be quite useful in understanding TV viewing habits. In order to comprehend how people watch TV and how the network copes with the addition of this new service, a number of empirical studies analysing IPTV traffic have been performed.

Cha *et al.* [42] presented the first analysis of IPTV workloads based on network traces from one of the world's largest IPTV systems[1]. The authors characterised the properties of viewing sessions, channel popularity dynamics, geographical locality, and channel switching behaviours. They discussed the implications of their findings and explicitly mentioned the support needed for fast channel changes, a problem I address in Chapter 5:

"The design of a system that supports fast channel switching (...) is imperative to both improving user experience and minimising the impact in the network." [42]

By means of simulations using the same dataset, in [43] these authors consider the limitations of current IPTV architectures based on static multicast distribution. They proposed the integration of P2P distributed systems into the Set Top Boxes, thus forming a cooperative P2P and IP multicast architecture. In this dissertation I also look at the problem of using static multicast, by proposing a scheme that is a compromise between static and dynamic multicast in Chapter 6.

Qiu *et al.* have also analysed channel popularity in the context of IPTV [161]. In this paper the authors captured the channel popularity distribution and its temporal dynamics. Later, the same researchers extended this work with the characterisation and modelling of aggregate user activities in an IPTV network [160]. For both studies they also used real data from an operational nation-wide IPTV system[2]. Their findings overlap with those of [42]. In addition, in [160] the authors generalised the analysis and developed a series of models for capturing the probability distributions and time-dynamics of user activities. Lastly, they also combined these models to design an IPTV workload generation tool.

Another empirical study of an IPTV network was presented by Mahimkar *et al.* [137]. The authors focused on characterising and troubleshooting performance issues on the largest IPTV network in North America[3]. These researchers developed a diagnosis tool capable of detecting and localising regions in the IPTV network experiencing serious performance problems.

3.2 A survey on techniques to reduce IPTV channel change delay

Channel change delay is one of the most severe problems affecting IPTV deployment, and for that reason a good amount of research was done on this field so far. In this section I present a brief survey of the proposed solutions to this problem in the literature.

[1] They used a dataset from Telefonica's IPTV service Imagenio, the same dataset I use in the current study.

[2] In this case, AT&T's.

[3] Again, AT&T.

3.2.1 Simple optimisations using pure multicast

An obvious way to reduce zapping delay is to encode the video stream with a higher frequency of I-frames. However, as explained in Section 2.1, such scheme would significantly increase the storage needs at the video server as well as the bandwidth needed to offer the service. This is therefore not a practical solution.

More feasible solutions include the optimisation of channel streaming and playout. Kopilovic and Wagner [129] have shown that it is possible to optimise channel streaming with respect to initial buffering without increasing the bandwidth. This way they set a limit on what is achievable by pure multicast without additional infrastructure. Kalman *et al.* [118, 119] have shown how adaptive media playout — the variation of playout speed of media frames depending on channel conditions — allows the client to buffer less data, thus introducing less delay, for a given level of protection against buffer underflow. In this scheme, the client varies the rate at which it plays out audio and video according to the state of its playout buffer. When the buffer occupancy is below a desired level, the client plays the media slowly, generating unnoticeable latency. This latency is then eliminated with periods of faster-than-normal playout. This scheme is similar to the adaptive piggyback techniques used to reduce I/O bandwidth in a Video on Demand (VoD) server [4]. In fact, several techniques used in the past to reduce VoD start-up delay [102] are now being transposed to mitigate channel change delay in IPTV.

3.2.2 Proxy server with boost streams

Most commercial solutions to the channel change delay problem attempt to ensure that an STB that is trying to join a new channel gets an auxiliary stream that starts with an I-frame and then offers some kind of mechanism to switch over to the main multicast stream. This is probably the most common fast channel change mechanism, and is used, for example, by the Windows Media Platform [141]. This solution, illustrated in Figure 3.1, requires the introduction of dedicated zapping servers in the network. Simultaneously with the request to joint the multicast group, the Set Top Box (STB) requests the channel from the zapping server (step 1 in the figure). The zapping server then transmits a unicast burst with a delayed stream that starts with an I-frame (step 2). This stream is sent at a higher than usual bit rate, for the play-out buffer to fill quickly. Thus the two main components of zapping delay are either removed — there is no waiting for the I-frame, since this is the first to be received — or significantly reduced — buffering time is lower. When an I-frame from the multicast flow finally arrives, the STB terminates the unicast flow and switches to the multicast one (step 3). In the figure the servers are co-located with the regional-metro routers, but they could be placed at other locations. There is a trade-off between the number of servers needed and the performance of the system. Fewer servers in the network means more requests to respond per server, with a possible increase in response time (i.e., in channel change delay).

This solution is expensive because it requires many dedicated servers to be added to the network. To mitigate this problem Begen *et al.* [27, 28] proposed a unified approach that can be used both to repair lost packets in real time and reduce the zapping delay. So, if you already have

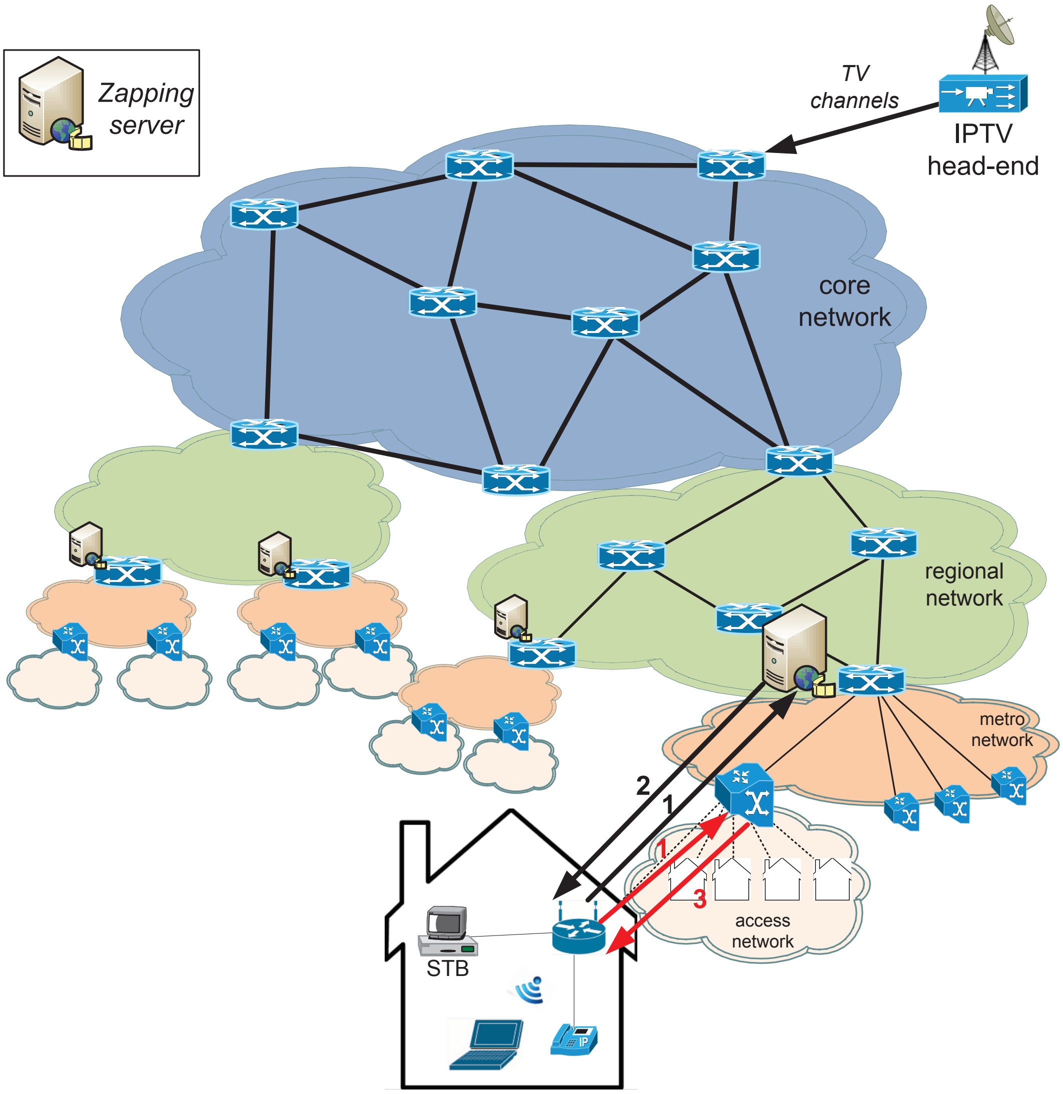


Figure 3.1: Zapping servers introduced in the IPTV network to mitigate the high channel change delay

a dedicated server to deal with lost packets, you may well use the same for reducing channel change delay. The authors proposed to use the unicast retransmission support of RTP [171] and RTCP [151], conventionally utilised to recover lost packets, to accelerate channel changes. This feedback mechanism is used to provide the key information needed by receivers to start processing the data prior to joining the multicast session.

In the steady state, multicast reduces IPTV traffic volume. But channel surfing disrupts this steady state when boost stream solutions are used in the network. The network sends the unicast boost stream to make channel changes fast superimposing an additional demand on top of the steady state demand. This demand is proportional to the number of users concurrently initiating a channel change event. Flash crowds of channel changes (when channel changes are correlated, for example at the completion of a popular program) place significant demands on the network and video server resources. In [186] D. Smith analysed this problem by constructing a mathematical model to determine the bandwidth demand of a channel change mechanism where unicast streams at higher than usual bandwidth are sent when viewers are changing channels. This model quantifies the extra bandwidth consumed by channel surfing. Smith looks particularly at commercial breaks since these periods are more disruptive to the steady state demand. By assuming a simple exponential distribution for the time between channel changes, the author finds that the peak demand during a commercial break is twice the steady state multicast demand.

Since these unicast solutions have this scalability problem, a few multicast-based solutions were proposed recently. Sasaki *et al.* [176], for instance, propose the STB to receive an additional multicast stream together with the original stream. This is simply a delayed version of the original stream, resulting in the buffering time being halved. If other viewers switch at the same time to the same channel, multicast suppresses any duplication of packets. A similar proposal was presented by Banodkar *et al.* In their proposal [23] the user joins a secondary multicast stream in association with the multicast of the regular quality stream. This secondary stream is of lower quality (it contains only I-frames, therefore it is not full motion video). During a channel change event, the STB does a multicast join to this secondary stream, allowing the user to experience smaller display latency. In the background, the playout-buffer of the original full quality stream is filled, and when the play-out point is reached this full quality stream is displayed and the transition is complete.

The most interesting multicast assisted zap acceleration system is probably the one proposed by Bejerano and Koppol [30]. The main objective of their system is to reduce the FID (First I-Frame delay): time until the following I-frame is received. For this aim they also deploy an additional server in the provider network (they call it a zapping accelerator, ZA). The ZA generates several time-shifted replicas of each TV channel media stream, and each of these replicas is identified by a unique multicast group. Also, all STBs are subscribed to a meta-channel, a low bandwidth multicast group with information on the earliest replica with an I-frame for each TV channel, to help the STB choosing the best one. When the user switches to a new channel, it joins the multicast group of this replica. This way the zapping delay experienced is lower and deterministic — it is bounded by the time-shift between two successive replicas (more replicas mean a reduced time-shift, thus a lower zapping delay). To reduce bandwidth consumption these replicas are sent only when there are users watching the channel, and several users can be served simultaneously by the same replica (since they are multicast streams). After a while the STB switches from the replica to the main stream. As usual, to allow the STB to perform this migration transparently the replicas are sent at a higher data rate. The fact that we need five or six of these higher bit rate replicas to guarantee a low zapping delay will increase the bandwidth usage by each TV channel in the network by about an order of magnitude. There is therefore a clear tradeoff between the number of ZAs and bandwidth consumption.

These multicast solutions are more scalable since the server and network load depend not on the number of viewers zapping, but rather on the number of TV channels the users are zapping to. This scalability enhancement allows a single server to serve more users, which means fewer servers are needed in the network, reducing costs. Also, with a multicast solution bandwidth consumption and server load are lower even during flash crowds of channel changes.

3.2.3 Video coding techniques

Other type of solutions to the channel change delay problem include altering or extending the video coding techniques that were the subject of section 2.1. One such technique is to include a picture-in-picture channel in the stream [65]. This channel has a lower bit rate (and resolution) than the regular channel. It is constructed with a small GOP size. When the STB tunes to the new channel it first tunes to this channel, which is temporarily displayed until the STB has received an I-frame from the regular channel and the play-out buffer has been filled to an acceptable level. A lower spacial resolution is displayed during the zapping period, and the system requires a higher bit rate because both channels are sent simultaneously. A similar scheme is presented in [35], where Boyce and Tourapis proposed embedding a lower resolution stream into a normal resolution one for each channel. The I-frames in the lower resolution scheme occur more frequently, and when the user switches to this channel these streams are decoded first, and only after a while is the normal stream used for playout.

One of the problems of the previous approaches is bandwidth inefficiency, so the authors of [115] targeted this particular point in their proposal SFCS (Synchronisation Frames for Channel Switching). In the schemes presented before, in order to switch from the lower quality stream to a higher quality stream (bitstream switching) it is necessary to wait for an I-frame, as these are the stream synchronisation points for the client. As explained in section 2.1, the drawback of using I-frames is that, since temporal redundancy is not exploited, they require a much larger number of bits than P-frames at the same quality. For this reason, I-frames are an inefficient solution when the actual requirements for stream synchronisation are taken into account. To be more bandwidth-efficient the authors proposed using switching frames (SP and SI-frames) instead of I-frames. The concept of switching frames was introduced in [121]. The main feature of SP-frames is that identical frames can be reconstructed even when different reference frames are used for their prediction. This allows them to replace I-frames in applications such as bitstream switching, as in this case. Albeit providing similar functionality, SP-frames have significantly better coding efficiency than I-frames: since they utilise motion-compensated predictive coding, they require fewer bits than I-frames to achieve similar quality. The same authors also proposed an extension to SFCS [114] to be compatible with encoding/decoding systems that do not support SI/SP-frames.

In [117] Joo *et al.* proposed an algorithm to control channel zapping time by adjusting the number of broadcast IPTV channels that are distributed close to users and the number of I-frames inserted into each channel, based on the user's channel preference information. They thus consider two variables — broadcasting channel distribution (positioning channels according to their popularity) and video encoding structure (adding extra I-frames to the normal video frames to decrease video decoding delay). With their algorithm they achieve an effective trade-off between channel zapping time and network utilisation.

The main problem of all these schemes is the increase in encoder complexity. Besides this, most require additional bandwidth. In addition, the user will experience a brief lower quality period immediately after zapping, and the transition from a low to a high-resolution channel can frequently cause undesirable glitches.

3.2.4 Predictive pre-joining of TV channels

As explained in Chapter 2, in IPTV systems typically only the channel the user is watching is delivered to the Set Top Box at any one time, due to bandwidth limitations in the access network. When a user switches to a new channel, the STB has to issue a new channel request towards the network. The fact that the channel the user switched to is not available in the STB is the main reason for a high channel change delay: the STB has to join the multicast group from this channel, synchronise with the video content and buffer some packets before play-out. In predictive pre-joining schemes, each STB simultaneously joins additional multicast groups along with the one requested by the user, thus anticipating future user behaviour. These schemes are thus based on the prediction of the next TV channels the user will switch to. If the prediction is right, the user will experience a small zapping delay, as the channel switched to is already synchronised in the STB.

The first paper proposing pre-joining of TV channels was by Cho *et al.* [51]. In their proposal the additional channels are simply the channels adjacent to the channel being watched. The main problem of this work was that no evaluation was given. The paper offered a mere description of the idea. Furthermore, without modifications their scheme would be very inefficient. The adjacent channels would be sent continuously with the requested channel. So, in the periods when the user is settled in a channel (i.e., not zapping), the adjacent channels would be transmitted in the access link consuming precious bandwidth.

Two other papers [81, 189] proposed a similar scheme, but solved the two problems of the original. First, they considered delivering the adjacent channels for a finite period only, thus their schemes are bandwidth-efficient. Also, they *evaluated* their proposals by developing an analytical model to investigate the performance of each of their schemes. The main problem of these two papers is that some of the assumptions they made in building their simple models have been proved wrong recently: by analysing real IPTV datasets, recent studies [160, 166] have shown that channel surfing behaviour should not be modelled with the simple Poisson processes the authors of [81, 189] (and others) used. The constant-rate Poisson models generally used as workload model are not capable of capturing the high burst of channel switches at particular periods[1]. For evaluation of IPTV studies it is therefore important to use actual IPTV trace data or reliable models based on empirical data, such as the one proposed by Qiu *et al.* [160].

Recently more sophisticated pre-joining schemes have been considered. Oh *et al.* [150] presented an hybrid scheme combining pre-joining and reordering. The authors considered two pre-joining schemes: a first where the adjacent channels, as above, are pre-joined, and a second where the most popular channels are pre-joined. They then combined these with a channel

[1] I return to this issue in the next chapter to prove this fact.

reordering scheme the same authors had proposed before [131], where popular channels are clustered together in the linear search sequences. Another recent paper proposing a pre-joining method for the same purposes was presented by Lee *et al.* [130]. Their scheme is based on both button and channel preference. The authors also included a method to determine the most efficient number of channels to pre-join. A small number of channels is pre-joined during viewing periods, with more channels being pre-joined during zapping periods. In these two papers simple mathematical models were also used for evaluation, with the same problems described before.

I target this category of techniques — predictive pre-joining of TV channels — in Chapter 5 of this dissertation.

3.3 Green networking

Since the seminal paper by Gupta and Singh [93], presented at SIGCOMM in 2003, the subject of *green networking* has received considerable attention. In recent years, valuable efforts have been devoted to reducing unnecessary energy expenditure. Big companies such as Google, Microsoft, and Amazon, are turning to a host of new technologies to reduce operating costs and consume less energy [122]. Google, for example, is planning to operate its data centres with a zero carbon footprint by using, among other things, hydropower, water-based chillers, and external cold air to do some of the cooling.

Several approaches have been considered to reduce energy consumption in networks. These include:

- The design of low power components that are still able to offer acceptable levels of performance. For example, at the circuit level techniques such as Dynamic Voltage Scaling (DVS) and Dynamic Frequency Scaling (DFS) can be used. With DVS the supply voltage is reduced when not needed, which results in slower operation of the circuitry. DFS reduces the number of processor instructions in a given amount of time, thus reducing performance. These techniques can reduce energy consumption significantly. Zhai *et al.* [219] have shown that theoretically the power consumption decreases cubically when DVS and DFS are applied jointly. As with a reduced frequency the time to complete a task increases, the authors show this is translated into an overall quadratic reduction in the energy to complete a task.

- Consuming energy from renewable energy sources sites rather than incurring in electricity transmission overheads [68], thus reducing CO_2 emissions.

- Designing new network architectures, for example by moving network equipment and network functions to strategic places. Examples include placing optical amplifiers at the most convenient locations [195] and performing complex switching and routing functions near renewable sources [68].

- Using innovative cooling techniques. Researchers in Finland, for instance, are running servers outside in Finnish winter, with air temperatures below -20 °C [155].

- Performing resource consolidation, capitalising on available energy. This can be done via traffic engineering, for instance. By aggregating traffic flows over a subset of the network devices and links allows others to be switched off temporarily or be placed in sleep mode [93]. Another way is by migrating computation, typically using virtualisation to move workloads transparently [33, 61]. Computation is migrated from several lightly loaded devices or servers to a consolidation server, and then the equipment that is freed up can be turned off.

- Reducing router processing, for example by switching transit traffic[1] at the optical layer [14]. This technique is the basis of the proposal I present in Chapter 7.

In the past few years numerous researchers have used these techniques to build greener networks. In the rest of this section I present a brief overview of the state of the art on this topic.

3.3.1 Fundamentals

Before attempting to reduce energy consumption, it is important to know the fundamentals in order to identify where significant savings can be obtained. In a series of two papers, Ronald Tucker explored the fundamental limits on energy consumption in optical communication systems and networks. In Part I [194] the author focused on the lower bound on energy in transport systems. Among other results, he concluded that it is possible to minimise the total energy consumption of an optically amplified system by locating repeaters strategically. In Part II [195] Tucker explored the lower bound on energy consumption in optical switches and networks, confirming a previous finding [14] that the energy consumption of the switching infrastructure is larger than the energy consumption of the transport infrastructure. Still on the fundamentals, Baliga *et al.* [15] suggested that the ultimate capacity of the Internet might eventually be constrained by energy density limitations and associated heat dissipation considerations rather than the bandwidth of the physical components.

Energy-awareness has increased in proportion with the emergence of recent studies that quantified the energy consumption of networks. In [14], for example, the authors presented a network-based model of power consumption in optical IP networks and used this model to estimate the energy consumption of the Internet. They estimated that the Internet currently consumes around 0.4% of electricity consumption in broadband-enabled countries, but that this figure is on the rise. Other studies have suggested an important increase of core network energy consumption. For instance, Tucker *et al.* [17, 19, 197] developed simple energy-consumption models in a series of papers and reached the overall conclusion that at low bit rates power consumption is dominated by the access network. However, as access rates to users increase, the energy consumption in routers, particularly core routers, will become significant and eventually dominate.

[1] Traffic not destined to the node under consideration.

3.3.2 No work? Then, sleep

One of the most common techniques to save energy is to shutdown network equipment (or some of its constituent components) whenever possible. In the pioneering work on green networking [93] the authors discussed the impact on network protocols of saving energy by putting network interfaces and other router and switch components to sleep. They considered changing routes during low activity periods so as to aggregate traffic along a few routes only, while allowing devices on the idle routes to sleep. In this position paper the authors concluded that sleeping was indeed a feasible strategy. Later, the same authors have also examined the feasibility of putting various components on LAN switches to sleep during periods of low traffic activity [92]. Based on traffic collected in their LAN, they concluded that sleeping is feasible in a LAN environment with little impact in other protocols, thus enabling energy savings.

Gupta and Singh continued their work on energy conservation in Ethernet LANs, and in [94] they proposed methods that allow for detection of periods of inactivity in these networks to obtain energy savings with little impact on loss or delay. Using real-world traffic workloads and topologies (from Intel Enterprise network), Nedevschi *et al.* [145] have also shown that such simple schemes for sleeping can offer substantial energy savings.

Other research works follow the same line. Chiaraviglio *et al.* [50] considered a realistic IP network topology and evaluated the amount of energy that can be potentially saved when nodes and links in the network are turned off during off-peak periods. They also proposed a simple algorithm to select the network equipment that must be powered on in order to guarantee the service. Fisher *et al.* [77] developed and evaluated techniques to save energy in core networks by selectively powering down individual cables of large bundled links during periods of low utilisation. Idzikowski *et al.* [108] estimated potential energy-savings in IP-over-WDM networks achieved by switching off router linecards in low-demand hours. All these works show that it is possible to achieve significant energy savings using such simple techniques.

3.3.3 Little work? Then, slow down: Adaptive Link Rate

It is recognised by the research community devoted to green networking that for systems and networks to be energy-efficient, energy proportionality should become a primary design goal [24]. An efficient device should consume energy proportionally to its output or utility. Unfortunately, most equipment is not energy-proportional. Fortunately, a serious effort is being made in the community to change the current situation.

A common example is the Ethernet. Ethernet power consumption is independent of link utilisation. Idle and fully utilised Ethernet links consume about the same amount of power. As the average utilisation of desktop Ethernet links is very low, in the range of 1 to 5 percent [52], and as this situation is likely to persist [149], the current state of affairs is undesirable. A way to improve Ethernet so that energy use is proportional to link utilisation is Adaptive Link Rate (ALR). ALR, proposed by Gunaratne *et al.* [90], is a means of automatically switching the data rate of an Ethernet link to match link utilisation. Based on simulation experiments using actual and synthetic traffic traces these authors have shown in [91] that an Ethernet link with ALR can operate at a lower data rate for over 80% of the time, yielding significant energy savings without compromising the quality of service.

3.3.4 Want to sleep? Then, delegate: Proxying

Desktop computers in enterprise environments consume a lot of energy in aggregate while still remaining idle much of the time. The question many researchers have asked in the past few years is how to save energy by letting these machines sleep while avoiding user disruption. To reduce energy waste by idle desktops the typical approach is to put a computer to sleep during long idle periods, with a proxy employed to reduce user disruption by maintaining the computer network's presence at some minimum level. The proxy can be co-located within the host (e.g., on an Ethernet NIC), or in another device (e.g., a LAN switch). The problem with this approach is the inherent trade-off between the functionality of maintaining network presence and the complexity of application specific customisation.

The simpler such mechanism is the Wake on LAN technology [10]. This is a mid-1990s industry standard that makes it possible for an Ethernet adapter to wake-up a sleeping desktop computer using a specially defined packet (a "magic packet"). A more sophisticated approach was proposed by Christensen *et al.* [52]. The authors propose a proxying Ethernet adapter that can wake-up a desktop computer in sleep mode when its resources are needed and otherwise handle routing protocol messages without waking up the computer. Thus the computer maintains its presence on the network without being fully powered on at all times. An in-depth evaluation of the potential energy savings and the effectiveness of proxy solutions was performed by Nedevschi *et al.* [144]. They considered two types of proxy. A simple one that performs automatic wake up triggered by a filtered subset of the incoming traffic, and a more elaborate one which incorporates application-specific stubs that allow it to engage in network communications on behalf of applications running in the machine that is now sleeping. Other examples of application-specific stubs exist for BitTorrent [11] and Gnutella [116]. A proxy prototype was also presented recently by Agarwal *et al.* [3]: Somniloquy. This proxy is an augmented network interface that allows PCs in sleep mode to be responsive to network traffic. Somniloquy achieves this functionality by embedding a low power secondary processor in the network interface.

Another approach to save desktop energy is by virtualising the user's desktop computing environment as a virtual machine, and then migrating it between the user's physical desktop machine and a VM (Virtual Machine) server. This is the idea behind Litegreen [61]. With this system, the user's desktop environment is "always on", maintaining network presence even when the user's machine is switched off. The idle desktops are consolidated on the server. A similar scheme is proposed by Bila *et al.* [33]. While Litegreen migrates the VM's entire memory state to the consolidation server, these authors propose partial migration, with only the working set of the idle VM being migrated. This allows an almost instantaneous and low energy cost migration.

3.3.5 Change paradigm: energy-aware infrastructures

A consensus exists in the green networking community: it is necessary to introduce energy-awareness in network design. Certainly, this has to be achieved without compromising either the quality of service or the reliability of the network. A good amount of effort has been therefore devoted to the design of novel energy-aware infrastructures. In some cases, new networking paradigms are being explored.

Until recently power consumption has been dominated by access networks [17]. As a consequence, some research has been devoted to this specific part of the network. To achieve power-savings in DSL networks, for example, Cioffi *et al.* [55] and Tsiaflakis *et al.* [192] proposed dynamic spectrum management (DSM). DSM improves DSL networks by tackling the crosstalk problem[1]. Typically, DSM algorithms focus in maximising data rates. Tsiaflakis *et al.* [192] extended these algorithms by incorporating energy efficiency as an objective. Panarello *et al.* [153] followed a different approach. They proposed a combined congestion control and rate-adaptation scheme for Internet access nodes which allows them to reduce energy consumption.

The energy consumption of big data centres ("clouds") has also been a hot topic of research. Heller *et al.* [98] presented a power manager and optimiser, ElasticTree, which dynamically adjusts the set of active network elements — links and switches — to satisfy changing data centre traffic loads. This introduces energy proportionality into the network even though individual network devices are not energy-proportional. Most studies on green cloud computing focus on the energy consumed in the data centre. However, the increase in network traffic that results from moving computation to the cloud may also have an impact in energy consumption. This is the topic of [18]. In this paper, Baliga *et al.* presented an analysis of energy consumption in cloud computing. By building mathematical models based on power consumption measurements and published specification of representative equipment, the authors showed that energy consumption in transport and switching can represent a significant percentage of the total energy consumption, and thus should be carefully considered. The authors argued that under some circumstances cloud computing can consume more energy than conventional computing.

A change in network architecture paradigms from host-oriented to content-centric networking (CCN) created new possibilities for energy-efficient content dissemination. The main difference between a CCN router and an IP router is that the former supports name-based routing and caching for content retrieval [112]. These content-caching capabilities can significantly reduce redundant content transmission and consequently avoid energy waste. A study by Lee *et al.* [132] shows that CCN is capable of outperforming conventional Content Distribution Networks (CDNs) and P2P networks in terms of energy-efficiency. The authors used a publicly available traceroute dataset to validate their claim.

Several proposals to change the current client-server paradigm of Video on Demand (VoD) services have been proposed recently. Valancious *et al.* [200], for example, proposed a new way to deliver VoD based on the Nano Data Center (NaDa) platform. NaDa uses ISP-controlled home

[1] The crosstalk is an electromagnetic interference generated by different lines operating in the same cable bundle.

gateways (Set Top Boxes) to provide computing and storage and adopts a managed peer-to-peer model to form a data centre infrastructure with these devices. The authors claim to achieve energy savings by analysing a set of empirical VoD access data. The fact that they are reusing already committed baseline power on underutilised gateways and that they avoid cooling costs is the justification for these results. Feldmann *et al.* [72] also explored the energy trade-offs between P2P, data centre architectures and CDNs in the context of Internet TV, and reached a somewhat different conclusion. Their results showed that P2P, albeit capable of reducing the power consumption of the service provider, increases the overall energy consumption. The reason is that P2P applications push the energy use out of the data centres and into the homes of content consumers (thus migrating the problem). Another paper on energy efficiency of VoD networks is by Baliga *et al.* [16]. The authors built an energy consumption model for these networks based on specifications of commercial equipment. Their main conclusion is that to have an energy-efficient architecture popular new-release movies should be widely replicated throughout the network (as power consumption of transmission dominates over storage), and progressively withdrawn to fewer data centres as their usage declines (as power consumption of storage becomes dominant).

The last type of proposals in this category includes changes to current network applications. As an example, Blackburn and Christensen [34] proposed changes to the BitTorrent protocol to make it “greener”. The current protocol requires clients to be fully powered on to be participating members in the P2P overlay network (a “swarm” in BitTorrent). The authors proposed simple changes such as including long-lived knowledge of sleeping peers and a new wake-up semantic. This allows clients to sleep when not actively downloading or uploading, yet still be responsive swarm members.

3.3.6 Shifting to save: Traffic Engineering

Novel traffic engineering algorithms and techniques have recently been proposed with the aim of reducing energy consumption. Shifting traffic around allows specific equipment to be switched off and computation to be performed in greener locations.

Restrepo *et al.* [169] proposed a novel energy reduction approach that takes load-dependent energy consumption information of communication equipment into account when performing routing and traffic engineering decisions. A similar work was done in [202], where the authors assumed the availability in the near future of networking hardware in which an interface can operate at various sending rates. The main idea of the technique they proposed is to distribute traffic across alternative paths in a way that maximises energy savings. In [54] Cianfrani *et al.* proposed a novel network-level strategy based on a modification of current link-state routing protocols, such as OSPF. According to this strategy, IP routers are able to power off some network links during low traffic periods. The authors proposed a modified Dijkstra Shortest Path First algorithm that detects links to power off. Switching between the active and sleep modes consumes considerable energy and time, which motivated Andrews *et al.* [13] to consider the scheduling problem jointly with the routing problem. Routing determines the path each connection should follow, while scheduling decides the active periods for each network element. By combining the two problems the authors were capable of simultaneously minimising energy and end-to-end delays.

From an environmental point of view, the objective of green networking is to minimise greenhouse gases emissions. Enforcing the use of renewable energy is an important step in this direction. Dong *et al.* [68] proposed a novel approach with this aim. Similar to the work by Shen and Tucker [180], the authors of [68] developed efficient approaches, ranging from Mixed Integer Linear Programming (MILP) models to heuristics to minimise energy consumption of IP over WDM networks. But Dong *et al.* go a step further and also attempt to reduce CO_2 emissions by maximising the use of renewable energy sources in the network. The idea is to ship information to distributed renewable energy locations, processing and switching remotely instead of transporting the energy generated by renewable sources.

3.3.7 Lacking information? Measure it

Device specification data-sheets of current network equipment do not include comprehensive energy consumption values. They report merely maximum rated power. This value is insufficient to understand the actual energy consumption of the networking device under different configurations or traffic loads. Unfortunately, due to a lack of empirical studies, much of the research on green networking, including many of the papers referred in this section, is based on these figures. To try to alleviate this problem, Chabarek *et al.* [44] measured the power demands of two widely used Cisco routers and created a generic model for router power consumption. In addition, the authors proposed optimisation techniques to determine the optimal system configurations that minimise power consumption while preserving performance requirements. A related work was presented in [136]. Mahadevan *et al.* presented a power measurement study of a variety of networking gear, and also proposed a novel network energy proportionality index. More recently, Sivaraman *et al.* [184] proposed a fine-grained profile of energy consumption on the NetFPGA platform[1]. By using a high-precision hardware-based traffic generator and analyser, and a high-fidelity digital oscilloscope, the authors devised a series of experiments allowing them to quantify the per-packet processing energy and per-byte energy consumption of a NetFPGA card.

To be able to evaluate the performance of energy-aware networks it is important to have a common framework for measuring and reporting the energy consumption of a network. With this goal in mind, Bianzino *et al.* [32] compared and contrasted various energy-related metrics and defined a taxonomy of green networking metrics, which is probably an important first step to reach a consensus in the research community that is devoted to these matters.

For more detailed surveys on green networking I forward the reader to [31] and [220]. Bianzino *et al.*'s survey [31] is computer networking-oriented, whilst the other, by Zhang *et al.* [220], is optical networking-oriented.

[1] The NetFPGA is a low-cost reconfigurable hardware platform optimised for high-speed networking. It consists of a fully programmable Xilinx FPGA based core with four Gigabit Ethernet interfaces, and functions as an IP router. Like commercial routers, the entire datapath is implemented in hardware. The NetFPGA can thus support full Gigabit line rates and has low processing latency [146].

3.4 From electronics to optics

The integration of optics and electronics in IP networks has been a hot topic of research in the past decade. The technique I propose in Chapter 7 to reduce the energy footprint of IPTV networks is an example of such integration. In this section, I review some research done on this subject. Additionally, in Appendix A I present some topics that, despite their orthogonality to the proposal presented in that chapter, are nonetheless closely related. Namely, in the appendix I address optical multicast, traffic grooming, and aggregated multicast.

3.4.1 Optics vs electronics: there is room for both

Optical technologies inherently high bandwidths greatly exceed the bandwidth of any conceivable electronic device. The information-carrying capacity of optics is thus well beyond the capabilities of electronics. This imbalance between optics and electronics is sometimes referred to as the "electronic bottleneck". By realising this fact, in the past decade several research groups have analysed the problem of integrating optical technology inside routers to scale its capacity and reduce its power consumption. An example is the work by Keslassy *et al.* [123]. In their paper the authors identified an optical switch architecture with predictable throughput and scalable capacity — the Load-Balanced switch proposed by Chang *et al.* in [46] — and extended it in order to solve the problems that made the original switch unsuited for a high-capacity router.

To take full advantage of the capabilities optics can offer, the final goal is to build an all-optical router. At present, however, there does not appear to be a compelling case for replacing electronic routers with all-optical packet switches [193]. The key challenge in finding a technically feasible solution to optical packet switching is the lack of an adequate optical buffering technology. The most commonly used optical buffers are based on fibre delay lines, which are physically very large and inflexible. Electronic RAM is still the most attractive choice due to its small size and low power consumption. Another problem is the still immature optical signal processing technology. Only very simple signal processing, such as wavelength conversion or regeneration, is amenable to photonic implementation [99, 100]. It seems therefore clear that electronics will continue to be the technology of choice for high-performance signal processing and for buffering in the future.

But can optical technology help in reducing the energy footprint of networks? In terms of processing capabilities, integrated nonlinear optical circuits still consume significantly more energy than CMOS in all but the very simplest of circuits [196]. This is mainly because in CMOS most of the switching energy is consumed during bit transitions, while photonic devices rely on optical non-linearities that require an ongoing supply of power. So it does not seem a good option in this respect. On the other hand, in terms of routing and switching capabilities, techniques such as optical bypass can be an interesting energy-friendly option [99]. This is the technique I propose in Chapter 7 to reduce the energy consumption of IPTV networks. I briefly explain its rationale in the following.

In core routers, power consumption is dominated by forwarding and cooling. Address resolution and packet forwarding consume approximately 40% of router power [15]. As most of the traffic handled by a router is transit traffic [175], such electronic processing is wasteful. Optical bypass can eliminate this expensive high-speed electronic processing at intermediate nodes, and thus save energy. Without bypass, all lightpaths[1] incident to a node must be terminated, i.e., all the data carried by the lightpaths has to be electronically processed and forwarded by IP routers. With optical bypass, traffic not destined for a given node is placed onto a WDM wavelength that is not processed by that router. This can be accomplished by placing a WDM circuit-switched optical cross-connect (OXC) between the router and the incoming optical port so as to direct channels not destined to that router directly to the node output [175]. Figure 3.2 illustrates such node in a simplified manner. With optical bypass the traffic transiting a node can therefore remain in the optical domain, as opposed to undergoing costly Optical-Electrical-Optical (OEO) conversions and per-packet inspections. This can significantly save the number of IP router ports and consequently reduce energy consumption [180]. As an aside, OEO conversions are also undesirable as they offset the high-speed of the optical transport. There are, however, limitations on the use of optical bypass. The most important is its coarse granularity. OXCs switch at the wavelength-level. This inflexibility in switching granularity can cause waste of bandwidth. Internet traffic has many small and diverse flows which emphasises the importance of resource sharing. The lack of multiplexing gain of all-optical switching is therefore a disadvantage that should be considered.

3.4.2 Hybrid architectures: the best of both worlds

Optical cross connects relieve electronics from processing and switching. The problem, as explained before, is that due to its coarse granularity, bulk transport in optics can be bandwidth inefficient, especially for bursty traffic. With electronic switching the packets or flows can be processed at a much finer granularity. Smartly combining the strengths of optics and electronics seems therefore to be a good option. This type of *hybrid* architecture (also called multi-granular [218] or translucent [179], among other nomenclatures) that represents a compromise between all-electronic and all-optical switching has been the subject of interesting research in the past few years.

S. Aleksic [7], for instance, examined different switching and routing architectures based on both pure packet-switched and pure circuit-switched designs by assuming either all-electronic and all-optical implementations. The author concluded that to build energy efficient networks a kind of dynamic optical circuit switching should be used within the core network together with an efficient flow aggregation at edge nodes. Enablers for this type of networks are novel hybrid optical cross-connect architectures combining slow (millisecond regime) and fast (nanosecond regime) switching elements, as the one proposed recently by Zervas *et al.* [218]. This equipment is able to switch at the fibre, waveband, wavelength and sub-wavelength granularities. Several other examples of hybrid architectures exist in the literature [37, 82, 88, 101, 177].

An example of a routing scheme that makes use of these hybrid architectures is the work by Huang and Copeland [105]. The authors proposed a hybrid wavelength and sub-wavelength rout-

[1] In an all-optical network, a lightpath is an optical point-to-point connection from a source to a destination.

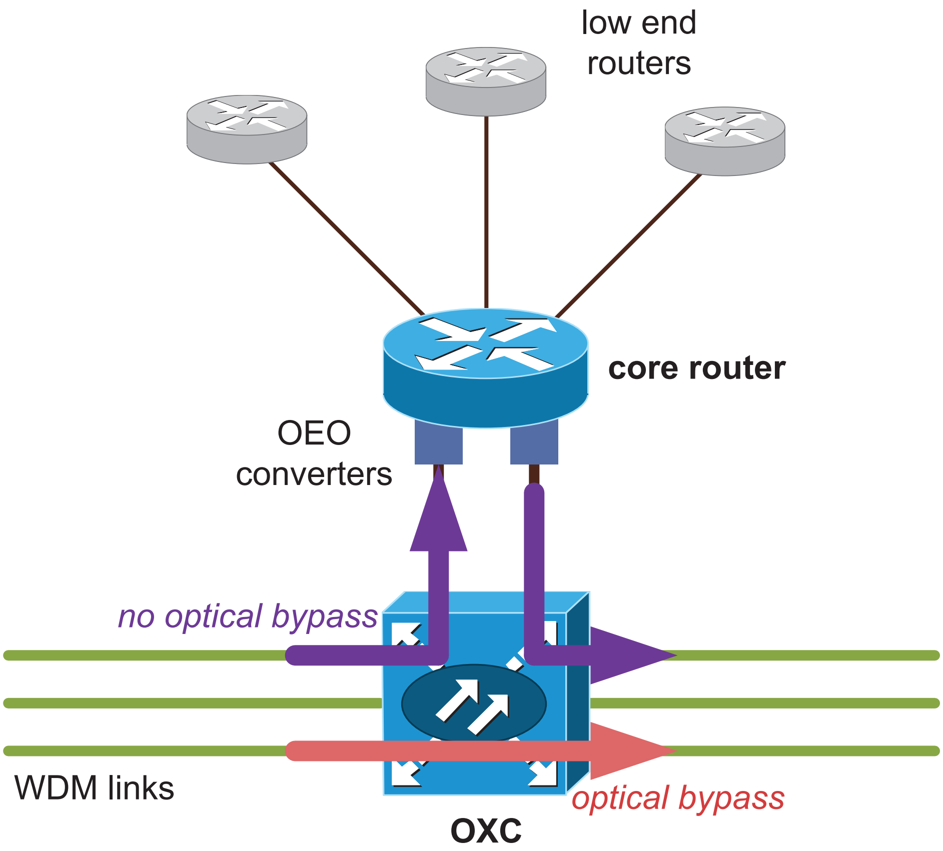


Figure 3.2: Optical bypass-enabled network node

ing scheme that can preserve the benefits of optical bypass for large traffic flows and still provide multiplexing gain for small traffic flows. The idea is to route traffic demands with large granularity using wavelength routing and those with small granularity using sub-wavelength routing. They therefore propose a "dedicated" set of wavelength channels to be optically switched and a "shared" one to be electronically routed. This scheme is similar to what I propose in Chapter 7 for IPTV systems.

The integration of optical circuit-switching techniques with electronic packet-switching requires a unified control plane. This is an essential component in the evolution of interoperable optical networks. Generalized Multiprotocol Label Switching (GMPLS) [140], the emerging paradigm for the design of control planes for OXCs, is a promising technology by providing the necessary bridges between the IP and optical layers [152]. GMPLS extends the Multiprotocol Label Switching (MPLS) [172] control plane to encompass several switching granularities, from packet and layer 2 switching to wavelength and fibre switching. The development of GMPLS required modifications to current signalling [22] and routing [127] protocols. Extensions to RSVP-TE [154] and OSPF-TE [126] in support of GMPLS were already standardised.

Chapter 4

Methodology and dataset

The ideas I propose and analyse in this dissertation are evaluated by means of trace-driven analysis. It is widely accepted [113] that a thorough evaluation using real workloads enables the assessment of future network architectures with an increased level of confidence. This chapter opens with an explanation of the motivation for the chosen methodology. Then, I describe the dataset used in this study, detailing how the data were collected, cleaned and treated. The chapter ends with an analysis that aims to validate the dataset. This validation consists in analysing specific characteristics of the data trace and contrasting the results obtained with those from a similar study made by other researchers using a different dataset.

4.1 Methodology

The research community working on IPTV systems has relied upon hypothetical user models which are sometimes different from reality and can lead to incorrect estimation of system performance. As I already mentioned in the previous chapter, constant-rate Poisson models are generally used as workload model for these systems. Examples include [189], [81], [150], [130], among others. Unfortunately, this model does not capture IPTV user behaviour well. Users switch channels more frequently than this simple model predicts. This fact was proved by Qiu *et al.* [160] recently. These researchers have characterised and modelled user activities in an IPTV network. They used real data from an operational nation-wide IPTV system[1]. Based on their analysis the authors developed a series of models for capturing the probability distributions and time-dynamics of user activities. They show that the simple mathematical models generally used in these studies are not capable of capturing the high burst of channel switches at around hours boundaries, and are thus not good models.

By analysing the dataset used in this dissertation[2], I also observe this fact. In Figure 4.1 I demonstrate, by means of an example, the problem of using a simple Poisson distribution as a mathematical model to represent the behaviour of IPTV users. The figure presents the Cumulative Distribution Function of the number of channel switches during one-minute periods

[1] AT&T's.

[2] Which is described in some detail in Section 4.2.

(a *zapping period*, according to [42]). The analysis was done on the whole dataset (containing all channel switching events from 255 thousand users over a six month period). In the figure I compare the empirical data with a Poisson distribution with parameter λ equal to 1.948.

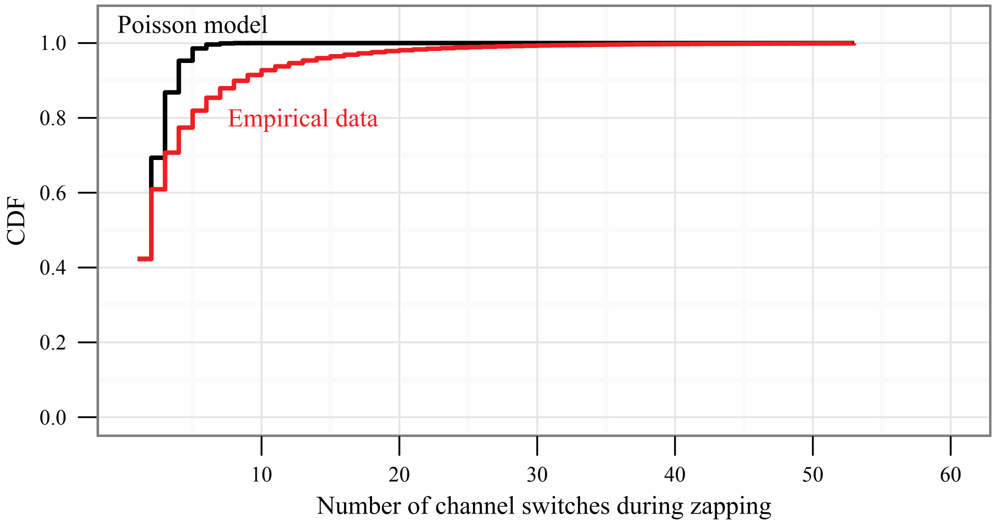


Figure 4.1: Number of channel switches in zapping mode

As can be seen, the Poisson model is conservative in terms of the number of channel switches a user performs during zapping periods. For example, the probability of a user making five channel switches or more in a one-minute period is negligible when using the Poisson distribution. But in fact by observing the empirical data one can conclude that there is a 20% probability of a user switching channels five times or more during a zapping period. This observation has important consequences for the current study. For instance, the fact that users enjoy zapping more than the Poisson model predicts is a stronger argument for the use of the scheme evaluated in Chapter 5.

The lack of an acceptable mathematical model for IPTV user behaviour[1] and the availability of an IPTV trace are the two reasons why I opted for trace-driven analysis as the methodology used to evaluate the schemes proposed in this dissertation. This IPTV trace from Telefonica is used as input to the analysis performed and presented in chapters 5, 6 and 7. In those chapters I explain the precise methodological details of each particular experience.

4.2 Dataset

I was fortunate to obtain a collection of IPTV channel switching logs from an IPTV service — Imagenio — offered by an operational backbone provider, Telefonica. Imagenio is a commercial, nationwide service, offering 150 TV channels over Telefonica's IP network. The access links use

[1] The realistic model proposed by Qiu *et al.* [160] was not available when the bulk of this study was being realised.

ADSL technology and the network is composed of 680 DSLAMs distributed along 11 regions. To give an idea of the scale of the dataset, the 700GB trace spans six months and records the IGMP messages on the channel changes of around 255 thousand users. The number of daily channel switchings clocks 13 million on average.

4.2.1 Data collection

I should start by clarifying that I was not responsible for trace collection. This process was executed by engineers at Telefonica. In this subsection I describe in some detail the data collection process. The information included here is the result of several discussions with researchers and engineers from Telefonica R&D[1].

As I explained in Chapter 2, whenever an IPTV user switches to a new TV channel, two IGMP messages are generated by the Set Top Box (STB) and sent towards the network: an IGMP leave request from the current TV channel, and an IGMP join request to the TV channel the user is switching to. In Telefonica, as in most IPTV networks, all channels are distributed continuously to all DSLAMs. This maintains the channels as close to the users as possible to help reduce channel change delay and avoids signalling messages in the IP network. The leave and join messages therefore arrive at the DSLAM, which then distributes the corresponding TV channel to the STB that requested the change. To collect the traces the DSLAMs were instrumented to send *all* IGMP join and leave requests sent by *all* STBs to a particular server in its region: a *local area server*.

Figure 4.2 pictorially illustrates the data collection process. Every time the DSLAM received an IGMP leave or join message from an STB it would forward this message to the local area server. One local area server served many DSLAMs, and some regions had more than one such server (for example, Madrid and Barcelona had four local area servers each). Each local area server then recorded every message received from the DSLAMs of its area into a log. The server was also running a script that periodically sent all log files to a central data collection server using a crontab[2]. There was only one central server keeping all logs.

Concerning the reliability of data collection, three sources of errors should be taken into account: errors in communication, problems/failures in the network elements, and possible lack of synchronisation. Concerning the first problem, all log messages were sent over UDP, so there was indeed the possibility of messages being lost. Second, log collection was regarded as a low priority process in Imagenio[3]. Therefore, in the event of a DSLAM processor overload, for instance, the logs would not be generated. As this is a private provider-managed network, however, these two problems are not severe. To guarantee the expected high quality of experience for its customers Telefonica rigorously controls the load of its network elements. Hence, the

[1] I would like to express my gratitude in particular to Pablo Rodriguez, Javier Benito and Enrique Urrea from Telefonica R&D, and to Meeyoung Cha from KAIST (intern at Telefonica R&D at the time of data collection) for kindly sharing all the details about the process.

[2] Cron is a time-based job scheduler in Unix-like computer operating systems. Cron enables users to schedule jobs to run periodically at certain times or dates. Cron is driven by a crontab file, a configuration file that specifies shell commands to run periodically on a given schedule.

[3] *"The main purpose of Imagenio is providing video service to customers, rather than collecting logs"*, I was told.

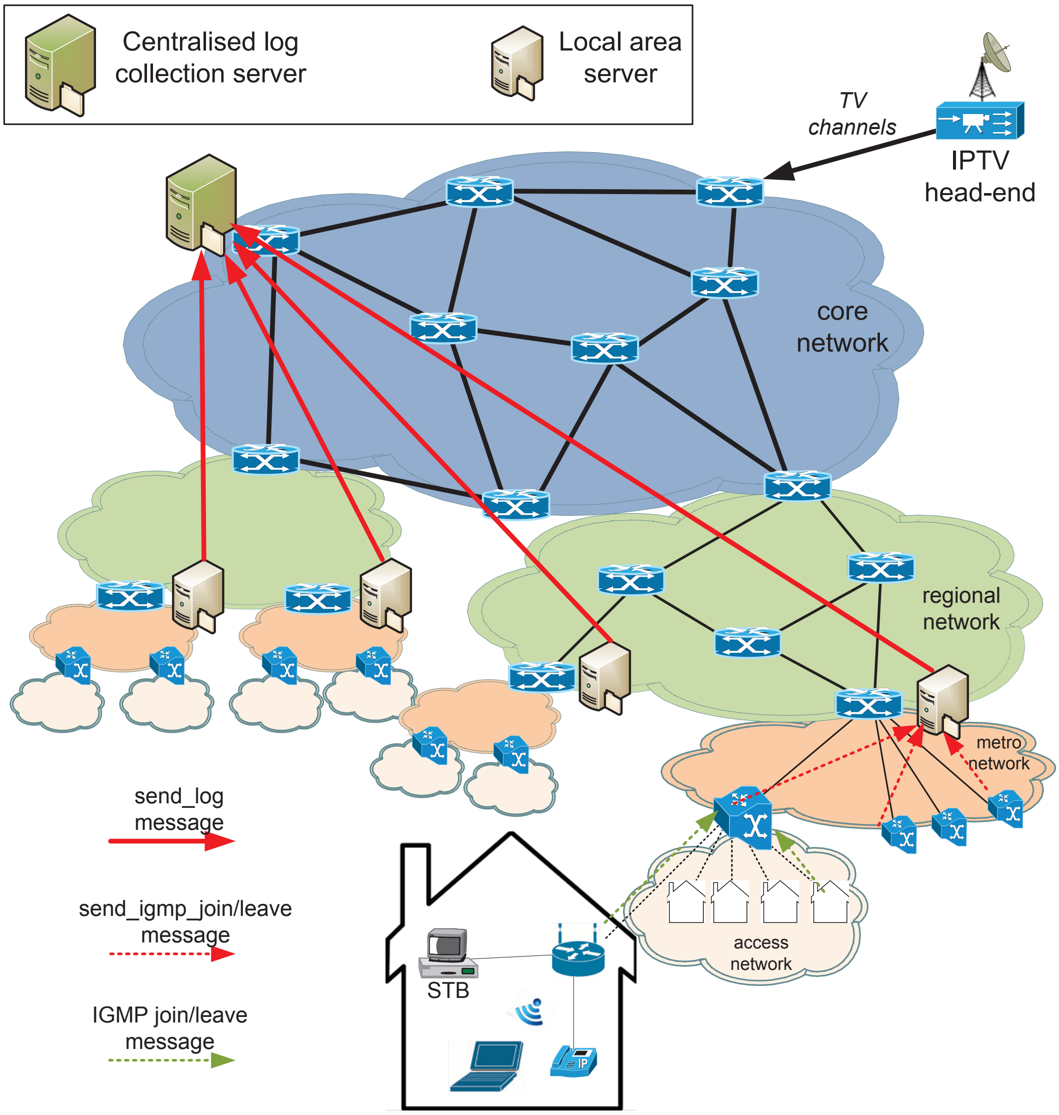


Figure 4.2: Data collection process

probability of these events occurring is low. Such low probabilities, put together with the scale of the dataset, gives guarantees that possible errors are rare and therefore have negligible impact on statistics. Finally, as this is a data trace, i.e., a *time-stamped* ordered record of all requests of the IPTV system, it is very important that the network elements are synchronised. In Telefonica's IPTV network all network elements are synchronised using the Network Time Protocol (NTP). The time reference is taken from a reference clock in Telefonica's network. NTP is known to achieve (worldwide) accuracy in the range of 1 to 50ms [190], which is one order of magnitude better than what is necessary for the current study.

4.2.2 Data characterisation

The trace includes all channel switching events from April 16th 2007 to October 20th 2007, six months in total. The log scales up to 150 TV channels, 680 DSLAMs, and 255 thousand users.

Table 4.1: Dataset statistics

Trace duration	6 months
Number of users	255 thousand
Number of DSLAMs	680
Average number of daily channel switching events	13 million
Size of the dataset	700 GB

Table 4.1 summarises these statistics. These data do not include any other information. For example, they do not capture performance related metrics such as network latency, jitter, and loss of the IPTV streams. They also do not capture the remote control commands issued by the user to switch channels.

A single line from the trace has the information on channel switching presented with the following format:

```
<MONTH> <DAY> <HOUR>:<MIN>:<SEC> [<A>.<B>.<C>.<D>.<E>.<F>] <SHELF1>
/<SLOT1>: Multicast client <IP_MUL> link UP from <IP_SRC>,
port <SHELF>/<SLOT>/<PORT>, vpi/vci <VPI>/<VCI>\$
```

For the purposes of this study, the most relevant information present in each line is the following:

1. `<MONTH> <DAY> <HOUR>:<MIN>:<SEC>`

   Timestamp in units of seconds.

2. `<A>.<B>.<C>.<D>`

   IP address of the DSLAM.

3. `<IP_MUL>`

   IP address of the multicast group (in most cases, a TV channel).

4. `<IP_SRC>`

   IP address of the source (in most cases, a Set Top Box).

5. `link UP`

   Multicast option of joining (UP) or leaving (DOWN) a channel.

An example line of the log follows.

```
Apr 16 00:05:03 [172.24.215.1.3.254] 1/8:  Multicast client
239.0.0.14 link UP from IP 10.90.1.74, port 1/2/18, vpi/vci 8/35
```

4.2.3 Data cleaning and parsing

The dataset made available by Telefonica has some information that is irrelevant for the purposes of this dissertation, which means it can be filtered without loss. In addition, the original format of the IPTV traces is not the most convenient for the study I present in Chapters 5, 6, and 7. For this reason I decided to clean and parse the data, a process I describe in this section.

The first step of the process is to filter all irrelevant information. This includes removing messages from all devices other than the Set Top Boxes, all messages from non-TV channels, and also some outliers. In more detail:

1. All messages with a source IP address outside the `10.x.x.x` range are removed. `10.x.x.x` is the STB IP address format. However, there is a very small percentage (less than 0.1% of the total) of requests with a different source IP address. An example is the address `0.0.0.0`, the default IP address of every STB. The first time an STB is plugged on the network, it starts sending packets with this address until the automatic installer assigns it the correct IP address.

2. All messages with a multicast IP address outside the `239.0.x.x` range are removed. These are non-TV multicast groups. There is also a very small percentage of requests in this category (again, less than 0.1% of the total). These include multicast groups used to manage Set Top Boxes, for example for bootstrapping and for upgrading the software.

3. Some outliers are also removed. When I analysed the data I detected a strange behaviour from some STBs. In particular, some STBs sent IGMP signals with a fixed periodicity for the trace duration (the whole 6 months). Such non-human behaviour is usually a characteristic of devices included in the network for testing purposes. I assume this is the case, and filter the information from these STBs. Note that I found only 4 such devices (out of the 255 thousand), hence its removal or inclusion would always be statistically irrelevant.

The second step of the process is to parse the data for a more convenient format. As explained in the previous subsection, each line of the trace includes information — namely, the STB IP address — which allows the separation of the channel switching events from each STB. I therefore create a single file per STB that includes all channel switch requests made by the users of a specific household. After this final step, the cleaned and parsed channel switching log used in this study has the following format:

```
#first line only:
STB IP = <STB_IP_ADDRESS> DSLAM IP = <DSLAM_IP>
#from the second to the final line:
<MONTH> <DAY> <HOUR>:<MINUTES>:<SECONDS>|<TYPE>|<CHANNEL_NR>
#...
```

Each log file includes, in the first line, the STB IP address and the DSLAM IP address. The rest are all channel switching requests — one per line — made by that STB during the whole period of the trace. Each of these lines includes the timestamp (date and time with the precision of one second), the request type (`UP` or `DOWN`), and the channel number. An example of part of one such parsed file follows.

```
STB IP = 10.74.59.98 DSLAM IP = 172.24.240.1
Jul 1 00:41:36|UP|23
Jul 1 00:41:44|DOWN|23
Jul 1 00:41:44|UP|25
Jul 1 00:41:48|DOWN|25
Jul 1 00:41:48|UP|182
Jul 1 00:41:51|DOWN|182
Jul 1 00:41:51|UP|30
Jul 1 00:41:51|DOWN|30
Jul 1 00:46:32|UP|31
```

4.2.4 Validation of the dataset

Trace-driven analysis have several advantages when compared with other evaluation methods. These include, but are not limited to [181], the similarity with the actual implementation of the system under evaluation and the fact of a trace being an accurate workload offering high level of detail. But using a data-trace from a real system has nonetheless some disadvantages. One issue that has to be taken into account is that of representativeness. Traces taken from one system may not be representative of the workload on another system. For this reason, validating the dataset is an important part of the process, to help ensure that the algorithms and ideas one wants to evaluate are done so on correct, useful, and representative data.

To validate a data trace, a possibility is to obtain a different trace under a different environment and use it to validate the original dataset [113]. An alternative is to compare the results obtained from the analysis of the same type of system using a different dataset, with the same analysis using our own. This is the technique I use in this section in order to validate the dataset used in the current study. As explained in Chapter 3, Qiu et al [160, 161] have analysed an IPTV system using a different dataset from the one used in the current study[1]. These researchers measured several characteristics of their IPTV system, explored TV channel popularity and characterised and modelled user activities. By comparing a relevant subset of their results with those obtained by analysing Telefonica's dataset, I believe it is possible to validate the correctness, usefulness and representativeness of these traces.

The first characteristic of an IPTV system I analyse is the number of online users during the course of a representative week. In [160] and [161] Qiu *et al.* found very strong diurnal patterns, with daily peaks at around 9PM, followed by a quick decrease in the number of online Set Top Boxes, reaching a daily minimum at 4AM, and then steadily rumping up during the

[1] The authors analysed a dataset from AT&T.

course of the day. Unfortunately, Telefonica's trace does not include information on when an STB is turned off. For this reason I differentiate an online from an offline user by assuming a user is offline when the *channel dwell time*, i.e., the time an STB stays tuned in a channel, is above one hour. This is the same procedure followed by Cha *et al.* [42] for the same purpose. I am thus assuming that users leave the STB on even when they switch the TV off. Of course, some users may watch TV for more than one hour without switching channels, especially when watching movie channels. In Figure 4.3 I show the time series of the number of online users over a representative trace period. The results are very similar to Qiu's. The strong diurnal pattern is present, with a daily peak in the evening (at around 10PM in Spain). It is also interesting to note that the lowest evening peaks correspond to Friday and Saturdays.

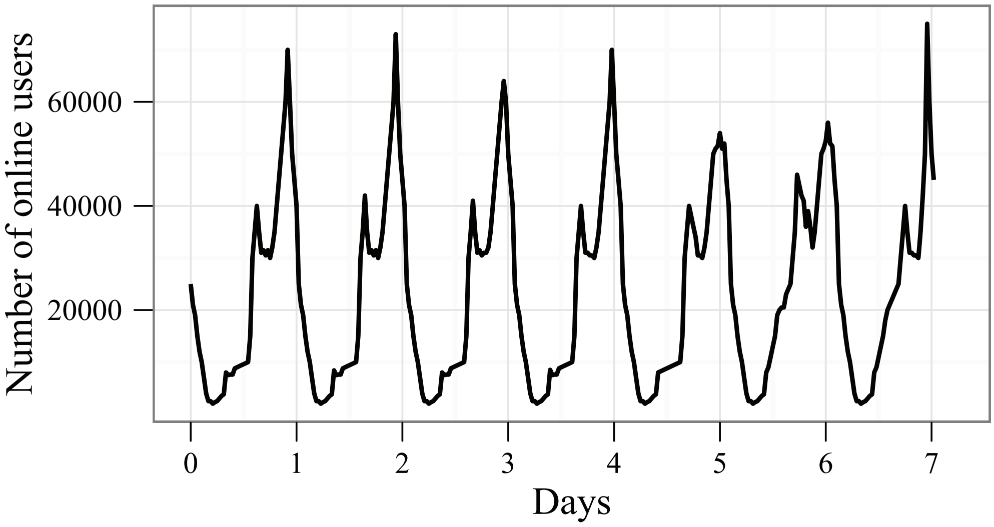


Figure 4.3: Number of viewers during a representative week

In [161] Qiu *et al.* examined the long term distribution of channel popularity using both *channel access frequency* — the number of channel switching requests to the channel — and channel dwell time (defined above). They observed that both distributions are very similar (I return to this similarity later), exhibiting high skewness, with the top 10% of channels accounting more than 90% of channel accesses. By analysing the channel switching events of a subset of 2200 random users (over the entire 6 month period), I also observed the same trend in my dataset, as Figure 4.4 attests (data points labelled "empirical data"). This figure shows the channel access frequency as a function of channel ranking. A channel rank of 1 indicates the most popular channel and the unpopular channels are at the tail of the distribution.

The high skewness of popularity is usually modelled using Zipf-like distributions. Qiu *et al.* have indeed shown that the 10% most popular channels can be modelled with this type of distribution. However, the exponential function achieves a better fit for the large "body" part of the distribution function. They thus proposed a hybrid model with the probability density

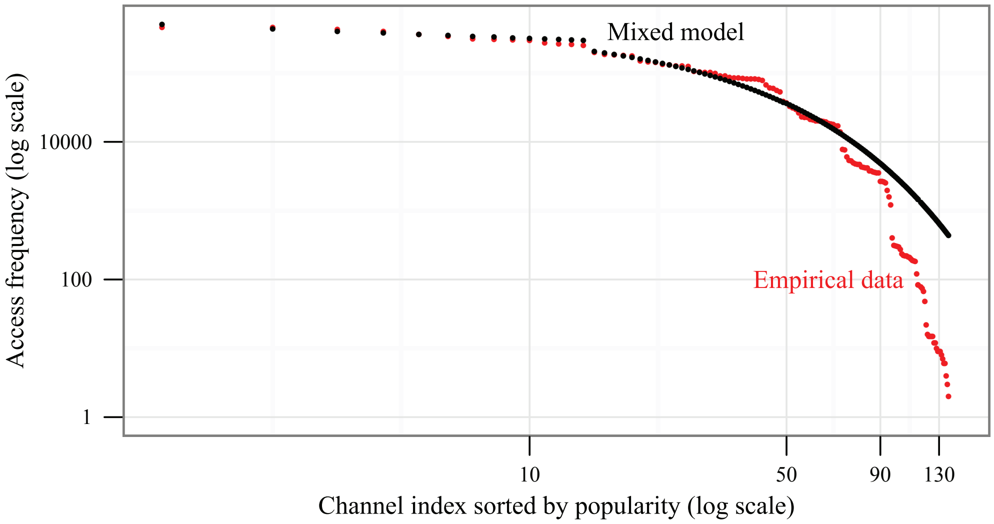


Figure 4.4: Channel popularity distribution

function expressed as follows,

$$f_o\,(i) = \begin{cases} C_1 i^{-\alpha}/C_0 & i < 10\% \text{ of available channels,} \\ e^{-\beta+C_2}/C_0 & \text{others,} \end{cases} \tag{4.1}$$

The parameters Qiu *et al.* found for the Zipf-like distribution, $f_1(i) = C_1 i^{-\alpha}$, and for the exponential distribution, $e^{-\beta+C_2}$, are presented in Table 4.2. Note that C_0 is a normalisation factor such that $f_0(\cdot)$ is a well-defined probability density function. This popularity model also fits quite well with the Telefonica dataset. Again, I analysed the channel switching events of a subset of 2200 random users (over the entire 6 month period). I fixed the values for C_1 and C_2, and the values for α and β that fit the empirical data are also presented in Table 4.2. The result can be observed in Figure 4.4 (data points labelled “Mixed model”). The lower value of α in the Zipf-like part of the distribution means that the popularity of the top 10% channels offered by Telefonica decays slower that in AT&T’s case. This may be justified by the size of the population. AT&T’s dataset has four times more users than Telefonica’s and hence the difference in popularity between top channels may be more pronounced. This is just a conjecture, however. The justification for the higher β parameter from the exponential distribution may be easier to accept. The higher β value for Telefonica’s dataset means the popularity of the bottom 90% channels decays more rapidly than in AT&T’s. The reason may be the total number of channels. In AT&T’s network, 630 channels compose the bottom 90% list of channels, which is five times Telefonica’s figure. This may justify the higher skewness of Telefonica’s graph, as users have less channel options.

I mentioned before that channel popularity based on dwell time and channel popularity based on access frequency produce similar results, as reported by Qiu *et al.* In their paper [161] the authors indeed found a very strong correlation between these two popularity measures. I

Table 4.2: Parameters of the channel popularity models

	AT&T trace	Telefonica trace
α	0.513	0.2
C_1	12.642	12.642
β	0.006	0.05
C_2	2.392	2.392

found the same correlation in Telefonica's dataset. This can be observed in Figure 4.5. This figure shows the scatter plot of the ranks of the channels according to each popularity measure. The x-axis shows the popularity rank according to channel access frequency, while the rank according to channel dwell time is shown on the y-axis. The points are spread well along the diagonal line, indicating strong correlation. Their Spearman rank correlation coefficient and their Pearson correlation coefficient are both equal to 0.97, demonstrating the strong correlation. Very similarly, Qiu *et al.* reported the values 0.98 and 0.97 for these coefficients, respectively.

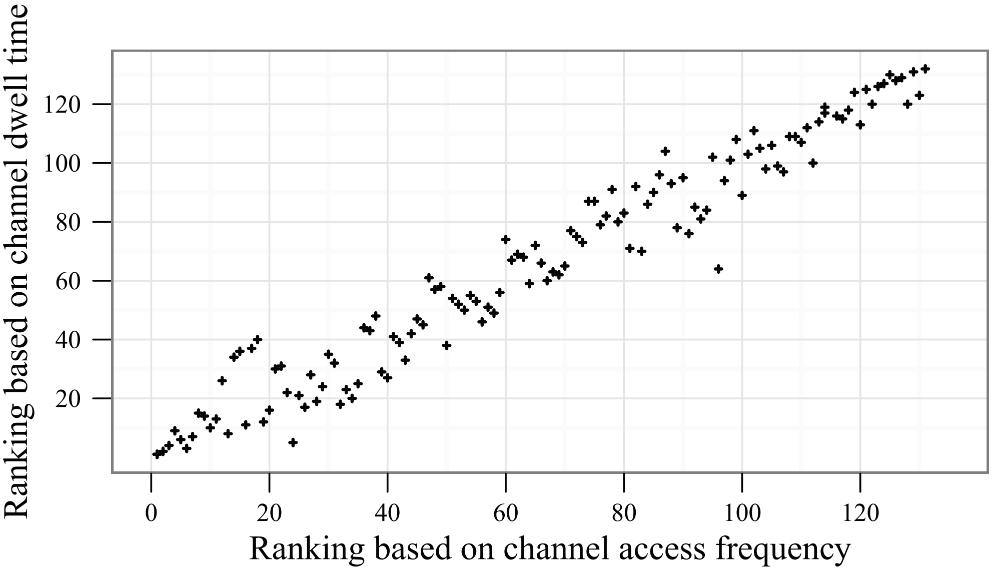


Figure 4.5: Correlation between channel access frequency and channel dwell time

An insight that is important to gain for the current work (in particular to Chapter 5) from the data is to understand how IPTV users switch channels. Do users switch *linearly*, up or down to the next or previous TV channel, or do they perform more targeted switching, with the user switching intentionally to a specific channel of choice (thus "jumping" several channels)? By analysing the whole dataset from Telefonica I observed that 55% of all channel switching was linear. Qiu *et al.* [160] reported 56% in the AT&T dataset. From these, in Telefonica's network 69% are up-channel-switches. This figure is equal to 72% in AT&T's case.

There is no such thing as a validated dataset [113], but I believe the similarity of the results obtained from this analysis of Telefonica's dataset with that from the AT&T studies helps increase the degree of confidence in the dataset used and in the results I present in this dissertation.

Chapter 5

Reducing channel change delay

One of the major concerns of IPTV network deployment is channel change delay (also known as zapping delay). As explained in Chapter 2 (Section 2.1.1), synchronisation and buffering of media streams can cause channel change delays of several seconds. The main concern in the industry and in the research community has been, in fact, to try to improve the performance on these two aspects, and several solutions have been proposed. One such solution is **predictive pre-joining of TV channels**. In this scheme each Set Top Box (STB) simultaneously joins additional multicast groups (TV channels) along with the one that is requested by the user. If the user switches to any of these channels next, the switching latency is virtually eliminated, and user experience is improved. The negative impact of this solution is additional load in the access network, and the evaluation presented in this chapter looks at the tradeoff between pre-join advantage in reduced switching latency versus the access network bandwidth cost.

As observed from the analysis of the data traces presented in Chapter 4 (Section 4.2), most channel switching events are relatively predictable: users very frequently switch linearly, up or down to the next TV channel. This favours this specific type of solution to the channel change delay problem. Previous work on this subject [81, 189] used simple mathematical models to perform analytical studies or to generate synthetic data traces to evaluate these pre-joining methods. I showed in Chapter 4 (Section 4.1) that these models are conservative in terms of the number of channel switches a user performs during zapping periods. They therefore do not demonstrate the true potential of predictive pre-joining solutions. This is an important motivation to perform an empirical analysis using the IPTV dataset available. Such realistic trace-driven analysis is the main differentiating point of my contribution. This is, to the best of my knowledge, the first empirical study of channel change delay reduction techniques.

The first pre-joining scheme I analyse in this chapter is very simple. In this scheme the neighbouring channels (i.e., the channels adjacent to the requested one) are pre-joined by the Set Top Box alongside the requested channel, during zapping periods. Notwithstanding the simplicity of the scheme, the trace-driven analysis shows that the zapping delay can be virtually eliminated for a significant percentage of channel switching requests. For example, when sending the previous and the next channel concurrently with the requested one, for only one minute after a zapping event, switching delay is eliminated for near half of all channel switching requests.

Importantly, this result is achieved with a negligible increase of bandwidth utilisation in the access link. Two other schemes are evaluated. The first considers pre-joining popular TV channels, but the results are unsatisfactory. The second is a personalised scheme where user behaviour is tracked to decide which channels to pre-join next. The improvement of this scheme over the simpler version is also insignificant.

5.1 Introduction

The offer of TV services over IP networks is very attractive as it represents a new source of revenues for network operators. IPTV offers network providers greater flexibility, while at the same time offering users a whole new range of applications. In order to compete in this market, IPTV operators have to at least guarantee the same Quality of Experience (QoE) offered by cable networks or over the air broadcasts. In this respect, one of the major concerns of IPTV network deployment is channel change delay (also known as zapping delay). This is the delay between the time the user switches to a particular TV channel and the time when the content is displayed on the TV screen. An analysis to the causes of this delay was presented in Chapter 2 (Section 2.1.1). I refer the reader to Figure 2.2 in particular. When a user switches to a new TV channel using his or her remote control, the STB issues a new channel request towards the network. After a certain time (the *network delay*), the first packets of that particular multicast group start flowing. Before play-out the STB still has to *synchronise* with the video stream (it has to wait for the next I-frame) and *buffer* some packets (to avoid starvation and to compensate for networked-introduced jitter and packet-reordering delay). The whole process therefore takes some time. Synchronisation and buffering of media streams can cause channel change delays of several seconds [80, 176, 182, 189]. Figure 2.3 in Chapter 2 pictorially summarises the contribution of each component of channel change time. It is known that this figure should be below 430 ms to guarantee an acceptable user experience [128], so this is a major concern for IPTV service providers that want to compete in this market.

By analysing the dataset described in Chapter 4, I observe that most channel switching events are linear: users switch up or down to the next TV channel very frequently[1]. Also, even when zapping is not linear, the "jump distance"[2] is usually small (i.e., there is a high probability for the user to switch to one of the neighbouring channels). These facts can be observed in Figure 5.1. This figure presents the Cumulative Distribution Function of the "jump distance" considering the analysis of the whole dataset (255 thousand users, 6 months, 13 million channel switches per day on average). The probability of zapping linearly ("jump distance" equal to 1) is close to 55%, and the probability of jumping to a close neighbour is also very high. For example, 80% of all channel change requests are to channels not more than six channels apart ("jump distance" equal to 6).

This kind of user behaviour is very favourable for a specific type of solution to the channel

[1] It is relevant to mention that at the time of data collection the deployed IPTV system supported an Electronic Program Guide.

[2] Assuming that the TV channels are numbered as a sorted list, as is common, the "jump distance" is the difference between the number of the channel switched to and the number of the channel switched from.

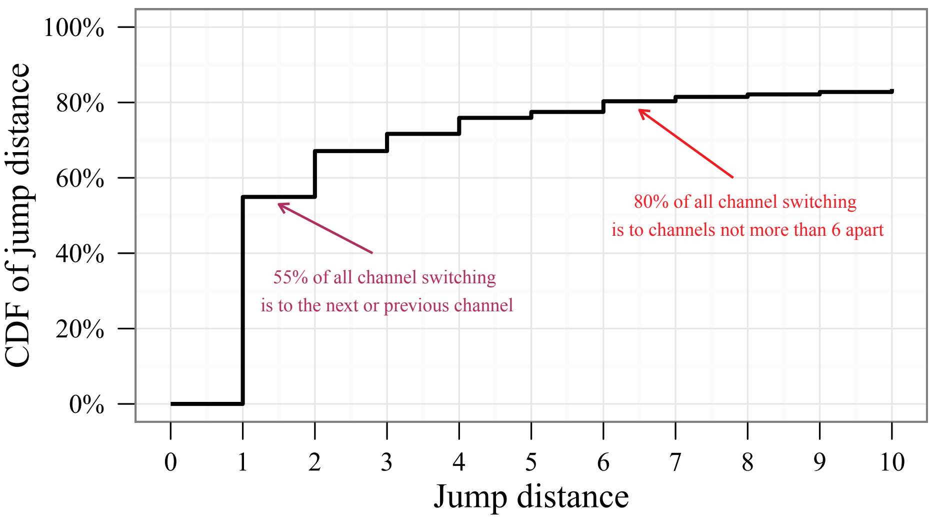


Figure 5.1: Cumulative distribution of zapping jump distance

change delay problem, namely, predictive pre-joining of TV channels. As explained in Chapter 3 (Section 3.2.4), in these schemes each Set Top Box (STB) simultaneously joins additional multicast groups along with the one that is requested by the user, thus anticipating future user behaviour. These schemes are thus based on the prediction of the next TV channel the user will switch to. If the prediction is right, the user will experience a small zapping delay because the channel is already synchronised in the STB. The negative impact of this solution is additional load in the access network, so there is a tradeoff between the advantage in reduced switching latency versus the access network bandwidth cost. Previous work evaluated this technique using analytical techniques or simulations based on simple mathematical models. As was proved by Qiu *et al.* [160] and as I demonstrated in Chapter 4 (Section 4.1), these simple models do not capture IPTV user behaviour well. For this reason, in this chapter, I perform a trace-driven analysis using the Telefonica dataset to evaluate this solution to the channel change delay problem. To the best of my knowledge, this is the first empirical study where channel change delay reduction techniques are evaluated using real IPTV usage data from an operational network provider. That is the main contribution of this work.

I consider several pre-joining schemes. In the first, the set of channels pre-joined are the neighbouring channels (i.e., channels adjacent to the requested one). These neighbouring channels are synchronised and buffered together with the requested one. Therefore, if the user decides to switch to any of these channels, the switching delay experienced is virtually zero. These additional channels are not sent to the STB continuously; they are kept during zapping periods only, to assure the scheme is bandwidth efficient[1]. One of the main advantages of this scheme is its simplicity. Notwithstanding its lack of sophistication, the trace-driven analysis shows that the zapping delay can be virtually eliminated for a significant percentage of channel switching requests. For example, when sending only two channels concurrently (the previous and the next,

[1] Bandwidth inefficiency was one of the problems of the original paper proposing predictive pre-joining of TV channels [51], as explained in Chapter 3 (Section 3.2.4).

respectively, thus assuming that the user will zap linearly), for only one minute after a zapping event, switching delay is eliminated for around 45% of all channel switching requests. This figure jumps to 60% if one considers zapping periods only (periods when the user is surfing/browsing, i.e., actively switching between channels). If the access network has enough bandwidth available to increase the number of neighbouring channels to eight, around 80% of all switching requests during zapping periods will experience no delay.

Globally, this scheme offers very interesting results. I demonstrate in this chapter that this simple scheme has a performance close to that of an optimal predictor. However, I also observe that user behaviour can vary significantly: it is true that many users enjoy zapping up and down, but others seem to zap less linearly. Therefore, while some users would benefit hugely from using this scheme, others would see only a relatively small improvement. With this limitation in mind I also consider other schemes to see if user experience can be improved for a wider audience. I first test a scheme where the most popular channels are pre-joined, either alone or together with some neighbours. This scheme proves inefficient when compared with pre-joining neighbours only. I also evaluate a personalised scheme. In this scheme user behaviour is tracked by maintaining information on user actions: does the user have favourite channels, preferring to switch to a particular channel or set of channels, or does he/she prefer to zap linearly? This scheme also accommodates temporal dynamics, capturing changes in user behaviour over time. In the end, the added complexity of the scheme does not result in an improvement over the simple scheme of pre-joining neighbours only.

As explained in more detail in Chapter 2 (Section 2.3), current operational IPTV networks are "walled gardens", with all TV channels distributed to the edge of the network (to the DSLAMs). However, due to access link bandwidth limitations, only one or two TV channels are distributed from this edge point to the Set Top Box. It is therefore important to underline that I assume in this study the access network is able to accommodate the peak bandwidth needed to distribute several TV channels concurrently. Most systems today distribute Standard Definition (SD) TV channels using MPEG-2, requiring 4 Mbps guaranteed bit rate per channel, thus sending channels in parallel increases the bandwidth requirements proportionally. I believe, however, that this is not a serious limiting factor of the type of schemes I analyse. In fact, in most OECD countries access networks already offer tens of Mbps of average download speeds. Japan and South Korea, for instance, have an average broadband speed close to 100 Mbps, with Japan already offering 1Gbps to some users, and it is expected other countries to follow this trend in the near future [157]. More importantly, by using these schemes the increase of bandwidth utilisation in the access link is negligible, since the concurrent channels are distributed to the STB during zapping periods only. I also assume that the STB is able to process (i.e., synchronise and buffer) several TV channels in parallel. In this study I take both these limitations in consideration, and restrict the number of neighbouring TV channels sent in parallel.

The rest of the chapter is organised as follows. The first scheme evaluated — pre-joining neighbouring channels — is described in Section 5.2. In section 5.3 I detail the methodology used to evaluate this (and the other) scheme(s), and in section 5.4 I present the results for this simple technique. The two sections that follow present and evaluate the other schemes.

First I consider pre-joining the most popular channels. Then I propose and present results of a personalised scheme where user behaviour is tracked. In Section 5.7 I discuss the advantages and disadvantages of the schemes under analysis, and I conclude this chapter in Section 5.8.

5.2 Simple scheme: pre-joining neighbouring channels

In current IPTV systems, when a user requests a TV channel using the remote control, this single channel is requested from the network and delivered to the Set Top Box. The idea behind all the schemes evaluated in this chapter is to pre-join, together with the channel requested, an additional set of TV channels concurrently, based on some sort of prediction. These will be synchronised and buffered simultaneously with the requested one. Therefore, if the following request is for a channel already present in the STB, there is no network, synchronisation, or buffering delay, and switching delay will be the result of STB processing only, thus virtually zero (i.e., way below the 430 ms needed to guarantee an acceptable viewing experience). The neighbouring channels stay in the STB for a limited, predefined period. I call this period the **concurrent channel time**. After this time, the STB sends IGMP leave requests and the neighbouring channels are removed.

The first scheme considered is very simple: the extra channels to pre-join are the neighbouring channels (i.e., channels adjacent to the requested one), and they are distributed to the STB during zapping periods only. An example of the use of this method is shown in Figure 5.2. In the figure I assume that after a channel switching event only two neighbouring channels (previous and next) are pre-joined additionally to the requested one. I also assume the user is in viewing mode in the beginning, i.e., he or she is settled watching channel x. The user then switches to channel y. Right after the channel change the STB enters in "zapping mode" and requests three TV channels from the network: the channel the user switched to, y, the next channel, $y + 1$, and the previous channel, $y - 1$. As there is no video data for channel y in the STB before the change, the viewer has to wait for the synchronisation and buffering of the video streams, thus experiencing zapping delay (represented by the gray box). When the user switches again, this time to channel $y+1$, he or she experiences virtually no delay, since the channel is already being received by the STB. As the user is in up-channel-switching mode, the STB sends an IGMP leave message from channel $y - 1$, and a join message for channel $y + 2$. When the user exits zapping mode[1], i.e., when it settles in channel $y + 1$, the STB leaves the neighbouring channels, y and $y + 2$. In the following channel switching event the user will therefore experience zapping delay, independently of the channel switched to.

5.3 Methodology

The schemes proposed in this chapter are evaluated by means of a trace-driven analysis for the reasons explained in Chapter 4 (Section 4.1). The IPTV trace detailed in that chapter is used as input to the analysis performed.

[1] After the channel concurrent time elapsing.

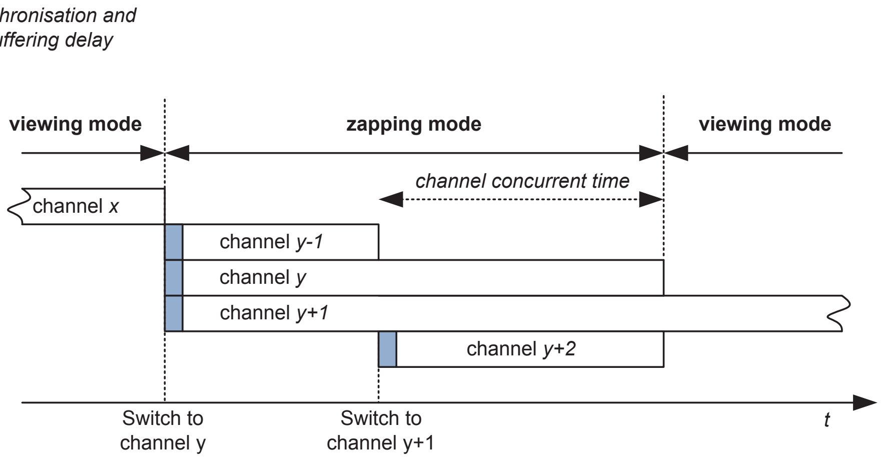


Figure 5.2: Predictive pre-joining of TV channels

A small detail needs clarification beforehand. Sometimes users zap linearly, from one channel to the next, swiftly (in less than the *normal IPTV switching delay*, which I consider to be 2 seconds in the rest of this chapter[1]). In these cases, the STB may not have time to synchronise to the requested channel, or to any of its neighbours. However, all these channels are already in synchronisation mode. Therefore, if the next change is to a channel already in the STB, the switching delay, albeit not being zero, will be less than the normal delay. In this case, I say the user experienced *partial delay* (some value between zero and the *normal delay*).

To evaluate the pre-joining schemes I developed a Python script that checks each line of the input trace to obtain each switching event. The current switching event is then compared with the previous, and one of these actions is performed:

1. If the time between two user switching events is above the concurrent channel time, no additional channel is in the STB, and therefore the user experiences the *normal switching delay*. The counter `normal_delay` is incremented.

2. If the time between two user switching events is below the concurrent channel time, there are additional channels in the STB. So, one of these three situations occurs:

   a If the user switches to a different channel from the ones in the STB, it will experience the *normal delay*. The counter `normal_delay` is incremented.

   b If the user switches to one of the neighbours in the STB, and if the time between two user switching events is above or equal to 2 seconds, the user experiences virtually no delay. This is due to the fact that the channel is in the STB, and is already synchronised. The counter `no_delay` is incremented.

   c If the user switches to one of the neighbours requested by the STB, but the time between two user switching events is under 2 seconds, the user experiences *partial delay*. As explained above, this is due to the fact that the channel was already requested by

[1] The reason why I consider 2 seconds is explained in Chapter 2 (Section 2.1.1). This is also the value usually considered in other studies on predictive pre-joining, such as [189], for instance.

the STB, but the user zapped rapidly, so it did not have time to synchronise. The counter `partial_delay` is incremented.

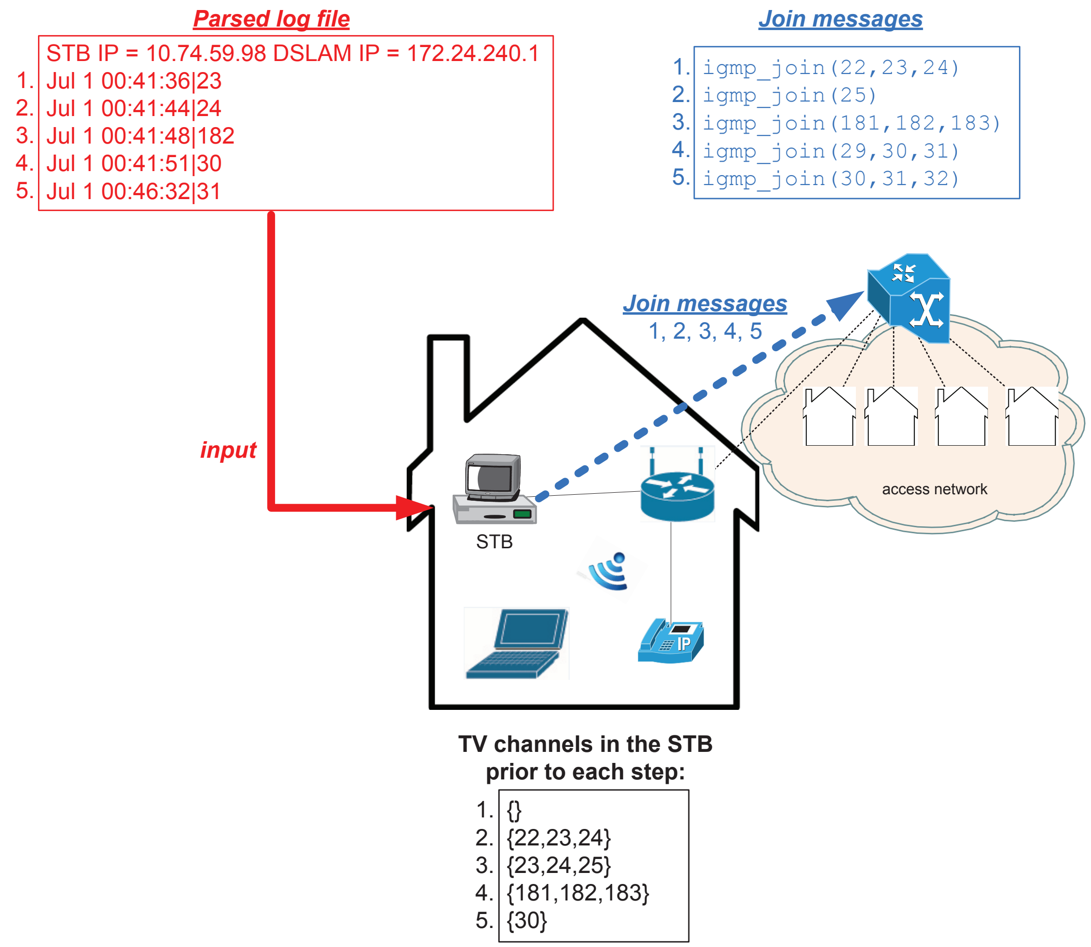
Figure 5.3 illustrates the proposed methodology with a simple example[1]. When the user turns the STB on it switches to channel 23 at 12:41:36am. This is translated into three IGMP join messages, to channels 22, 23, and 24, respectively, sent to the network. After a couple of seconds (the normal switching delay considered) these three TV channels are being distributed to the STB, so at the time of the next switching event, at 12:41:44am, they are all being received simultaneously. The user then switches to channel number 24. The channel is being received by the STB, already synchronised, so the user will experience virtually no channel change delay (the counter `no_delay` is incremented). For that reason, only a join message is sent to channel 25[2] (channels 23 and 24 are already being received). Next, the user switches to channel 182. As this channel is not available in the STB, the STB has to send a join message to channels 181 to 183, and the user will experience the normal zapping delay (the counter `normal_delay` is incremented). In any case, after the concurrent channel time two IGMP leave messages are sent by the STB to leave the additional channels. For this reason, and assuming the concurrent channel time is equal to one minute, although at 12:46:32am the user performs up-channel-switching to channel 31, this channel is not being distributed to the STB anymore, and so the user will experience the normal channel switching delay. Again, the counter `normal_delay` is incremented.

5.4 Evaluation

I consider two dimensions in the analysis. The first is the time the neighbouring channels are sent concurrently to the STB, the **concurrent channel time**, already referred to above. After this time, the STB sends IGMP leave requests and the neighbouring channels stop being distributed. The scheme leading to the best results would be to send these channels *always*, i.e., never leaving the neighbouring channels. However, this is inefficient, as it unnecessarily increases access link bandwidth utilisation. Therefore, in the schemes under analysis I maintain the neighbouring channels in the STB during a small period, between 10 seconds and 2 minutes. But I also present the results for the case “always” referred to before, for comparison. I choose this range of values in accordance with the definition of zapping (or surfing) periods in previous research [42], which is also in line with the way Nielsen Media Research demarcates viewing events. The second dimension is the **number of neighbouring channels** to send concurrently. Current Set Top Boxes typically receive one or two TV channels in parallel. Although the technology for a STB to stream more channels in parallel is available today, these are typically low cost devices, thus constrained in terms of its processing and memory capabilities. As the costs of processors, memory and storage continue to fall, devices capable of processing more channels in parallel will plausibly become cost-effective. Anyway, considering the limitations of STB processing and of

[1] This figure is based on the reference architecture presented in Figure 2.5.

[2] I omit IGMP leave messages.



Parsed log file

```
STB IP = 10.74.59.98 DSLAM IP = 172.24.240.1
1. Jul 1 00:41:36|23
2. Jul 1 00:41:44|24
3. Jul 1 00:41:48|182
4. Jul 1 00:41:51|30
5. Jul 1 00:46:32|31
```

Join messages

```
1. igmp_join(22,23,24)
2. igmp_join(25)
3. igmp_join(181,182,183)
4. igmp_join(29,30,31)
5. igmp_join(30,31,32)
```

TV channels in the STB prior to each step:

1. {}
2. {22,23,24}
3. {23,24,25}
4. {181,182,183}
5. {30}

Figure 5.3: Proposed methodology

access link bandwidth, I decide to restrict the number of concurrent channels to a minimum of 2 and a maximum of 8. In fact, the gain of sending more channels would be small, as can be inferred from Figure 5.1. All the results presented in this section and in the rest of the chapter arise from the analysis of the whole data set (255 thousand users, 6 months, 13 million channel switches per day on average).

Figure 5.4 illustrates the percentage of switching events that experience virtually zero delay. The x-axis is the percentage of switching requests that experience no delay, and the y-axis is the concurrent channel time, T. The main conclusion is that by using this very simple scheme we can reduce zapping delay to a significant number of switching events. For example, by pre-joining only 2 neighbours, the previous and the next channel, for only one minute, the delay is reduced to virtually zero to around 45% of the switching events. If the access link has enough bandwidth available to increase the number of neighbouring channels to eight, around 60% of all switching requests experience no delay. It is important to underline that when a user watches a long program, without switching channels for an extended period of time, then *any* scheme except the one that always joins the predicted channels will have a delay. As a final note, only 2 to 3% of the requests will experience partial delay in all cases. For this reason, and to keep the presentation of the results as clear as possible, I do not include this information in the figures.

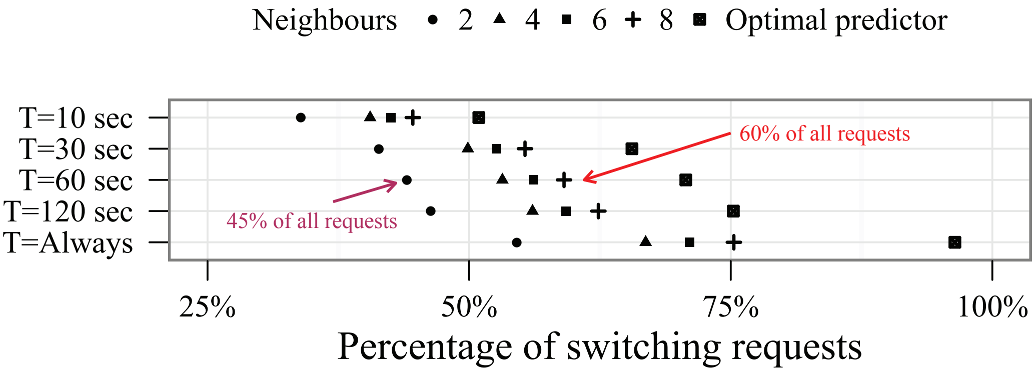


Figure 5.4: Percentage of requests that experience no delay by using the simple scheme, for various values of channel concurrent time T and number of neighbours

In Figure 5.5 I consider only zapping periods. Focusing on zapping periods in detail is important, because arguably it is in these periods that the user expects a swift zapping experience. I consider that a user is in "zapping mode" if the time between consecutive switching requests is less than one minute (again, in accordance with previous research [42] and Nielsen Media Research [147]). Therefore, all events for which the switching time was above one minute were removed (for this reason, I logically do not include the results with *concurrent channel time* above one minute). One can see that, for example, by pre-joining only 4 neighbours for one minute, more than 70% of the switching requests during zapping periods will experience virtually no delay.

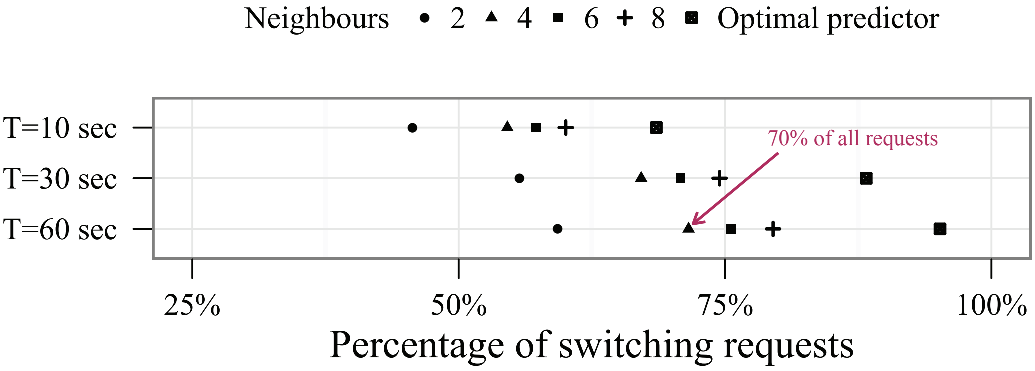


Figure 5.5: Percentage of requests that experience no delay by using the simple scheme during zapping periods only, for various values of channel concurrent time T and number of neighbours

The pre-joining schemes that are the subject of this study involve the prediction of the next channel an IPTV user is going to switch to. Obviously, sometimes these predictions are wrong, so it is important to understand how these schemes compare with an algorithm based on complete knowledge. Such a predictor always knows to which TV channel the user will switch next, and is therefore used as a benchmark for comparison. For convenience, I refer to this predictor as "optimal". Such "optimal" predictor would not do much better than most of the schemes I tested, as can be attested from the previous figures[1], where I already included its results. To make this comparison clearer, in Figure 5.6, I show the *performance gap* of each of the schemes under evaluation to the optimal predictor. This performance gap is defined as the difference between the percentage of requests the optimal predictor would benefit and the same percentage using the simple scheme. It is interesting to note that the simple scheme produces results that are not very distant from the optimal. An optimal scheme would not perform much better than a scheme that sends 6 or 8 neighbours, for instance.

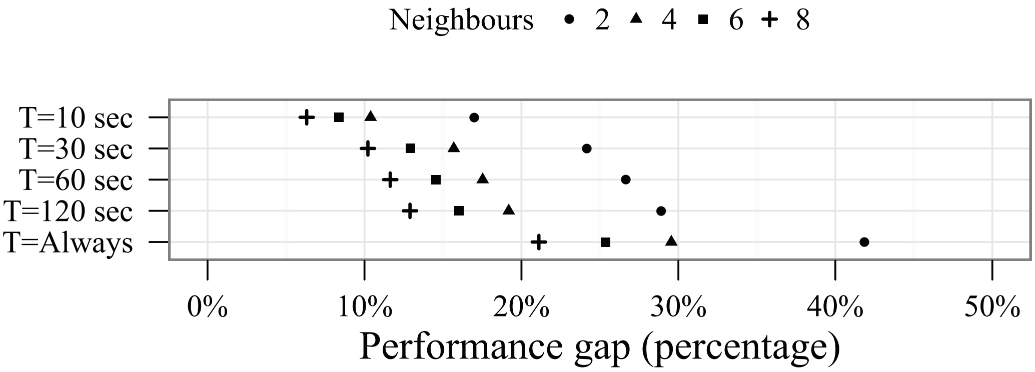


Figure 5.6: Performance gap between optimal predictor and the simple scheme for various values of channel concurrent time T and number of neighbours

Currently, most IPTV service providers distribute TV content in SD format encapsulated as MPEG-2 streams, so each TV channel needs 4 Mbps of guaranteed bitrate. As explained before, it is important that the access link can accommodate the distribution of several TV channels concurrently. I assume that is the case. However, when several TV channels are sent in the access link other broadband applications (P2P, web browsing, etc.) are affected, so it is important to quantify its impact. That is the purpose of Figure 5.7. In this figure I illustrate the average bandwidth consumption across the trace period, to understand the impact these simple schemes will have in this particular. One can observe that by limiting the *concurrent channel time* to zapping periods only, the average bandwidth is very close to the 4 Mbps current IPTV services usually require. This is due to the fact that zapping periods are relatively rare events during the course of a normal day [166]. So, even though the access link will have to accommodate peaks of high bandwidth consumption[2] during zapping periods, these average out during the course of the day.

5.5 Pre-joining popular TV channels

Some measurement studies on IPTV analysed channel popularity in detail, concluding that TV channels popularity is highly skewed and can be characterised by a Zipf-like distribution for top channels and an exponential distribution for non-popular ones [42, 161]. An interesting question

[1] It is worth noting the (at a first glance) curious fact of the optimal predictor not achieving 100% even with $T = always$. The reason is the 2 to 3% of the requests that experience partial delay.

[2] Note that the plot $T = always$ in the figure also represents these peaks.

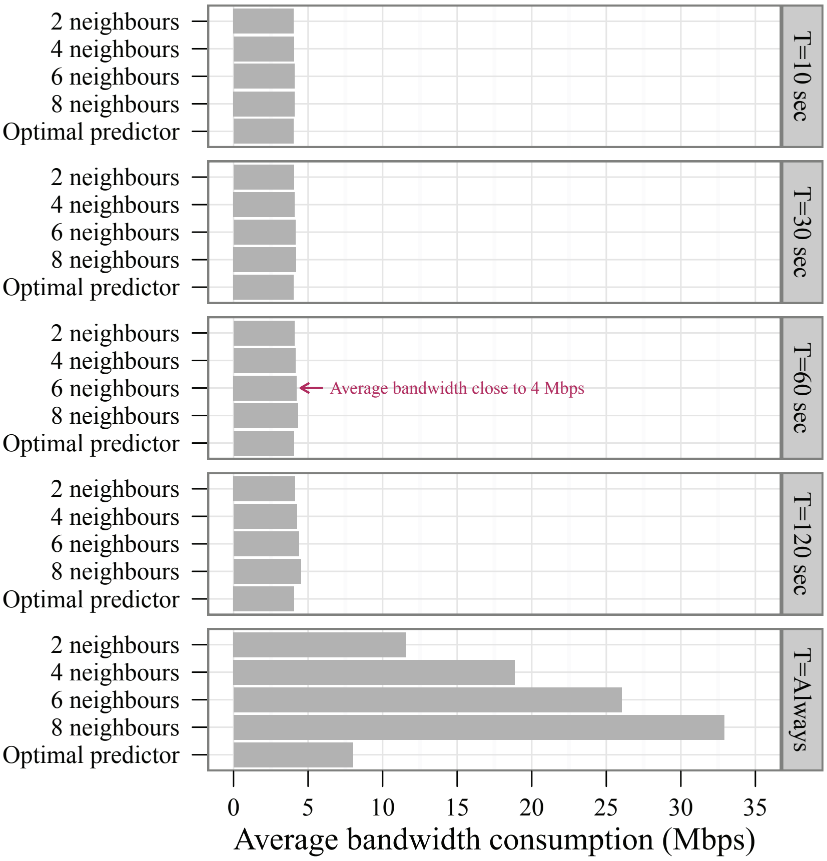


Figure 5.7: Average bandwidth consumed by the simple scheme for various values of channel concurrent time T and number of neighbours, compared with an optimal predictor

is thus to investigate if pre-joining the most popular channels is effective in reducing channel switching delays. In fact, recent studies on pre-joining schemes to reduce channel switching delay have considered this variable [150]. In this section I evaluate a scheme where the seven most popular TV channels[1] are pre-joined, and hybrid schemes, where the STB pre-joins both the seven most popular channels and a subset of neighbours. There are two reasons why I consider here the seven most popular channels, instead of any other number. First, these are the national, free-to-air broadcast channels. Second, I tested the scheme considering a different number of popular channels, and the conclusions to be drawn are the same. The results can be analysed in Figure 5.8. To make the distinction clearer, I use different data point types for the hybrid ("popular channels included") and the neighbours-only ("popular channels excluded") schemes. I considered a concurrent channel time of 2 minutes in this analysis.

By pre-joining the seven most popular channels, the number of switching requests that experience a small channel change delay is less than 15% of all requests. This result is quite poor when compared with any of the schemes where neighbouring channels are pre-joined. Also, the hybrid schemes show a very small improvement over pre-joining the neighbours only. Since pre-joining all these popular channels represents a very significant increase on the peak access

[1] Considering the channel popularity analysis presented in Chapter 4 (Section 4.2.4).

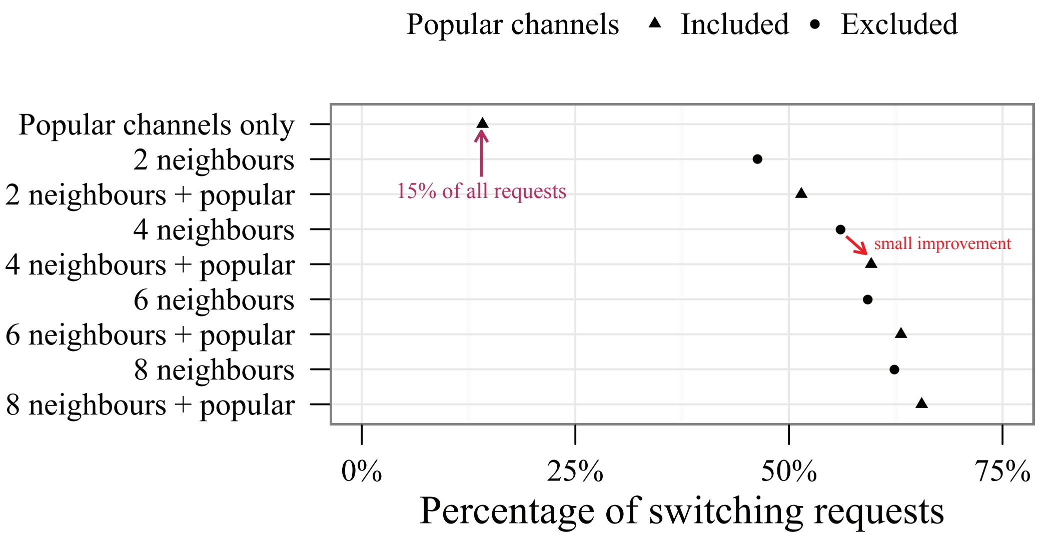


Figure 5.8: Percentage of requests that experience no delay by pre-joining the seven most popular channels, for a channel concurrent time T equal to two minutes. Different data point types are used for the hybrid ("popular channels included") and the neighbours-only ("popular channels excluded") schemes

bandwidth (and on the STB processing requirements), I conclude that pre-joining the popular channels is not an effective scheme in reducing channel change delay.

5.6 Personalised scheme: tracking user behaviour

The simple scheme evaluated in Section 5.4 offers interesting results. Notwithstanding its simplicity, a significant number of requests are to channels available in the STB, resulting in no zapping delay experienced by the user. But will *all* users experience such benefit? To assess this, I invite the reader to look at Figure 5.9. Here, I show the results of using the simple scheme presented in Section 5.2, but instead of the average, as before, I now present the median, 5th and 95th percentile, to understand how the scheme performs on a per-user basis. It is clear from the figure that the variance is high. Some users would benefit significantly from using the simple scheme (the "linear zapping fans"), but others would experience a smaller improvement.

My main objective in this section is to try to reduce the variance of Figure 5.9 without decreasing its median. Put in other words, the aim is to improve user experience for a wider audience, but doing so without affecting the experience of the "linear zapping fans". For this purpose I devise a personalised scheme. The idea is to track user actions to build a prediction model. To achieve this each STB will record and maintain information on both the probability of the user zapping linearly (to maintain the performance for the "linear zapping fans") plus the probability of him or her switching to each of the different channels (trying to capture their "favourite" channels, to improve the performance for other type of users).

This scheme requires two data structures:

1. A channel popularity vector, CP. This vector maintains a counter for each particular

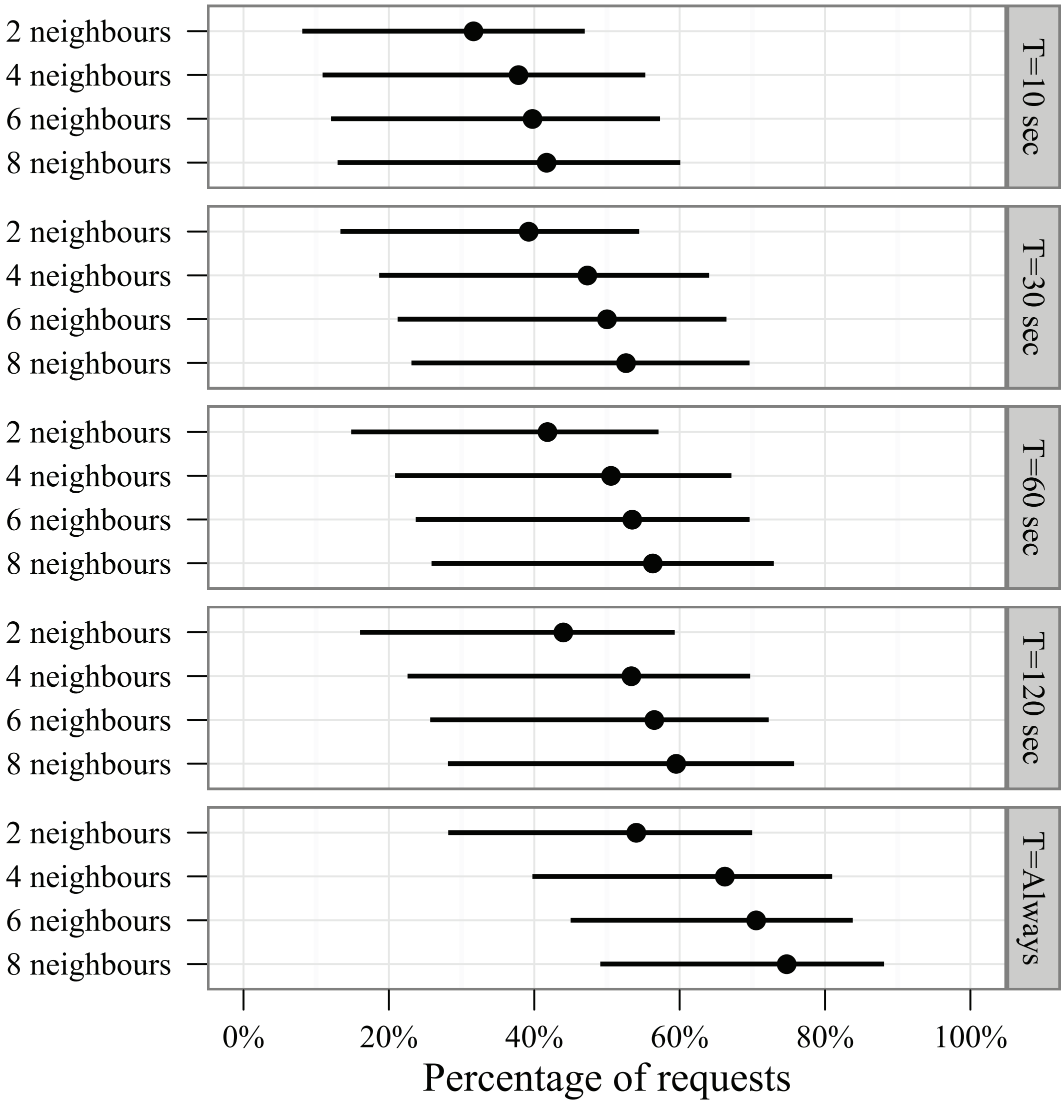


Figure 5.9: Variance of the percentage of requests that experience no delay by using the simple scheme for various values of channel concurrent time T and number of neighbours. In the graph the median, 5th and 95th percentile are presented

channel. Every time the user switches to a new channel, its counter is incremented by one. The favourite channel of a particular user will be the one corresponding to the maximum value present in the vector.

2. A "jump distance" popularity vector, JDP. This vector maintains a counter for each particular "jump distance". Every time a user jumps from channel number i to channel number j this distance is calculated as $j - i$ and the counter for this particular distance is incremented by one. A "linear zapping fan" that enjoys switching mostly to the next and to the previous channel will have maximums at positions $JDP[1]$ and $JDP[-1]$.

From these two vectors I then get the N TV channels that correspond to the highest probabilities of the set $[CP; JDP]$. N is the number of additional channels to be joined concurrently with the requested one. So, depending on user behaviour, the STB will either pre-join some neighbours, or some favourites, or a mix of neighbours and favourites.

Another important aspect of user behaviour is its temporal dynamics. It is known [42, 160, 161] that user behaviour changes with the time of day (for instance, morning *vs* evenings), day of the week (weekend *vs* weekday), and even period of the year (holiday *vs* working period). Besides this fact, a Set Top Box is usually shared by many people in one household, and different people may have very different behaviours (a child that zaps constantly *vs* the grandparents that settle in very specific channels). Considering this, I test the personalised scheme considering two types of vectors:

1. Non-ageing vectors. In this case, the two vectors CP and JDP do not age.

2. Ageing vectors. In this case, the two vectors CP and JDP age. The objective is to capture changes in user behaviour. The vectors age in accordance with Equation 5.1, an exponential moving average. The coefficient α represents a constant smoothing factor between 0 and 1. A higher α discounts older observations faster. $O[k]$ is a vector representing the k-th observation. One of its elements will be equal to 1 (corresponding to the channel switched to), while the others are all zero. $V[k]$ represents the value of each counter (in the vector considered) at the time of the k-th observation.

$$V[k+1] = \alpha O[k] + (1-\alpha)V[k] \tag{5.1}$$

5.6.1 Evaluation

As with the previous schemes, this one is evaluated by means of a trace-driven analysis using the method explained in Section 5.3. I evaluate this scheme for a single value of the concurrent channel time, 60 seconds. I also restrict the number of pre-joined channels to 2, 4 and 10. I tested the scheme for various values of α, ranging from 0.01 to 0.9. For clarity sake, only a subset of the results is presented in Figure 5.10 (specifically: two additional channels, 60 seconds of concurrent channel time, and six values for the parameter α).

As explained before, the main goal of this scheme is to reduce the variance of Figure 5.9. As a measure of the variability of the results, I analyse the standard deviation, represented graphically in the figure. As can be observed, the standard deviation value is basically the same for any scheme tested. The use of this scheme decreases very slightly the standard deviation, but the decrease is insignificant and is therefore imperceptible in the figure. In conclusion, this personalised scheme does not improve user experience when compared with the simple scheme analysed in Section 5.2. This result implies that the benefit of the scheme arises solely from maintaining the neighbours as an option. There was no gain in adding the favourites option. I speculate that users that are not “linear zapping fans” are not “zapping fans” in general. It is left as future work to understand if that is the case, and also devising other techniques to improve the zapping experience for a wider population.

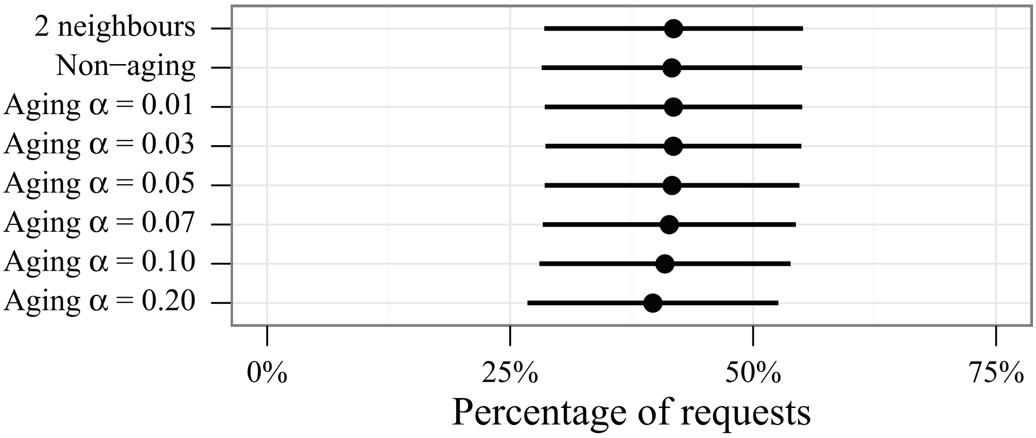


Figure 5.10: Variance of the percentage of requests that experience no delay by using the personalised scheme for a channel concurrent time T equal to 60 seconds, 2 neighbours, and several values of α. In the graph the median and the standard deviation are presented

5.7 Discussion

The pre-joining schemes evaluated in this chapter include several positive points:

1. They eliminate channel change delay for a very significant percentage of requests.
2. There is no user perceived picture distortion during the zap process, since no extra low bandwidth streams are used for the zapping period and no low quality video is needed.
3. There is no bandwidth increase in the core, regional, or metro networks. Also, the increase of the average access bandwidth is residual. It is a fact that the peak bandwidth increases (I discuss this increase below), but since zapping periods are rare events in the course of a day, the access bandwidth is not affected on average.
4. No changes need to be made to the core of the network, to any network element, or to the media server. There is no need for extra servers in the network. The only change needed is an upgrade of the STB software. This is therefore a cheap solution, very simple to implement.

There is no perfect scheme, of course, and this one is not without its drawbacks:

1. Not all requests are improved, so the user experience will vary: some requests will experience virtually no delay, while others will experience the normal IPTV delay, which is high.
2. The access network will need to accommodate the peak bandwidth for sending several channels in parallel, during zapping periods. In this study I considered sending between 2 to 8 channels in parallel. As SDTV channels using MPEG-2 require around 4 Mbps guaranteed bit rate per channel, sending this number of channels in parallel increases the bandwidth requirements proportionally. The access networks in most developed countries

already offer tens of Mbps of average download speeds and these are expected to increase in the near future [157], so this peak bandwidth is attainable on the access link.

3. The Set Top Box needs to be able to synchronise and buffer several TV channels in parallel, which may increase its cost.

Overall, the pre-joining schemes studied in this work offer interesting results. They have, however, the downsides referred to above. To mitigate these drawbacks it is, first of all, necessary to control the number of TV channels sent in parallel, matching it to the resources available (this would lessen problems number 2 and number 3 above). Also, other schemes may be used in parallel with this one to overcome the fact that user experience is variable (problem number 1). For example, when the user requests a channel that is not present in the STB, a boost stream could be requested as in the schemes presented in Chapter 3 (Section 3.2.2). Such hybrid scheme could be an interesting proposition in terms of quality of experience and cost. User experience would not vary, since all requests would experience reduced channel change delay. The cost of the solution would be lower than a pure boost-stream solution as fewer requests would be sent to the network, due to the fact that a significant percentage of requests would be served by the STB alone. In consequence, fewer dedicated servers would be required and the overall solution would be cheaper.

5.8 Conclusions

In this chapter, I investigated a specific type of techniques to reduce channel change delay in IPTV networks, namely, predictive pre-joining of TV channels. In these schemes each Set Top Box (STB) simultaneously joins additional multicast groups along with the one that is requested by the user, thus anticipating future user behaviour. To evaluate these schemes I have performed an empirical analysis using the dataset described in Chapter 4. To the best of my knowledge, this is the first empirical study where channel change delay reduction techniques are evaluated using real IPTV usage data.

In the first scheme evaluated the neighbouring channels (TV channels adjacent to the requested one) are pre-joined by the Set Top Box during zapping periods, simultaneously with the one requested. Thus, in the event of the user switching next to any of these channels, switching latency is virtually eliminated. As TV users enjoy zapping linearly — i.e., they tend to switch up and down using the remote control — this scheme seemed favourable. Indeed, the main conclusion of my analysis is that by using such a simple scheme, the zapping delay can be virtually eliminated for a very significant percentage of channel switching requests. As an example, by sending the previous and the next channel concurrently with the requested one, for only one minute after a zapping event, switching delay is eliminated for around half of all channel switching requests. This is achieved with a negligibly increase of the average bandwidth utilisation in the access link.

I have compared this simple scheme with an ideal predictor, having realised that its performance was close to the optimal case. However, user behaviour can vary significantly, leading to a high variation of the results: although some users would benefit tremendously from using this simple scheme, others would see only a relatively small improvement. To address this problem I designed and evaluated other schemes. The first was to pre-join popular TV channels, but this scheme proved inefficient. The second was a personalised scheme where user behaviour is tracked. The results showed the improvement over the simple scheme was statistically insignificant.

While in this chapter I was concerned with a specific aspect of IPTV user's quality of experience, in the next two chapters I will change the focus to the design and operation of an IPTV network, by proposing novel techniques to increase its resource and energy efficiency.

Chapter 6

Resource and energy efficient network

The previous chapter was devoted to the improvement of IPTV users' quality of experience. In the next two chapters the focus moves to the design and operation of IPTV networks. In these chapters I propose novel techniques to increase the resource and energy efficiency of IPTV infrastructures. The first such technique is based on a simple paradigm: "Avoid waste!" [167].

IPTV services are bandwidth intensive, requiring low latency and tight control of jitter. To guarantee the quality of experience required by its customers, service providers opt to build *static* multicast trees for the distribution of TV channels. Referring to the reference architecture presented in Figure 2.5 (Chapter 2), this means *all* DSLAMs join *all* multicast groups (they thus receive content of *all* TV channels). As particular channels have no viewers at particular time periods, this method is provably resource and energy inefficient. In this chapter, I argue that the expected increase in the quantity and quality of the TV channels distributed in IPTV networks will become a serious issue, bandwidth and energy wise. To alleviate this problem, I propose a *dynamic* scheme where only a selection of TV multicast groups is joined by the network nodes, instead of all. This scheme is evaluated by means of a trace-driven analysis using the dataset described in Chapter 4. The objective is to study the tradeoff between the bandwidth savings of using this technique and the number of requests that will experience higher channel change delay as a consequence.

I demonstrate that by using the proposed scheme IPTV service providers can save a considerable amount of bandwidth while affecting only a very small number of TV channel switching requests. To understand how these bandwidth savings are translated into energy savings, I also develop a power consumption model for network equipment based on real measurements reported recently in the literature. I conclude that while today the bandwidth savings have reduced impact in energy consumption, with the introduction of numerous very high definition channels this impact will become significant, justifying the use of resource and energy efficient multicast distribution schemes.

6.1 Introduction

We have been depleting the natural environment since the times of the industrial revolution. There is a scientific consensus that our planet will be unable to provide long-term support if this trend persists. Today, the Internet (excluding home networks, PCs and data centres) consumes about 0.5% of the current electricity supply of a typical OECD nation. Although this still represents a relatively small share of the global energy consumption, this fraction is expected to increase quickly [14, 197]. By recognising this fact, several consortiums are working to improve the ICT sector's energy efficiency. GreenTouch [89], for instance, aims to increase network energy efficiency by a factor of 1000 from current levels by 2015. To accomplish its goal, it focuses on the design of new network architectures and on the creation of the enabling technologies on which they are based.

IPTV is a resource intensive service with stringent quality of service requirements. It requires high bandwidth, low latency and low jitter. As explained in Chapter 2, each video stream is encoded at a bit rate that can vary from around 4Mbps (SDTV) to 20 Mbps (HDTV). In the future this figure may increase by one or two orders of magnitude, with the advent of ultra high definition video standards (2K, 4K, UHDTV) [83]. Besides the increase in the resolution of each TV channel, and its consequent bandwidth requirements upgrade, the number of TV channels offered is also expected to increase. AT&T already offers 700 TV channels [160] for their IPTV customers. According to a recent press release by the European Commission [57], over one thousand channels have been established in the UK alone until 2009. But recent trends anticipate the likely growth of the number of TV channels in the near future. Narrowcasting services — broadcasting to a very small audience [133] —, for example, are growing in importance. Niche channels are emerging to offer TV services to narrowly targeted audiences [26, 170]. An extreme example of this type of services was recently announced by Portugal Telecom, the largest telecommunications service provider in Portugal. As part of its IPTV service, Meo, Portugal Telecom is now offering the possibility for any customer to create his or her own TV channel [159]. At the time of writing ten thousand TV channels have already been created [60], and some are quite successful. This trend is expected to continue in the future as there seems to be clear market opportunities in offering this type of long tail service [12]. This calls for novel, efficient distribution schemes.

Unfortunately, current IPTV networks are not efficient. Service providers opt to distribute all TV channels, continuously, everywhere. Referring to the network architecture presented in Figure 2.5 (Chapter 2), this means *all* DSLAMs join *all* multicast groups. Originally, IP multicast had a dynamic nature, but IPTV providers opted for *static* multicast to guarantee the quality of experience required by its users (i.e., to guarantee a small channel change delay, as explained before), and to reduce service complexity (in terms of state and control traffic overheads). But static multicast is provably inefficient, as is demonstrated below. As soon as the number of channels surpasses the number of users at a certain access node, sending all channels is wasteful. Also, as already mentioned in Chapter 4 (Section 4.2.4), recent work [42, 161] has shown that channel popularity is highly skewed (following a Zipf-like distribution). While a small number of channels is very popular, dozens or even hundreds of TV channels are very rarely watched. Service providers seem to recognise this problem and are already concerned with the efficiency of their IPTV networks [40].

By analysing the dataset described in Chapter 4, I indeed observe that for the most popular channels there is always at least one viewer per access node (DSLAM), at any one time. In this chapter I call a channel that has at least one viewer in a particular network node (be it a DSLAM or a router) an **active channel** in that particular node. Despite some channels always having viewers, the number of active channels in each DSLAM is rarely above 60. This can be seen in Figure 6.1. This graph presents the average number of actives channels in every DSLAM and regional-core router in the network, as a function of the number of users. This figure is based on analysis of the whole dataset (255 thousand users, 6 months). It can be clearly observed that the network is wasting resources by distributing all 150 TV channels everywhere. In the figure, the shaded area represents the bandwidth savings opportunities for an IPTV service provider. With the likely increase of the number of channels (and of its bandwidth requirements) this inefficiency may become problematic.

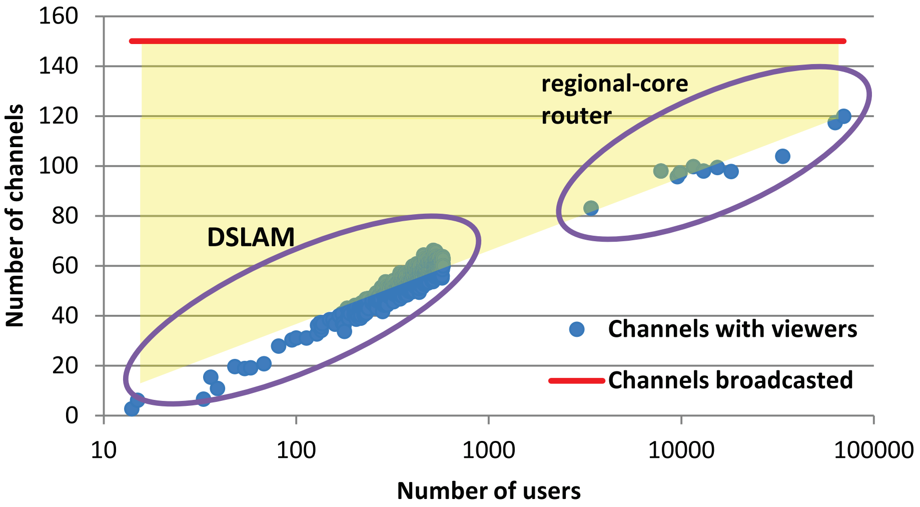


Figure 6.1: Average number of active channels (TV channels with viewers) per network node (including DSLAMs and core-regional routers). Nodes are ordered by the number of users they serve

Considering the above, I propose to reduce these inefficiencies by not building static multicast trees, i.e., not distributing all TV channels, continuously, everywhere. Instead of the network nodes joining all multicast groups, I propose each node to join only a *limited selection* of channels. A relevant point about this scheme is its dynamic nature. This is important due to channel popularity dynamics [161]. This design goal is fulfilled by each network node joining only the active TV channels plus a small subset of the inactive ones. This scheme is dynamic because the list of joined channels varies with user activity.

To evaluate the proposed scheme I perform a trace-driven analysis on the dataset described in Chapter 4. I evaluate this scheme in two ways. First, I analyse the tradeoff between the bandwidth savings of using this technique and the number of requests affected. A request is considered affected when the user requests a channel that in that particular moment is not part of the "selected channels" list of that node (i.e., that node has not joined that particular multicast tree). In such case, the user will experience a higher-than-usual channel change delay. This is due to the fact that a join message to that multicast channel will have to go up towards the source until it finds the nearest leave of the multicast tree. As I explained in the previous chapter, channel change delay is a problem in IPTV networks, so it is important the number of affected requests to be as low as possible to avoid jeopardising service quality. In addition, reintroducing dynamics in the multicast network reintroduces protocol overhead. Control messages start flowing in the network as multicast trees are joined and pruned. This also calls for a tight control over the number of requests affected. Second, I analyse how these bandwidth savings translate into energy savings. For this purpose I create a power consumption model for routers, based on real measurements [184], and evaluate it considering several scenarios.

The results show that it is possible to significantly reduce the bandwidth used by IPTV services in the network, while affecting only a very small number of switching requests. Considering current IPTV service offerings (hundreds of SD or HD TV channels), these savings have negligible impact on energy consumption. Considering futuristic scenarios, with IPTV offers of thousands of very high definition TV channels, the savings of using such dynamic multicast scheme become meaningful. The relative advantage of the proposed scheme is even more significant if one considers the use of equipment with more energy-proportional [24] power profiles.

The rest of the chapter is organised as follows. In Section 6.2 I present the scheme proposed in this chapter: selective joining. I then present the methodology used to evaluate this scheme in Section 6.3, and evaluate it in Section 6.4. I analyse the impact of using the proposed scheme on energy consumption in Section 6.5. The effects of using this scheme on channel change delay are briefly discussed in Section 6.6, and the chapter closes with Section 6.7.

6.2 Selective joining

Currently, IPTV operators distribute all TV channels continuously everywhere, in order to minimise channel change delay. All access nodes in the network join all TV multicast groups. This means that all DSLAMs are leafs of the multicast tree of every TV channel.

My proposal, *selective joining*, is for each node to join only *a subset* of the complete selection of TV channels. Namely, each node will join:

1. the channels for which there are viewers (the active channels) plus

2. a small subset of inactive channels. Inactive channels have no viewers in the node under consideration. I call the number of inactive channels that are joined by the node the *size of the inactive joined set*, or `inactive_set_size`.

This scheme requires a single data structure to be maintained at each network node, containing two elements: one to store information on the joined channels, `joined_set`, and another to record the number of viewers for each joined channel, `num_viewers`. An inactive channel will have its corresponding `num_viewers` variable equal to zero.

The proposed scheme is a form of hybrid multicast. The interaction between the user and the DSLAM is still via normal IGMP, but static multicast is replaced by a semi-dynamic multicast in the IP network. The dynamic nature comes from the `joined_set` always including the active channels. For this reason, the "selected channel" list changes *dynamically* with user demand. At the same time, the channels that have an higher probability of being watched in the future are "automatically" joined. Popular channels are the ones people watch more, hence there is a high probability of them having at least one viewer, which means they are usually included in the `joined_set`. In summary, a largely static group of popular channels will be kept in the list while a dynamic group of less popular channels leaves and joins with channel popularity dynamics. Selective joining can thus be seen as a form of cross-layer optimisation, using user level information about content popularity to drive the hybrid protocol.

6.3 Methodology

The scheme proposed in this chapter is evaluated by means of a trace-driven analysis. The IPTV trace detailed in Chapter 4 is used as input to the analysis performed. I analyse the use of the proposed scheme in two particular nodes in the network topology (I refer the reader to Figure 2.5 again): the DSLAMs and the core-regional routers. The reason why I chose these two particular locations is related to the information I can extract from the data trace, as will be made clear in the following.

Recall from Chapter 4 (Section 4.2.3) that I parsed the raw IPTV trace data to create a single file *per STB*. Each file includes all channel switching requests made by the users of a specific household. The first line of each file included the DSLAM IP address users send their IGMP signals to. With this information I am able to create a single time-ordered trace file that includes all switching events sent to a *specific DSLAM*. This allows the evaluation of this scheme at the DSLAM level. An example of the new version of the trace, for a particular DSLAM, follows.

```
Jul 1 00:41:36|UP|23|10.74.59.98
Jul 1 00:41:44|DOWN|23|10.74.59.98
Jul 1 00:41:44|UP|25|10.74.77.101
Jul 1 00:41:48|UP|182|10.74.80.80
Jul 1 00:41:48|UP|23|10.74.120.1
Jul 1 00:41:51|DOWN|182|10.74.80.80
Jul 1 00:46:32|DOWN|23|10.74.120.1
Jul 1 00:47:04|DOWN|25|10.74.77.101
```

As can be seen, the difference from the parsed version illustrated in Chapter 4 (Section 4.2.3) is the inclusion of information on the STB that sent the IGMP request. This information allows me to understand how many viewers each channel has at each moment, *per DSLAM*.

I now refer the reader again to the reference network topology presented in Figure 2.5. The IP network usually has a two-level, hierarchical structure [79]: the regional and the core networks. That is the case of the network where the IPTV traces were collected from. One core-regional router[1] aggregates all traffic from a specific region. Thus, the traffic sent from all DSLAMs destined to the core goes through this router. Unfortunately, the IPTV traces do not include information about the region each DSLAM belongs to. Fortunately, there is very easy way to deduce this information from the data. The IPTV service studied includes as part of its channel bundle one or two regional channels *per region*. In the trace, each of these channels is numbered differently from the others. Instead of being just a single number, it also includes information on the region it is distributed on. For example, channel numbers `8-M` and `9-M` are the regional channels from Madrid. With this information I am able to create a single time-ordered trace file that includes all switching events sent to all DSLAMs in a *specific region*. This allows the evaluation of this scheme at the core-regional router level.

To evaluate the proposed scheme I developed a Python script that checks each line of the input trace, to obtain each switching event received by each node (a DSLAM or a core-regional router). The current switching event is analysed, and one of these actions is performed:

1. If it is an `UP` event, and

   a if the channel is in the `joined_set`, the channel was joined by the node before. The global counter `hit` and this channel's counter `num_viewers` are incremented. The counter `hit` counts the number of requests served quickly by this node.

   b if the channel is not in the `joined_set`, the channel was not pre-joined by the DSLAM. The global counter `miss` is incremented. The counter `miss` counts the number of requests that are not served quickly by this node. The node has to send the join message towards the source which increases the channel change delay. The channel number is added to the `joined_set`, and its `num_viewers` is set to 1.

2. If it is a `DOWN` event, decrement this channel's `num_viewers` counter. Then,

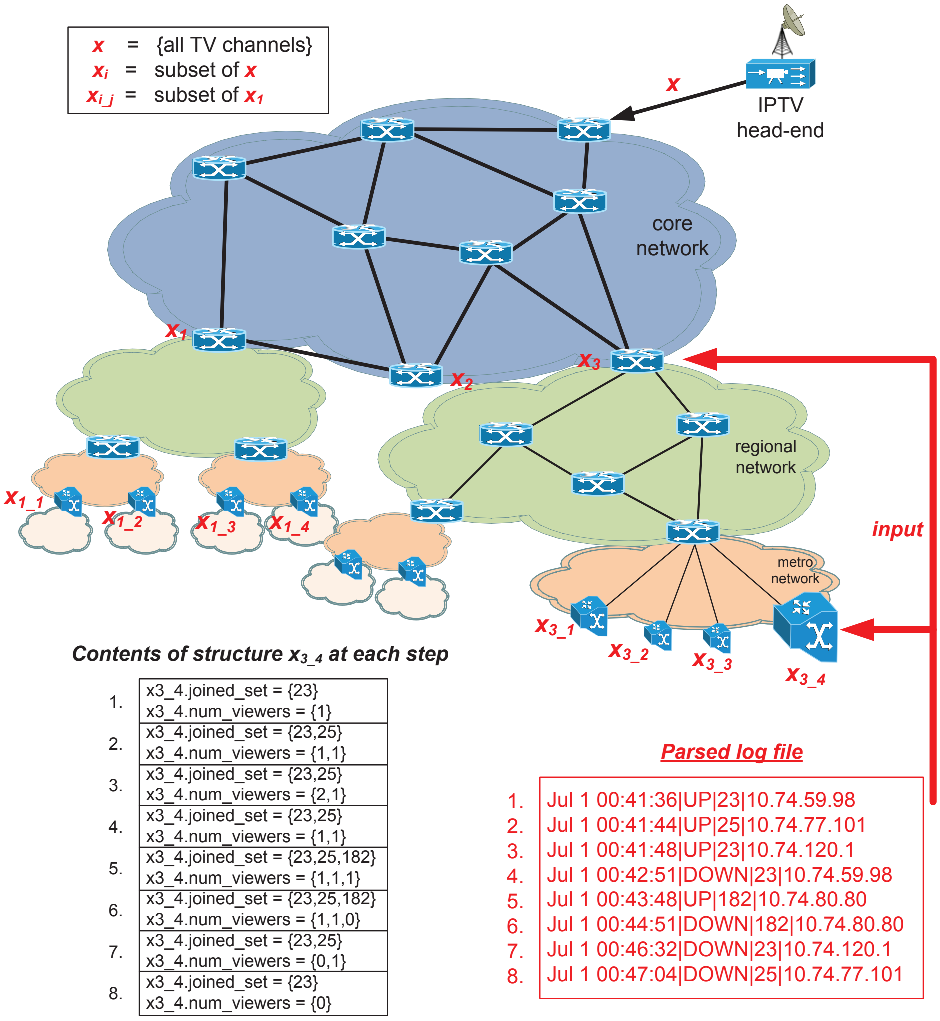
   a if the number of viewers is above zero, keep the channel in the `joined_set`.

   b if the number of viewers is equal to zero, the channel is inactive. In that case, if the number of inactive channels is below the `inactive_set_size`, keep the channel in the `joined_set`. Else, choose one of the inactive channels and remove it from the `joined_set`. In this case, the DSLAM sends an IGMP leave message from this channel towards the source.

Figure 6.2 illustrates the proposed methodology with a simple example[2]. I assume the sample trace is from a DSLAM, and that the `inactive_set_size` is equal to 1. This means there can be only one TV channel without viewers in this DSLAM. At 12:41:36am the DSLAM receives a join message for channel 23 from the STB with IP address `10.74.59.98`. As the channel

[1] In fact this router is replicated for reliability and dependability reasons.

[2] This figure is based on the reference architecture presented in Figure 2.5.



Contents of structure x_{3_4} at each step

1.	x3_4.joined_set = {23}
	x3_4.num_viewers = {1}

2.	x3_4.joined_set = {23,25}
	x3_4.num_viewers = {1,1}

3.	x3_4.joined_set = {23,25}
	x3_4.num_viewers = {2,1}

4.	x3_4.joined_set = {23,25}
	x3_4.num_viewers = {1,1}

5.	x3_4.joined_set = {23,25,182}
	x3_4.num_viewers = {1,1,1}

6.	x3_4.joined_set = {23,25,182}
	x3_4.num_viewers = {1,1,0}

7.	x3_4.joined_set = {23,25}
	x3_4.num_viewers = {0,1}

8.	x3_4.joined_set = {23}
	x3_4.num_viewers = {0}

Parsed log file

```
1. Jul 1 00:41:36|UP|23|10.74.59.98
2. Jul 1 00:41:44|UP|25|10.74.77.101
3. Jul 1 00:41:48|UP|23|10.74.120.1
4. Jul 1 00:42:51|DOWN|23|10.74.59.98
5. Jul 1 00:43:48|UP|182|10.74.80.80
6. Jul 1 00:44:51|DOWN|182|10.74.80.80
7. Jul 1 00:46:32|DOWN|23|10.74.120.1
8. Jul 1 00:47:04|DOWN|25|10.74.77.101
```

Figure 6.2: Proposed methodology

is not in the `joined_set` (maintained in the structure x_{3_4} in the figure), the global counter `miss` is incremented, and the DSLAM sends the join message towards the source. The channel number is added to the `joined_set` x_{3_4}, and its `num_viewers` counter is set to 1. This can be observed from structure x_{3_4}'s contents, step 1. Then, at 12:41:44am, the STB with IP address `10.74.77.101` sends a join message to channel 25, and a similar procedure occurs. Four seconds later the STB with IP address `10.74.120.1` sends a join message to channel 23. As the channel is in the `joined_set`, this time it is the global counter `hit` that is incremented. This channel's `num_viewers` counter is incremented (to 2). When the down message to this channel arrives at the DSLAM from STB `10.74.59.98`, its counter is decremented to 1. As there are still viewers the channel is kept in the `joined_set`. When the new IGMP down message to this channel arrives at the DSLAM, at 12:46:32am (step 7), its counter `num_viewers` is decremented to zero, which means there are no viewers for this channel. It became inactive. At that time, there is another inactive channel in the `joined_set` (channel 182; step 6). As there are two inactive channels and the `inactive_set_size` is equal to 1, one of these two channels is removed. In this example I assume the least recently watched channel is removed from the `joined_set` list (channel 182).

6.4 Evaluation

As I said in the previous section, to evaluate the proposed scheme I perform a trace-driven analysis on the dataset presented in Chapter 4. All results I present in this section arise from the analysis of the whole data set (6 months, 255 thousand users). I test two schemes to decide which channel to remove from the `joined_set` when a channel becomes inactive. The first is to remove an inactive channel randomly. The second is to remove the least recently watched channel. The two schemes produce indistinguishable results, so for clarity sake only the results from one scheme — the random — are shown.

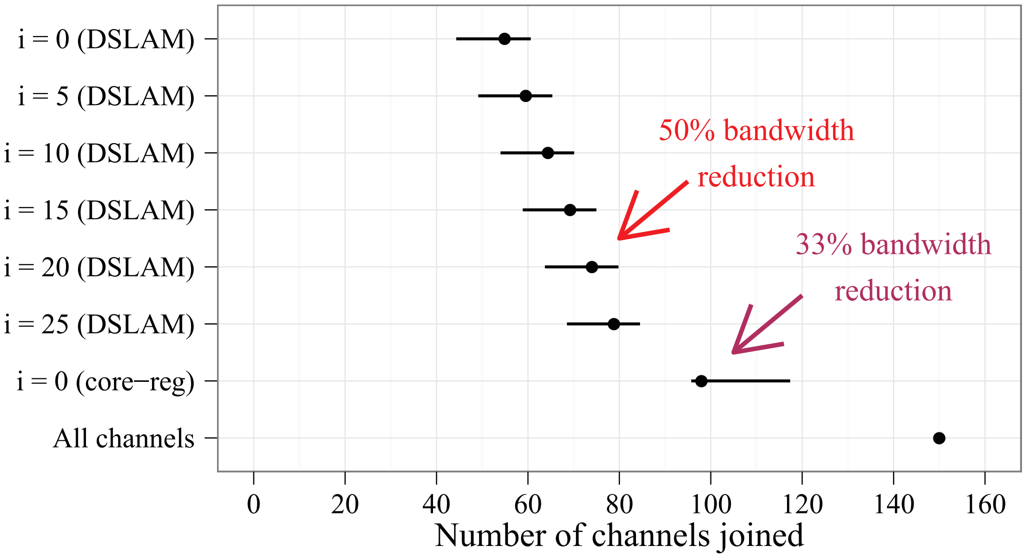


Figure 6.3: Number of channels joined when using the selective joining scheme for various values of the inactive set size

In Figure 6.3 I present a graph with the number of TV channels joined by each node (x-axis) as a function of the number of inactive channels that are joined by the node (the `inactive_set_size`, i in the figure). The figure presents the median, 10th- and 90th-percentile. I consider two types of nodes (for the reasons explained before): DSLAMs and core-regional routers. At the DSLAM level I present the results for five different values of the `inactive_set_size`, i. As can be seen, not joining all 150 TV channels represents bandwidth savings in the network. With an `inactive_set_size` of only 20 TV channels, bandwidth can be reduced by 50%. At the regional level ("core-reg" in the figure) I present the results considering $i = 0$ only, i.e., the core-regional routers join only the active channels[1]. In this case, the average bandwidth savings are equal to 33%. In the figure, I also compare the proposed scheme with the one currently used by IPTV providers ("All channels").

In order to analyse the effect these schemes will have on the quality of experience of IPTV

[1] The reason why I present only this value will become clear in the next paragraph.

Table 6.1: Description of the three scenarios

Scenario	Media format	Bit rate	TV channels	**Bandwidth savings**
150SD	SDTV	4 Mbps	150	**0.3 Gbps**
700HD	HDTV	20 Mbps	700	**7 Gbps**
3kUHD	4K	200 Mbps	3000	**300 Gbps**

users, I now inspect the percentage of requests to channels not joined by the node. As explained before, in this case the node has to send a join message to this channel towards the source which increases the channel change delay. This percentage is calculated as the value of the counter `miss` divided by the total number of requests. The results are shown in Figure 6.4, again for various values of i. For an `inactive_set_size` of 20, the percentage of requests affected at the DSLAM level is less than 2% on average. At the core-regional level this figure is almost negligible. In core-regional routers, joining active channels only thus seems a good option.

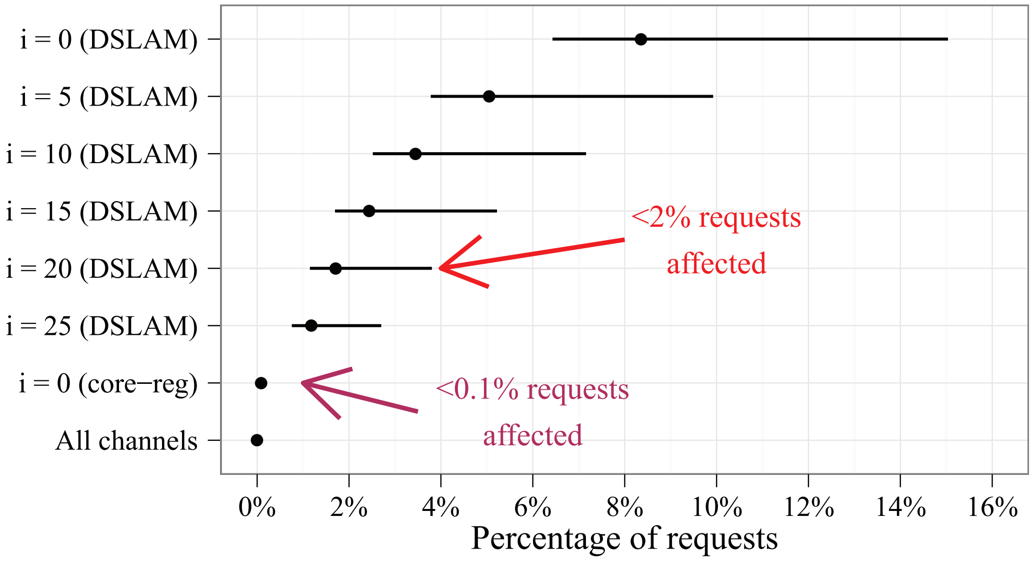


Figure 6.4: Percentage of requests affected for various values of the inactive set size.

A decrease in the number of TV channels joined by a node represents bandwidth savings. By joining fewer channels, the nodes processes, and the links transport, less bits. How significant these savings are, both in resource and energy terms, is therefore intrinsically dependent on the data rate at which channels are distributed. To clearly understand the significance of the savings achieved by the proposed scheme, I analyse three scenarios, characterised in Table 6.1.

In the first scenario, 150 TV channels are distributed in Standard Definition format (SDTV). This represents the IPTV service offering under analysis, at the time of trace collection. A bandwidth saving of 50% means reducing load by 300 Mbps. This is not very significant. Currently, however, most IPTV networks already offer more channels (AT&T offers 700 [160]) in high definition (HDTV). Assuming such scenario the bandwidth decrease now accounts to around 7 Gbps. Looking further into the future, as explained in section 6.1, one can anticipate many more channels and even higher quality streams (digital cinema standard 4K, for instance, or UHDTV [83]). In the futuristic scenario I therefore assume 3000 4K TV channels. In this case, the bandwidth savings are already very significant, with a magnitude of several hundred Gbps.

6.5 Impact on energy consumption

Saving bandwidth may be, *per se*, an important objective. Nevertheless, in this chapter I additionally analyse the impact the proposed scheme has on energy consumption. In principle, bandwidth savings should result in energy savings. Less bits need to be transported in the links, and less bits need to be processed by the routers. Also, reducing load in the network offers more opportunities to put some equipment to sleep or to adapt line rates, in order to save energy. In this section I try to understand if the bandwidth savings reported in the previous section are translated into relevant energy savings.

6.5.1 Power consumption model

To be able to quantify the energy savings achieved by using the selective joining scheme, in this section I build a power consumption model of a network node. Several factors affect the power consumption of such node [136]:

1. Base chassis power. This is the power to maintain the chassis on. It is a fix amount independent of load, including the power consumed by components such as fans, memory, etc.

2. Number of active linecards. A linecard is the electronic circuit that interfaces with the network.

3. Number of active ports in each linecard.

4. Port capacity. This is the line rate forwarding capacity of individual ports.

5. Port utilisation. This is the actual throughput flowing through a port, relative to its capacity.

Based on these variables, I use the following model of power consumption P of a router.

$$P = P_{ch} + \sum_{i=0}^{L} P_{l_i} \tag{6.1}$$

In equation 6.1 P_{ch} refers to the power consumption of the chassis. L is the number of linecards that are active, and P_{l_i} is the power consumption of linecard i. The power consumption of each linecard is calculated based on the model proposed by Sivaram *el al.* [184] for a NetFPGA card, and is presented as Equation 6.2. By using a high-precision hardware-based traffic generator and analyser, and a high-fidelity digital oscilloscope, the authors devised a series

Table 6.2: Linecard power profile

Energy component and description	Estimate from [184].
Power consumed by unconnected linecard card (P_c)	6.936 W
Power consumed per connected Ethernet port (P_E)	1.102 W
Per-packet processing energy (E_p)	197.2 nJ
Per-byte energy (E_b)	3.4nJ

of experiments allowing them to quantify the per-packet processing energy and per-byte energy consumption of such linecard.

$$P_l = P_c + KP_E + N_I E_p + RE_b \tag{6.2}$$

In this equation:

- P_c is the constant baseline power consumption of the NetFPGA card (without any Ethernet ports connected).
- K is the number of Ethernet ports connected.
- P_E is the power consumed by each Ethernet port (without any traffic flowing).
- N_I is the input rate in packets per second (pps).
- E_p is the energy required to process each packet.
- R is the traffic rate in bytes per second. I am assuming the input rate is equal to the output rate.
- E_b is the total per-byte energy. This includes the energy required to receive, process and store a byte on the ingress Ethernet interface; and the energy required to store, process and transmit a byte on the egress Ethernet interface.

The inputs to this model are presented in table 6.2, again based on the measurements reported in [184].

In Figure 6.5 I present the power consumption of a router based on this model, and assuming $P_{ch} = 430W$. This value for the power consumption of the chassis is based on power measurements of the Cisco GSR 12008 router, performed by Chabarek *et al.* [44]. This is the power profile for a router with four linecards with 4x1Gbps ports each. The plot presents power consumption as a function of traffic load. Note that the y-axis present values from $y = 400[W]$ to $y = 500[W]$ only. This is therefore a zoomed version of the power profile. The reason why I present it this way first is the fact that current network equipment is not energy proportional [24]. The baseline power (from maintaining the chassis powered on) is very high and is, by a large margin, the main component of router power consumption. But this zoomed version allows the observation of these relatively significant power consumption “jumps” at regular intervals. The small jumps represents turning on a new Ethernet port in the linecard, while the bigger jumps represent turning on one linecard.

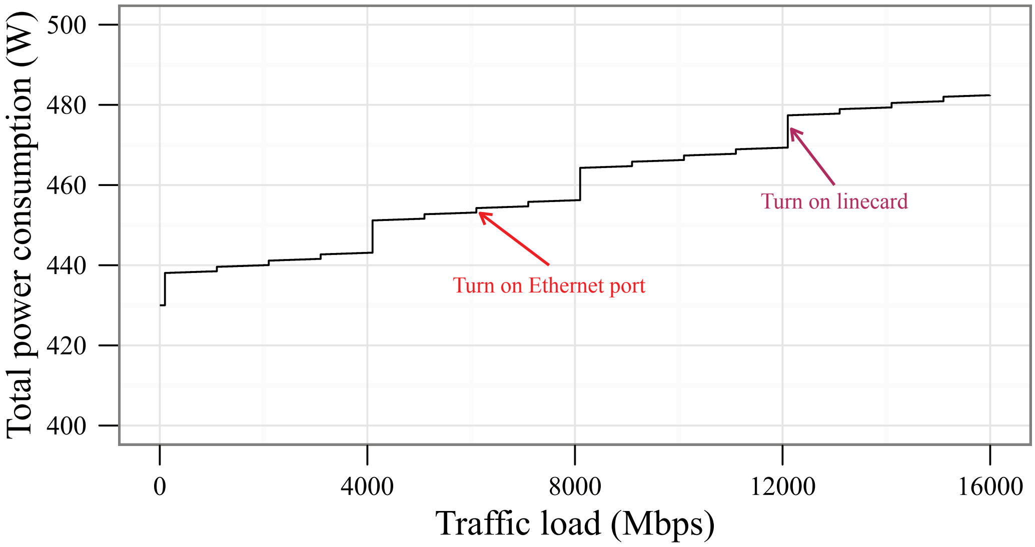


Figure 6.5: Power consumption model (zoomed)

To contextualise, Figure 6.6 shows the previous figure zoomed out. This figure clearly illustrates how far away current routers are from an energy-proportional behaviour. Anyway, with the ongoing green research novel network devices having lower energy when idle are expected in the future. I consider this trend in the analysis that follow.

6.5.2 Results

I now analyse how the bandwidth savings reported in Table 6.1 translate into energy savings. As shown in Section 6.4, the scheme proposed in this chapter — selective joining — allows an IPTV provider to reduce its network bandwidth consumption without affecting user experience significantly. In the analysis I consider the use of the selective joining scheme in an IPTV network with the following configuration. The DSLAMs join the active channels plus 20 inactive channels (i.e., they set their `inactive_set_size` to 20), and the routers join only the active channels (i.e., they set their `inactive_set_size` to 0). As illustrated in Figure 6.3, this represents an average traffic decrease of 50% and 33% to the DSLAMs and core-regional routers, respectively (while maintaining an acceptable quality of experience, as can be attested in Figure 6.4). Assuming such scenario, the traffic decrease in the regional network (I again refer the reader to the reference architecture, Figure 2.5) would be between 50% (decrease in DSLAMs load) and 33% (decrease in core-regional router load). In the core network, the traffic decrease would vary between 33% (core-regional router) and 20%. The justification for these 20% is given in Chapter 7. In that chapter I demonstrate that at any particular moment an average of one fifth of the TV channels does not need to be distributed in the IPTV network, as they do not have a single viewer.

I consider the three scenarios presented in Table 6.1 in the analysis: `150SD`, `700HD` and `3kUHD`. For the first scenario, I assume a router with four linecards with 4x1Gbps Ethernet ports each, as in Figure 6.6. For the second scenario I scale up the node to sixteen linecards of the same

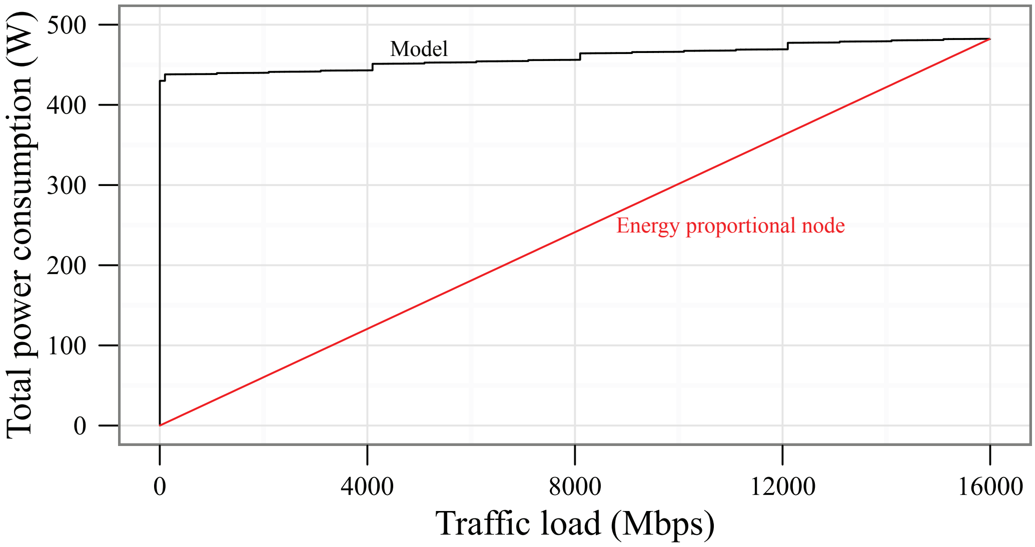


Figure 6.6: Current router power consumption *vs* energy-proportional node

type, for it to be able to handle the increased aggregate throughput. The capacity of each node is now assumed to be equal to 64Gbps. The capacity of the nodes of the third scenario has to scale up to the Tbps range. I assume fourteen 4x40Gbps linecards for an aggregate capacity of 2.2Tbps. This is a different type of linecard from the one measured by Sivaram *et al.* [184]. I therefore assume a 4x40Gbps linecard presents the same power profile as forty 4x1Gbps.

A final note concerning the assumptions made in this analysis. For their infrastructures to be reliable and to provide high performance, network providers build densely interconnected networks with many redundant paths [44]. In their networks, pairs of routers are typically connected by multiple physical cables that form one logical bundled link [69]. I therefore assume that any pair of routers will maintain multi-bonded channels to inter-communicate. I also assume the routers will use each of these parallel channels to its full capacity before deciding to use a new free channel, i.e., before turning on a new port/linecard.

The power savings for the three scenarios under consideration are presented in Figures 6.7 and 6.8, for the regional and core network, respectively. The graphics illustrate the relative power savings of using the selective joining scheme as a factor of the baseline traffic load according to equation 6.3. The baseline traffic load is the load of a node that does not use the proposed scheme. This load obviously includes IPTV traffic.

$$\frac{P\,(baseline) - P\,(selective\_joining)}{P\,(baseline)} * 100 \tag{6.3}$$

In Equation 6.3, $P\,(baseline)$ is power consumption at baseline traffic load, whereas $P\,(selected\_joining)$ is power consumption when using the proposed scheme (a lower value due to the decrease in IPTV traffic). In the figures I present results for baseline load values varying from 25% to 75%.

The results for the regional and core networks are presented separately, but as the conclusions

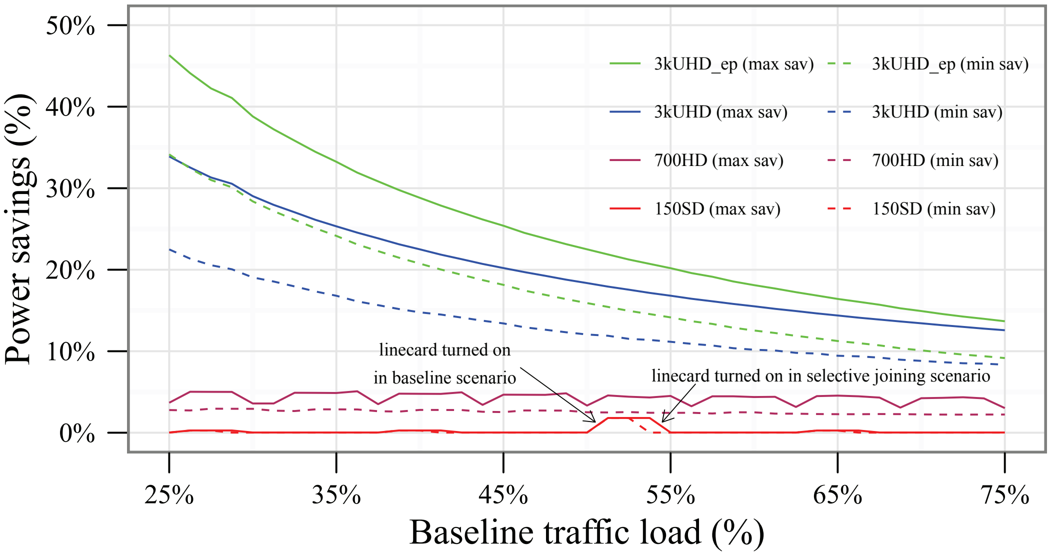


Figure 6.7: Power savings after introducing the proposed scheme as a function of the baseline traffic load in the node (regional network)

to be drawn are basically the same, their analysis is made jointly. For each scenario I present a line with maximum and minimum power savings, for the reasons explained in the paragraph that opened this subsection. The dashed lines represent the minimum savings obtained in each scenario, while the solid lines represent maximum savings. Each scenario is represented in different colours. The first conclusion one can draw from the observation of the graphs is that for current scenarios the bandwidth savings achieved with the proposed scheme will have negligible impact on energy consumption. In the 150SD scenario, the power savings would represent less than 1% on average, while in the 700HD scenario they would increase to just around 3%.

The most interesting results occur in a scenario with many TV channels at high resolutions, as the 3kUHD one. In particular under normal traffic load conditions, with values below 30%[1]. The advantage of using the proposed scheme seems clear. Considering such load conditions, the power savings are on the order of 30% in the regional network, and 20% in the core. The decrease observed as load increases was expected. As traffic load increases, the baseline power consumption (the divisor) increases faster than the relative power reduction (the dividend), and therefore the relative power reduction gain (the quotient) decreases. This is true in all scenarios, but is less pronounced in the first two, 150SD and 700HD, because the resulting power savings are small. For this reason this trend is only perceptible in the futuristic scenario.

One aspect that deserves explanation is the lines in the plots not being completely smooth (the little “steps”). This is particularly evident in the first two scenarios, 150SD and 700HD. The reason is that the x-axis represents the baseline traffic load in the node (without using the proposed scheme), while the power savings arise from the new traffic load (using the proposed

[1] Networks are provisioned at present for the worst case and many times overprovisioned. This is a design choice from network operators that allows them to protect their networks against multiple failures, to handle traffic variability and to support the rapid growth of traffic volume. According to a measurement study of Sprint’s backbone network presented some years ago [111], a very significant percentage of the network links (69%) *never* experienced a load above 30% in the period analysed.

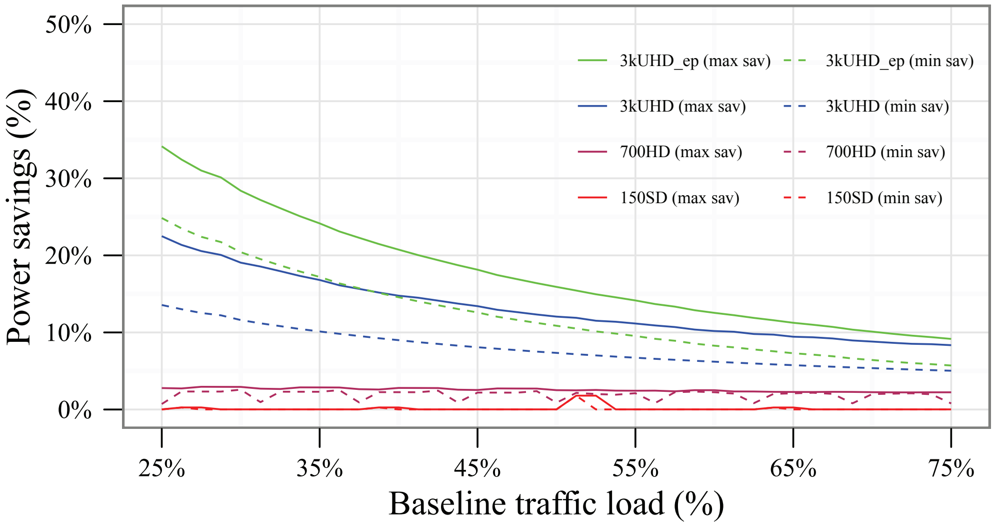


Figure 6.8: Power savings after introducing the proposed scheme as a function of the baseline traffic load in the node (core network)

scheme) being lower. The power saving peaks that appear in the graph represent transition points, when a particular event that increases significantly the energy consumption occurs: in this case, when an additional linecard needs to be turned on. For instance, in the 150SD scenario there is a peak precisely in the middle of the plot. This is because a 50% load in that scenario represents a data rate equal to 4Gbps. At this point, the network node has to turn on a new linecard (recall that I am assuming 4x1Gbps linecards). With the proposed scheme, the network load would be lower than the baseline traffic load, a bit under 4Gbps. So the linecard does not need to be turned on yet. While the traffic load does not increase over that transition point the proposed scheme therefore presents a higher-than-average power saving advantage. In the 700HD scenario the same occurs, but more frequently. This is due to the fact that in this scenario the network nodes have eight times more linecards, so the effect occurs eight times more than in the 150SD case. A similar effect occurs in the futuristic scenario. But, as the baseline power consumption is much higher than in the first two, the bumps are less pronounced, and are hence imperceptible in the figure.

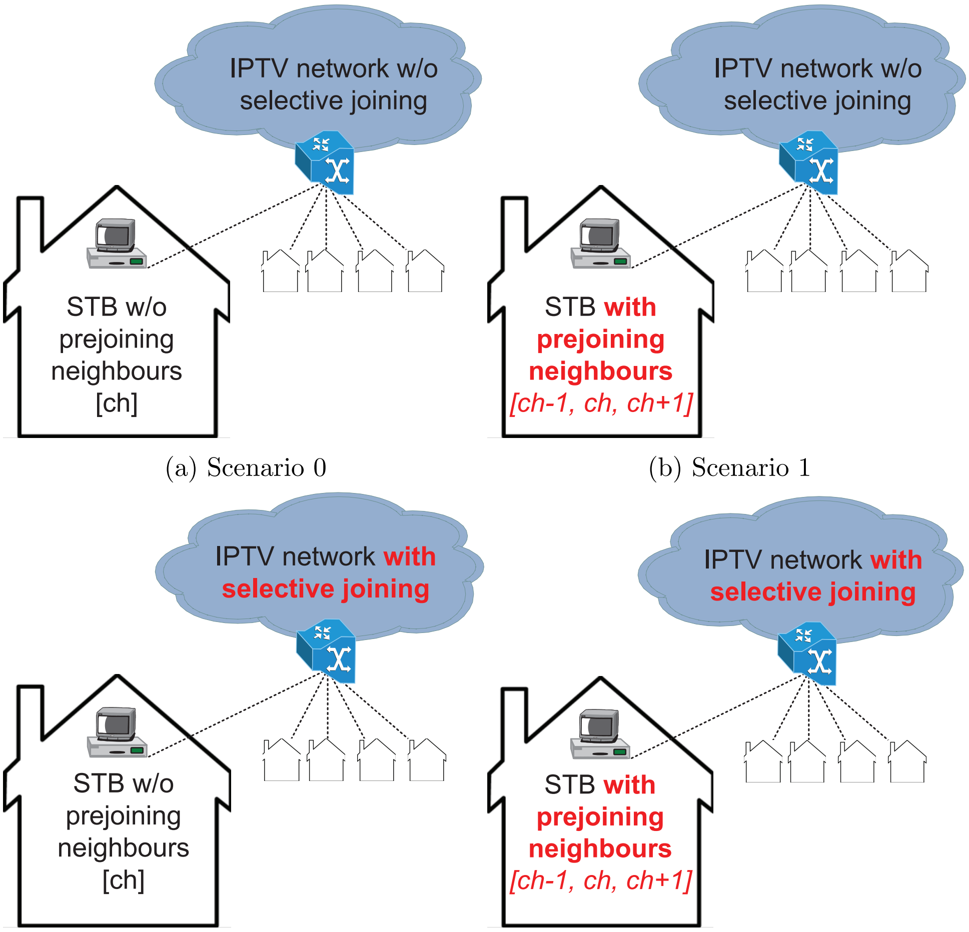
In the plots I also include, for the futuristic scenario 3kUHD, the situation where all routers are energy-proportional (EP) (3kUHD_ep). These nodes have the energy-proportional traffic profile presented in Figure 6.6. In this case, the energy saving advantage from using the proposed scheme is even more pronounced. Note that by looking at the EP model from Figure 6.6 one would expect the power savings to be huge when compared with current routers' power profiles. This is a fact, but at a first glance this does not look to be the case in the graphs just presented. The reason for this is the type of analysis I am making here. To make it absolutely clear, the blue lines show the improvement of using the proposed scheme considering the power profile of current routers over not using it (assuming that same profile of course). The green lines show the improvement of using this scheme assuming EP nodes over not using it (again, *assuming EP nodes*). Having EP nodes will *always* decrease the power significantly, even if the proposed scheme is not being used. That is a fact that can be easily inferred from Figure 6.6. But the important aspect to emphasise is that using the selective joining scheme leads to a higher relative gain considering that different starting point in the analysis (i.e., the use of EP routers).

6.6 Discussion: effect on channel change delay

The scheme proposed in this chapter represents a tradeoff between channel change delay and operational efficiency. As explained, if the user requests a channel that was not joined previously by a network node, the request has to go up towards the source to the nearest branch of the multicast tree. It will therefore experience a larger than usual delay, due to an increase in the network delay. This problem is mitigated by two factors. First, and as explained in Section 2.1.1, the network delay is a small contributor to the overall zapping delay in IPTV: buffering and stream synchronisation are the largest contributors to this delay. Second, if one of the main objectives of the proposed scheme is satisfied — namely, to affect a very small number of channel requests — the number of signalling messages transported in the network will not be significant, and hence overall network delay may not be seriously affected. And, as shown in this chapter, in particular in Figures 6.3 and 6.4, a significant increase in resource and associated energy efficiency is possible while affecting just a very small number of channel change requests.

Anyway, a small contradiction becomes clear. The conclusion of the previous chapter is that zapping delay may be removed through the addition of more channels, whereas the conclusion of the current chapter is that the removal of unused channels from the aggregate bundle provides an avenue for saving power. Clearly there is a trading relationship between the zapping delay and the need to save power. To make this relationship explicit I present a simple quantitative analyses in this section.

I consider the four scenarios depicted in figure 6.9. Scenario 0 represents the way IPTV networks operate today. As explained in Section 2.3, static IP multicast is used, with each DSLAM joining all TV multicast groups and thus receiving content from all TV channels. The DSLAM then distributes a single TV channel to each Set Top Box (STB). Scenario 1 is the proposal presented in Chapter 5: pre-joining some neighbouring channels to reduce channel switching delay. For the analysis I assume that two channels are pre-joined and that the STB does not leave these groups while the requested channel is being watched (according to the notation used in the previous chapter, $Neighbours = 2$ and concurrent channel time $T = always$). In this case, 55% of all switching requests will experience no delay (Figure 5.4). Scenario 2 is the proposal presented in the current chapter: selective joining to increase operational efficiency. The DSLAM does not join all TV multicast groups, but only a selection of the available channels. In this case, I assume that only the active channels are joined. Recall that fewer channels joined by the DSLAM represents a reduction in bandwidth and energy consumption, as analysed in this chapter. Also note that in the current chapter I have assumed thus far that only one TV channel is distributed to each STB at any one time (i.e., so far the pre-joining scheme from the previous chapter was not used jointly with the scheme presented in this chapter). Finally, in scenario 3 I consider the two proposals from the previous and the current chapter together.


IPTV network w/o selective joining
STB w/o prejoining neighbours [ch]
IPTV network w/o selective joining
STB with prejoining neighbours [ch-1, ch, ch+1]
IPTV network with selective joining
STB w/o prejoining neighbours [ch]
IPTV network with selective joining
STB with prejoining neighbours [ch-1, ch, ch+1]

(a) Scenario 0 (b) Scenario 1

(c) Scenario 2 (d) Scenario 3

Figure 6.9: The four scenarios considered

I assume the DSLAM is using the selective joining technique and the STB is pre-joining two neighbours along with the requested channel.

To quantify the trading relationship between the switching requests that experience no delay and the number of channels distributed to a DSLAM, I wrote a simple simulation experiment in C. The objective of the simulation is to quantify how many channels are active, in the DSLAM, on average, at any one time. Note that this simulation is only performed for scenarios 2 and 3. In the other two scenarios all TV channels are always distributed to the DSLAM so no simulation is necessary. The input to the simulation was the long term distribution of channel popularity I obtained empirically from the analysis of the dataset (Figure 4.4). I generate a random number based on this distribution for each STB in order to simulate what TV channel the user is watching, and therefore what channels the DSLAM is distributing to the STB. In scenario 2 only the channel the user is watching is distributed, while in scenario 3 several channels are distributed to the STB: the one the user is watching, the previous and the next[1]. This is performed for every STB covered by a single DSLAM. This way it is possible to quantify

[1] It is relevant to mention that I have information not only of the popularity of each channel but also on channel ordering.

how many channels are active, on average, in a single DSLAM. For these scenarios I consider that the DSLAM serves 409 households (i.e., STBs), which was the average number of STBs a DSLAM served in Telefonica's IPTV network at the time of data collection. I also consider that the network distributes 150 TV channels, which is this provider's service offering. I run the simulation 1000 times, and calculate the median, 90th and 10th percentile. The results are presented in Figure 6.10.

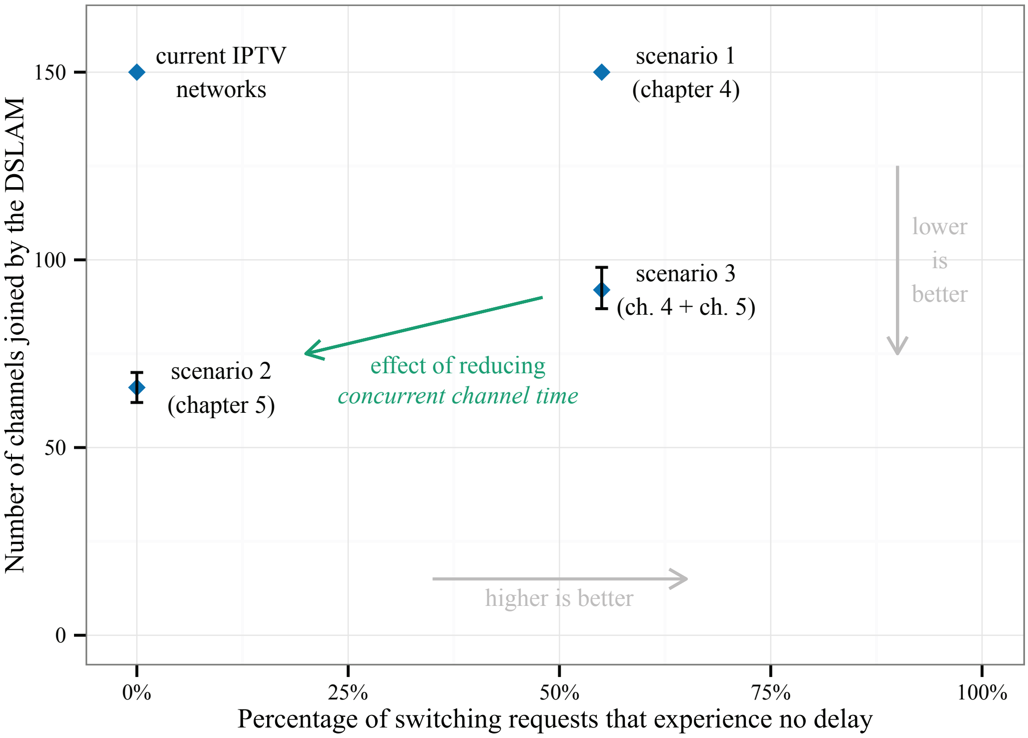


Figure 6.10: Trading relationship between the switching requests that experience no delay and the number of channels distributed to a DSLAM.

In scenario 0 all 150 TV channels are distributed to the DSLAM, and all switching requests experience the normal IPTV delay. Therefore, the number of switching requests that experience no delay is zero in the plot. In scenario 1, again, all TV channels are distributed to the DSLAM. However, as predictive pre-joining is used, 55% of all switching requests experience no delay (according to figure 5.4 and the assumptions made above, $Neighbours = 2$ and $T = always$). As shown before, with this scheme user experience is improved. In scenario 2 selective joining is used, so not all TV channels need to be distributed to the DSLAM. Fewer channels joined by the DSLAM represents bandwidth savings which are translated in energy savings, as reported in this chapter. As predictive pre-joining is not used, all switching requests experience the normal IPTV delay. Finally, the results from scenario 3 illustrate the compromise between operational efficiency (i.e., bandwidth and energy savings) and user quality of experience. By using the two schemes proposed in chapters 5 and 6 it is possible not only to improve user experience but also to increase network efficiency (although less significantly than in scenario 2). The reason is twofold.

As more channels are distributed to each STB (the requested channel plus the neighbours), the number of active channels that need to be joined by the DSLAM naturally increases, when compared to scenario 2. However, this increase is not very significant. The reason may reside in the neighbouring channels of popular channels also being popular. That being the case, there is a good probability that these channels are already amongst the active channels joined by the DSLAM. If the period the neighbouring channels are sent concurrently with the requested channel is finite (i.e., if $T \neq always$), then the percentage of switching requests that experience no delay is reduced (again, I refer the reader to Figure 5.4). As in this case fewer channels will be distributed from the DSLAM to the households one also expects the number of TV channels joined by the DSLAM to decrease. This situation is depicted as the green arrow in the figure (note that this decrease is not necessarily linear, as the arrow suggests).

6.7 Conclusions

Delivering TV streams in an IP network consumes a significant amount of resources. As the number of TV channels increases and the quality of the streams improves (with the resulting increase of its bandwidth requisites), resource and energy efficiency will increasingly become a concern. IPTV service providers will therefore need to reconsider their IPTV distribution networks. Fortunately, the majority of users tend to enjoy the same TV channels: 90% of all TV viewing is restricted to a small selection of channels [42, 161]. In this chapter I showed that IPTV providers should take advantage of this fact. Instead of multicasting all TV channels continuously everywhere, IPTV networks should judiciously choose *which* TV channels to distribute *where*, at any one time. In other words, they should move from their static multicast distribution schemes.

I call the method I have proposed to achieve this goal, selective joining. Contrary to static multicast solutions, where network nodes join all TV multicast groups, in this scheme the nodes join only a selection of channels. Namely, the active TV channels (those for which there is at least one viewer connected to that node) plus a small subset of the inactive ones (those for which that particular node has no viewers). I evaluated selective joining by performing a trace-driven analysis using the dataset described in Chapter 4. I contrasted the bandwidth savings achieved with the number of requests affected, and concluded that a tradeoff is possible. Bandwidth can be reduced significantly by distributing less TV channels in the network, without compromising service quality, i.e., affecting only a very small percentage of channel switching requests.

A power consumption model was also developed to assess how these bandwidth savings translate into energy savings. The main conclusions were that despite nowadays energy savings not being significant, in a plausible medium term scenario the energy advantage of using such dynamic multicast distribution scheme becomes evident. And as network equipment evolves to having more energy-proportional power consumption profiles, using the selective joining scheme increases its relative advantage further when compared with static multicast.

The first technique I proposed in this dissertation to increase the resource and energy efficiency of an IPTV network was based on a simple paradigm: “avoid waste!” [167]. The technique

I present in the next chapter is based on a different paradigm: the introduction of energy-efficient optical switching technologies in these distribution networks.

Chapter 7

Optical bypass of popular TV channels

In the previous chapter I proposed a technique to increase the resource and energy efficiency of IPTV distribution networks. This technique was based on a simple paradigm: avoiding waste. The technique I propose in this chapter is based on a different paradigm: introducing optical switching in the network. The rationale for this proposal is the fact that optical switching techniques are more energy-efficient than their electronic counterpart. In particular, I assess the opportunities for performing *optical bypass* in IPTV networks. With optical bypass, traffic not destined for a given network node is not processed electronically by that node. This traffic is all-optically switched, i.e., it is switched at the optical layer and is therefore not processed by the IP layer. By avoiding electronic processing and performing optical switching instead, energy savings are to be expected.

In this chapter I propose a novel *energy* and *resource* friendly protocol for core optical IPTV networks. The fundamental concept is to blend electronic routing — switching at the IP layer — and optical switching — switching at the optical layer. The objective is to glue the low-power consumption advantage of circuit-switched all-optical nodes with the superior bandwidth-efficiency of packet-switched IP networks. The former assures the energy-friendliness of the scheme, whereas the latter guarantees its resource-friendliness. The main idea is to optically switch popular TV channels while still processing electronically the unpopular ones. Popular TV channels are watched by many, having viewers everywhere in the network, at any time. Even considering a semi-dynamic multicast network, as proposed in the previous chapter, these channels have to be distributed continuously everywhere. This type of long-lived flow is the perfect target to optically bypass the core network nodes. The unpopular channels have less viewers, and hence do not need to be distributed continuously everywhere. For bandwidth efficiency reasons, these channels are switched at the IP layer, to allow their quick removal from or insertion to the multicast network as needed.

By analysing the dataset described in Chapter 4, I assess the opportunities for optical bypass when using the proposed protocol in a real IPTV network. I observe that 50% of the TV traffic can be optically switched at the network core. Additionally, I demonstrate that the protocol does not impose significant control overhead to the network. As its update interval can be long, it is possible to guarantee a low overhead without compromising performance.

As the main objective of the proposed scheme is to reduce energy consumption, I analyse its impact in this respect. The main conclusion is that with the introduction of optical bypass the energy advantage increases further and quite significantly when compared with the scheme proposed in the previous chapter.

7.1 Introduction

To guarantee the quality of experience its users demand, current IPTV networks distribute all TV channels continuously everywhere. I have shown in Chapter 6 that this is a wasteful use of network resources and that it has an impact in energy consumption. Fortunately, a majority of users enjoy the same TV channels, allowing the distribution of only a selected set of TV channels without impacting the expected quality of experience significantly. Energy can therefore be saved by avoiding waste, as I have demonstrated before.

With the goal of reducing energy consumption even further, in this chapter I consider the introduction of energy-friendly optical switching techniques in the core of optical IPTV networks. While in the previous chapter I considered only the IP layer, being agnostic to the layers below, in this chapter I consider the particular case of optical IP networks, and therefore also the optical layer. In legacy (or first generation) core optical networks, optics was essentially used for transmission and simply to provide capacity [162]. All the switching and other intelligent network functions were handled by electronics. These networks were therefore optical-electrical-optical (OEO) based, with all traffic routed to a node being converted to the electric domain, regardless of weather or not the traffic was destined for that node [175]. The second generation optical networks include routing and switching at the optical layer [162]. The most significant development of this new type of networks is the advent of optical bypass, where traffic transiting a node can remain in the optical domain, instead of performing energy-costly OEO conversions [175].

With the introduction of optical bypass capabilities in the IP network, traffic not destined for a given IP router is placed onto a WDM wavelength that is not processed by that router. Instead, this traffic is all-optically switched. The use of this technique allows some work to shift from electronic routers to optical switches, which is seen as an important strategy for managing the growth of network power consumption in the future [14, 180, 220]. Besides reducing electronic processing in routers, the potential for energy savings arises from the switching energy required by an all optical cross connect being orders of magnitude below that of electronic routers [14, 205]. Due to the circuit-switching nature of optical networks [162], however, only *long-lived flows* can be considered realistic targets for optical bypass. Conveniently, some IPTV traffic is in this category. Some TV channels are very popular, having viewers everywhere in the network, at any particular time. Optically switching such long-lived flows can therefore be advantageous energy-wise. Other less popular and niche channels have periods without any viewers in particular locations, so it is wasteful to distribute them continuously everywhere. The dynamic nature of electronic packet-switching nodes is therefore ideal to switch this type of traffic. This guarantees the network is bandwidth efficient, by allowing these TV channels to be quickly removed from or added to the network as needed.

Considering the above, in this chapter I propose a *hybrid* protocol to be used in the core of IPTV distribution networks, blending electronic routing with all-optical switching. Its "hybrid" nature comes from the assumption that the network core is composed of hybrid nodes, each including a WDM optical cross connect (OXC) and a multicast-enabled IP router, as illustrated in Figure 7.1. The inclusion of the OXC between the input ports and the router allows optical bypass to be performed. The main idea of the scheme is for popular TV channels to be all-optically switched (switched at the optical layer), while the rest are electronically routed (switched at the IP layer). The network distributes the two different groups of channels in two (disjoint) sets of wavelengths. The wavelengths from one set optically bypass the nodes, whereas the other wavelengths are sent to the routers for processing.

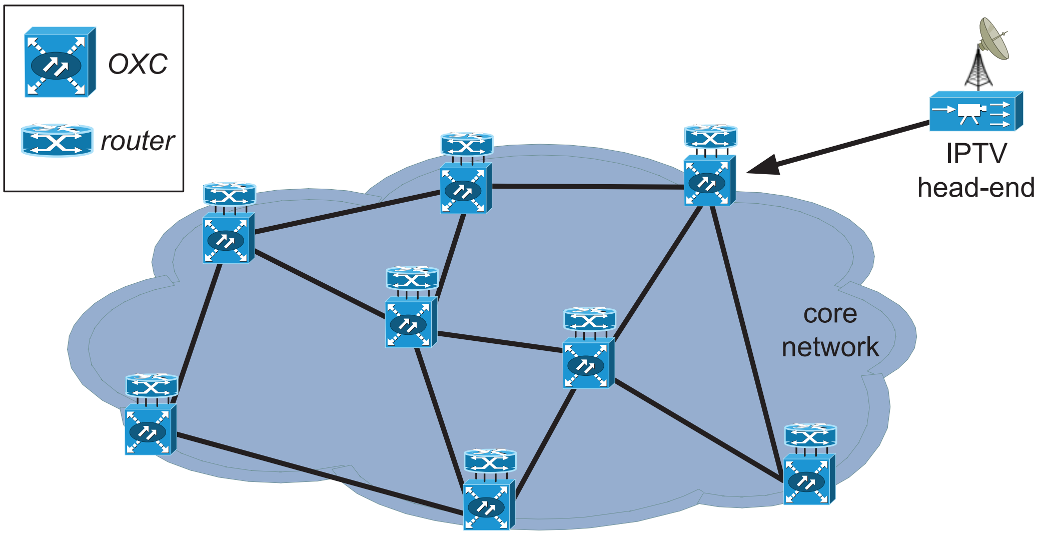


Figure 7.1: Core network topology considered

I evaluate the proposed protocol by means of a trace-driven analysis of the dataset described in Chapter 4. First, I demonstrate that the protocol is scalable as its update interval can be long. This implies that the control overhead is small. Such result was expected as channel popularity is relatively stable over short time frames [161]. Afterwards, I show that half of all TV traffic can be optically switched in the network core without decreasing bandwidth efficiency.

To understand the impact of this protocol on energy consumption I develop a power consumption model for the hybrid node considered. The power consumption model for the router is the one based on real measurements [184] used in the previous chapter. The model for the optical layer components is based on specifications from manufacturers and on real measurements. The main conclusion of the analysis is that the introduction of optical bypass further increases the energy savings already achieved by using the selective joining scheme proposed in the previous chapter. At normal traffic loads (less than 30%), the power savings considering current and futuristic scenarios jump from 10% to 15% using selective joining only to more than 40% if one considers optical bypass.

But it is possible to increase energy efficiency even further. Considering the baseline traffic above, distributing all TV channels optically (i.e., with all IPTV traffic optically bypassing the core nodes), instead of only the popular channels, would increase power savings to around 60%. However, this comes with an increase of bandwidth inefficiencies, as all channels have to be distributed. By using the hybrid scheme I propose in this chapter bandwidth can be reduced by around 20%. The growing importance of niche channels and the expected increase in the quantity and quality of TV channels already discussed in the previous chapter argue in favour of such hybrid *resource* and *energy* efficient schemes.

The rest of this chapter is organised as follows. In Section 7.2 I explain how optical bypass can be used to reduce the energy footprint of IP networks. Then I describe the protocol proposed in this chapter in Section 7.3. I detail the methodology used in the analysis in Section 7.4, and evaluate the use of this protocol in Section 7.5. I analyse its impact on energy consumption in Section 7.6, discuss the advantages of such hybrid scheme when compared to pure all optical distribution in Section 7.7, and close this chapter in Section 7.8.

7.2 The use of optical bypass to save energy

An optical IP network can be seen as being made up of two layers, the IP layer and the optical layer [85]. This is shown in Figure 7.2. In the IP layer, a core IP router connects to an optical switching node (an Optical Cross Connect, OXC, in the figure) via short-reach interfaces and aggregates data traffic from low-end access routers. The optical layer provides capacity for the communication between IP routers. The OXCs are interconnected with optical fibre links, each usually containing several wavelengths using WDM technology. Associated with each fibre, a pair of wavelength MUX/DEMUX are deployed to multiplex and demultiplex wavelengths. Associated with each wavelength, one transponder is connected to the router to perform OEO conversions and transmit the data [180].

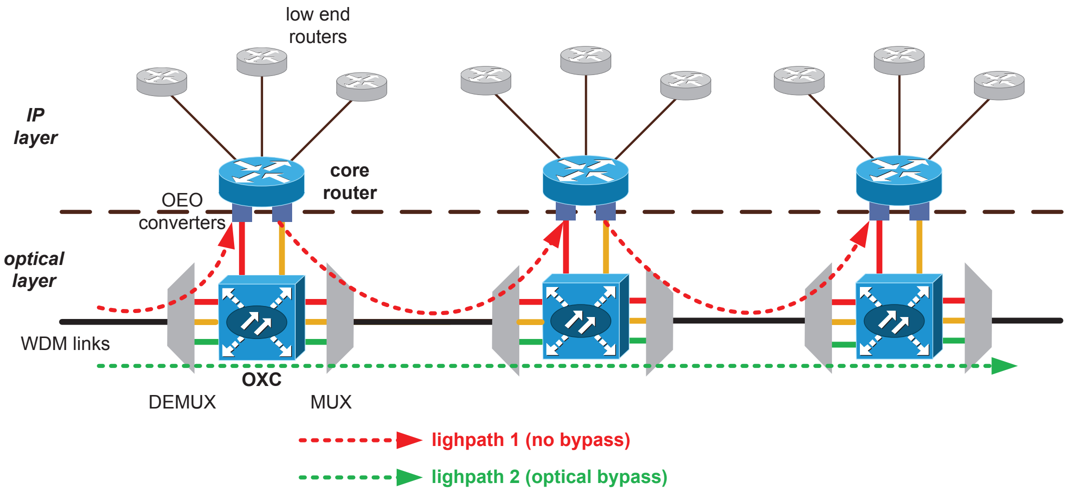


Figure 7.2: Optical network employing optical bypass techniques

In the first generation of optical networks, all the lightpaths[1] incident to a node had to be terminated, i.e., all the data carried by the lightpaths would be processed and forwarded by IP routers. This is represented in the figure by lightpath 1. The red wavelength is OEO converted at each node. In contrast, the new generation of optical networks includes elements such as the OXCs which allow some lightpaths to bypass the node. This approach allows IP traffic whose destination is not the intermediate node to directly bypass the intermediate router via a cut-through lightpath. This is represented by lightpath 2. The green wavelength bypasses all nodes.

Several researchers have pointed out recently that optical bypass technology is one important method to reduce the power consumption of IP networks [14, 220]. Shen and Tucher [180], Hou *et al.* [104] and others have indeed proposed using optical bypass to reduce the energy consumption of IP over WDM networks. This technique can save energy because it can reduce the total number of active IP router ports, and these play a major role in the total energy consumption of an optical IP network [180]. Introducing optical bypass results in the possibility of some ports and even whole linecards being turned off. Also, as the OEO converters consume a significant amount of energy, by reducing their use the node's energy footprint is further reduced. Shifting traffic from power hungry routers to low power optical switches by means of optical bypass is therefore an effective technique to save energy in optical networks.

7.3 Protocol for optical bypass in IPTV

In modern IP networks most packets transit multiple core routers [142]. These packets are fully processed at each intermediate node, with its headers inspected and forwarding lookup being performed. These are very energy-consuming tasks [15]. I propose in this chapter some of the core network's IPTV traffic to optically bypass the routers, thus reducing such electronic processing. As I already mentioned in this dissertation, some TV channels are very popular [42, 161] having viewers everywhere in the network. My proposal is to distribute these popular TV channels in a specific set of wavelengths that are all-optically switched in the intermediate nodes. The rest of the TV channels are distributed on a disjoint set of wavelengths that is sent to the routers for electronic processing. As these channels are distributed only upon request, in a particular moment a TV channel without viewers is not distributed. I showed in the previous chapter that not distributing channels without viewers in the network core (i.e., by using the scheme evaluated in section 6.4 with `inactive_set_size` set to zero) does not compromise user experience significantly.

I assume each core network node is a hybrid node as in Figure 7.1. Each node includes a multicast-capable optical cross connect (OXC) where optical bypass can be performed, and a multicast-enabled IP router. I further assume these nodes are GMPLS-capable. As explained in Chapter 3 (Section 3.4), a unified control plane such as GMPLS allows the integration of optical circuit-switching techniques with electronic packet-switching.

[1] Recall from Chapter 3 (Section 3.4) that a lightpath is an optical point-to-point connection from a source to a destination.

Algorithm 1 Processing at the IPTV source

```
while true do
    sleep($\Delta\tau$)
    send_to_core-reg_nodes(ACTIVE_CHANNELS_REQUEST)
    {Wait until all requests are received...}
    $CPop \leftarrow$ ALL_TV_CHANNELS
    $CNonPop \leftarrow \emptyset$
    for $i = 1$ to NUMBER_OF_NODES do
        $CPop \leftarrow CPop \cap ActiveCh[i]$
    end for
    {$CPop$ now includes all popular TV channels}
    for $i = 1$ to NUMBER_OF_NODES do
        $CNonPop \leftarrow CNonPop \cup (ActiveCh[i] \notin CPop)$
    end for
    {$CNonPop$ now includes the other TV channels with viewers}
    $\lambda_o \leftarrow$ [Wavelengths filled with $CPop$ channels]
    $\lambda_e \leftarrow$ [Wavelengths filled with $CNonPop$ channels]
    send_to_all_nodes(SWITCHING_CHANGE_REQUEST, $\lambda_o$, $\lambda_e$)
end while
```

Algorithm 2 Processing at each core-regional node

```
while true do
    MESSAGE = msg_rcv_from_source()
    if MESSAGE == ACTIVE_CHANNELS_REQUEST then
        $ActiveCh \leftarrow$ get($McastFwdTable$)
        send_to_source($ActiveCh$)
    end if
end while
```

Algorithm 3 Processing at each core node

```
while true do
    MESSAGE = msg_rcv_from_source()
    if MESSAGE == SWITCHING_CHANGE_REQUEST then
        for all $\lambda \in \lambda_o$ do
            switch_optically($\lambda$)
        end for
        {Wavelengths in the set $\lambda_o$ are optically bypassed}
        for all $\lambda \in \lambda_e$ do
            route_electronically($\lambda$)
        end for
        {Wavelengths in the set $\lambda_e$ are sent to the router}
    end if
end while
```

The protocol for optical bypass in IPTV networks proposed here consists of three algorithms. Algorithm 1 runs at the IPTV source, algorithm 2 runs at core-regional nodes (I refer the reader to the reference architecture in Figure 2.5), and algorithm 3 runs at the core nodes (including core-regional ones). The details of the proposed protocol follows:

1. After a specified time interval, $\Delta\tau$, the source transmits a message requesting all hybrid core-regional nodes to submit their active channels (algorithm 1, lines 2-3). Recall that an *active channel* is a channel for which there is at least one viewer. This message sent by the source serves as a trigger for all core-regional routers to send this information back to the source as soon as possible. Considering that all nodes are GMPLS-capable, this information can be sent as an RSVP-TE Notify message, for example. RSVP-TE Notify messages were added to RSVP-TE[1] to provide general event notification to nonadjacent nodes [154].

2. Each regional-core node then sends information on its *active channels* to the IPTV source. As the active channels are those being distributed by the regional-core router to its region, the multicast forwarding table of this router contains a line with their multicast group addresses and the interfaces used to forward packets to[2]. The information requested can thus be easily retrieved and sent back to the source (algorithm 2, lines 3-6). Again, an RSVP-TE Notify message can be used for this purpose.

3. Once the source receives these sets from all routers, it checks which TV channels should be optically switched (the popular ones), and which should be electronically routed (the remainder channels with viewers). The popular channels are those which have viewers everywhere. Their multicast group addresses are present in the multicast forwarding tables of every core-regional router. The intersection of all sets received by the source thus results in a new set with the list of popular channels[3] (algorithm 1, lines 6-8). The union of the active channels of each set which are not popular results in a new set with the non-popular TV channels (algorithm 1, lines 9-11).

4. The TV channels are distributed, from the source, in two distinct sets of wavelengths: λ_o and λ_e. The popular channels are distributed using N different wavelengths: $\lambda_o = N \times \lambda$. The others are sent in a disjoint set of M different wavelengths: $\lambda_e = M \times \lambda$. The number of wavelengths in each set depends on the number of TV channels and its bit rate, and on the capacity of each wavelength. The IPTV source decides the composition of each set of wavelengths and informs all core nodes of its decision (algorithm 1, lines 12-14). This information can be sent in the form of an RSVP-TE PATH message. This is one of the

[1] As its name implies, the Resource Reservation Protocol - Traffic Engineering (RSVP-TE) is an extension of the resource reservation protocol (RSVP) for traffic engineering, and is used as part of the GMPLS control plane for this purpose.

[2] Note that the multicast state of all active channels is maintained in the forwarding table of the core-regional router, including those channels that are being all-optically switched.

[3] I am abusing the term “popular” in this chapter. If one TV channel has a single viewer in each region then it is included in the popular set. I use this term to ease the understanding of the scheme.

messages used to allocate resources in the network. In multicast scenarios, only one PATH message needs to be sent to multiple receivers, thus conserving network bandwidth.

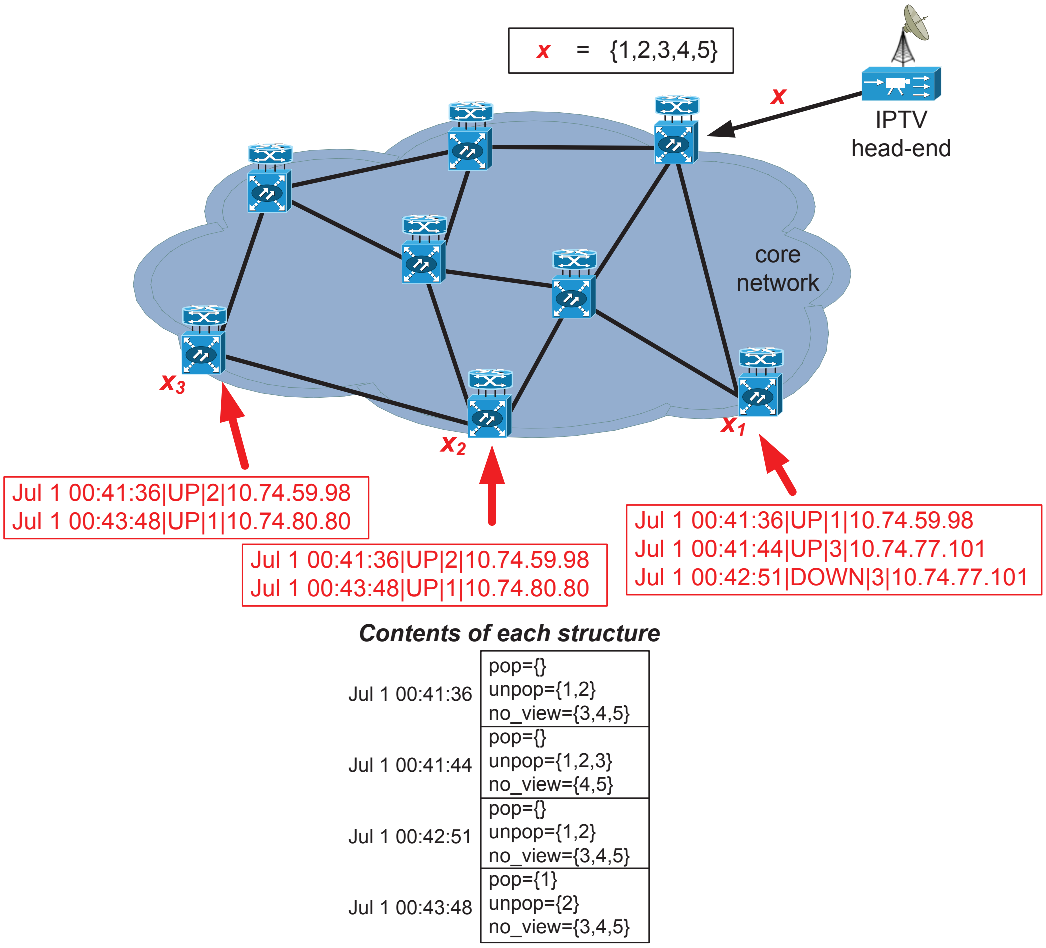
5. Each core node then sets up its switching state to optically switch the λ_o group (these wavelengths will therefore optically bypass the routers), and electronically route the λ_e group (algorithm 3, lines 3-10).

7.4 Methodology

The scheme proposed in this chapter is evaluated by means of a trace-driven analysis. The IPTV trace detailed in Chapter 4 is used as input to the analysis performed. As I mentioned before, I restrict the analysis of the proposed scheme to the optical network core, as this is the only location where it is realistic to assume the presence of OXC equipment in the medium-term. Recall that in Chapter 6 I parsed the IPTV trace data with the objective of creating a single time-ordered trace file that includes all switching events sent to all DSLAMs in a *specific region*. This allows the evaluation of this scheme at the core-regional router level, which is my intention here.

To evaluate the proposed scheme I developed a Python script that checks each line of the input trace, to obtain each switching event that occurs in that specific region. In a similar manner to what was done in the previous chapter, for every switching event I record the set of active channels, `active_channels`. For each core-regional router I maintain one such structure. The intersection of all sets, at time t, is the set of popular channels, `pop`, at time t. These channels have at least one viewer per region. The reunion of all channels that, at time t, are at least in one `active_channels` set but are not in the `pop` set is the set of non-popular channels, `unpop`, at time t. These channels have at least one viewer in the network, but there are regions where they have no viewers. The channels that are not in the `pop` nor in the `unpop` sets are included in the `no_view` set. These channels have no viewers anywhere in the network. By running this script I am thus able to know, with the precision of one second for the trace duration (recall that the trace has one second precision), the number of channels with users everywhere (popular), somewhere (unpopular), and nowhere (no viewers).

Figure 7.3 illustrates the proposed methodology with a simple example. I assume the network distributes only five channels, numbered from 1 to 5. At 12:41:36am an `UP` message for channel 1 is received in node x_1's region from the STB with IP address `10.74.59.98`. In regions x_2 and x_3 `UP` messages for channel 2 are sent at exactly the same time. These two channels are hence included in the `unpop` set while the other three channels remain in the `no_view` set. The set `pop` remains empty as there are no channels with viewers everywhere. Nine seconds later node x_1 receives a join message to channel 3. This channel is included in the `unpop` set. Around one minute later it is again removed from this set, and included in the `no_view` set, after a `down` message is sent to the same node. Finally, at 12:43:48am two UP messages are sent to channel 1 in regions x_2 and x_3. As this channel is now active in every region, it is removed from the `unpop` set and included in the set `pop`.



Jul 1 00:41:36|UP|2|10.74.59.98
Jul 1 00:43:48|UP|1|10.74.80.80

Jul 1 00:41:36|UP|2|10.74.59.98
Jul 1 00:43:48|UP|1|10.74.80.80

Jul 1 00:41:36|UP|1|10.74.59.98
Jul 1 00:41:44|UP|3|10.74.77.101
Jul 1 00:42:51|DOWN|3|10.74.77.101

Contents of each structure

Jul 1 00:41:36	pop={}
	unpop={1,2}
	no_view={3,4,5}

Jul 1 00:41:44	pop={}
	unpop={1,2,3}
	no_view={4,5}

Jul 1 00:42:51	pop={}
	unpop={1,2}
	no_view={3,4,5}

Jul 1 00:43:48	pop={1}
	unpop={2}
	no_view={3,4,5}

Figure 7.3: Proposed methodology

7.5 Evaluation

As explained before, the proposed protocol is evaluated by performing a trace-driven analysis on the IPTV dataset. All results I present in this chapter arise from the analysis of the whole data set (6 months, 255 thousand users). The evaluation is threefold. First, I investigate the scalability of the protocol. Second, I analyse the opportunities for optical bypass when running the proposed protocol in the network under study. Finally, in the next section I analyse the impact the use of this protocol has in power consumption of the IPTV network.

7.5.1 Scalability

For a network protocol to be scalable it is important that it does not impose a significant processing overhead to the network nodes and that it does not add a great amount of signalling traffic to the network. By guaranteeing a relatively long update interval for the control information (the $\Delta\tau$ variable in the proposed protocol) it is possible to guarantee a low overhead to the nodes and to the network as a whole. On the other hand, to assure the best performance it is important that the network state[1] is consistent with network usage (in this particular case, it should reflect channel popularity). Having a short update interval marries with this objective.

It is known that channel popularity is relatively stable over short time frames, and that it becomes more dynamic when longer time frames are considered [161]. Regular updates may therefore not be needed. To attest this, I analyse the henceforth called *TV channel churn rate* in the 11 core-regional nodes of this network. I compare the active TV channels at time τ with the active channels at time $\tau + \Delta\tau$, for different values of $\Delta\tau$. The number of channels that are different between the two sets in two consecutive periods is the TV channel churn rate. The results are shown in Figure 7.4, for each region, and for five values of $\Delta\tau$. The median of the channel churn rate over the whole period of the trace (6 months) is presented, with the lower and upper error bars representing the 5th- and 95th-percentile, respectively.

By analysing the results in Figure 7.4, I conclude that the churn rate is usually quite low, particularly for values of $\Delta\tau$ below 1 hour. A long update interval of 15 minutes, for instance, is a good compromise. It does not represents a significant overhead to the network, while at the same time guarantees that the network state changes with channel popularity dynamics.

7.5.2 Opportunities for optical bypass

The protocol proposed in this chapter divides the TV channels into three groups: the popular channels, the unpopular channels, and the channels without viewers. The channels from the former group optically bypass the routers. Those from the second group are sent for the router for electronic processing. Finally, those from the latter group are not distributed by the IPTV source. To understand the opportunities for optical bypass in the core of the IPTV network, I need to quantify how many channels would be included in each group at regular intervals. For this purpose, I retrieve the number of channels in each set `pop`, `unpop`, and `no_view` periodically, for the whole trace. I consider for the analysis an update interval equal to 15 minutes, for the reasons explained above. This is the periodicity with which I retrieve the number of channels in each set. In Figure 7.5 I present the results obtained (median, 5th-, and 95th-percentile) from the analysis of the whole dataset.

I start the analysis from the bottom. On average, one fifth of the TV channels do not need to be distributed by the IPTV source. This is the reason why I used this number for the analysis made in Section 6.5.2. Recall from that section that not distributing this traffic has a negligible impact on the service (Figure 6.4). The remaining 80% TV channels are distributed to the network core. Around 50% of the TV channels can be optically bypassed. This means that, on average, at any one time, half of the channels have at least one viewer in each region. The number of channels requiring electronic processing can thus be reduced to around 30%. In the next section I investigate the impact this has on energy consumption.

[1] In this context, the network state consists of the wavelength switching configuration at each node, and the set of TV channels transported in each wavelength group, λ_o and λ_e.

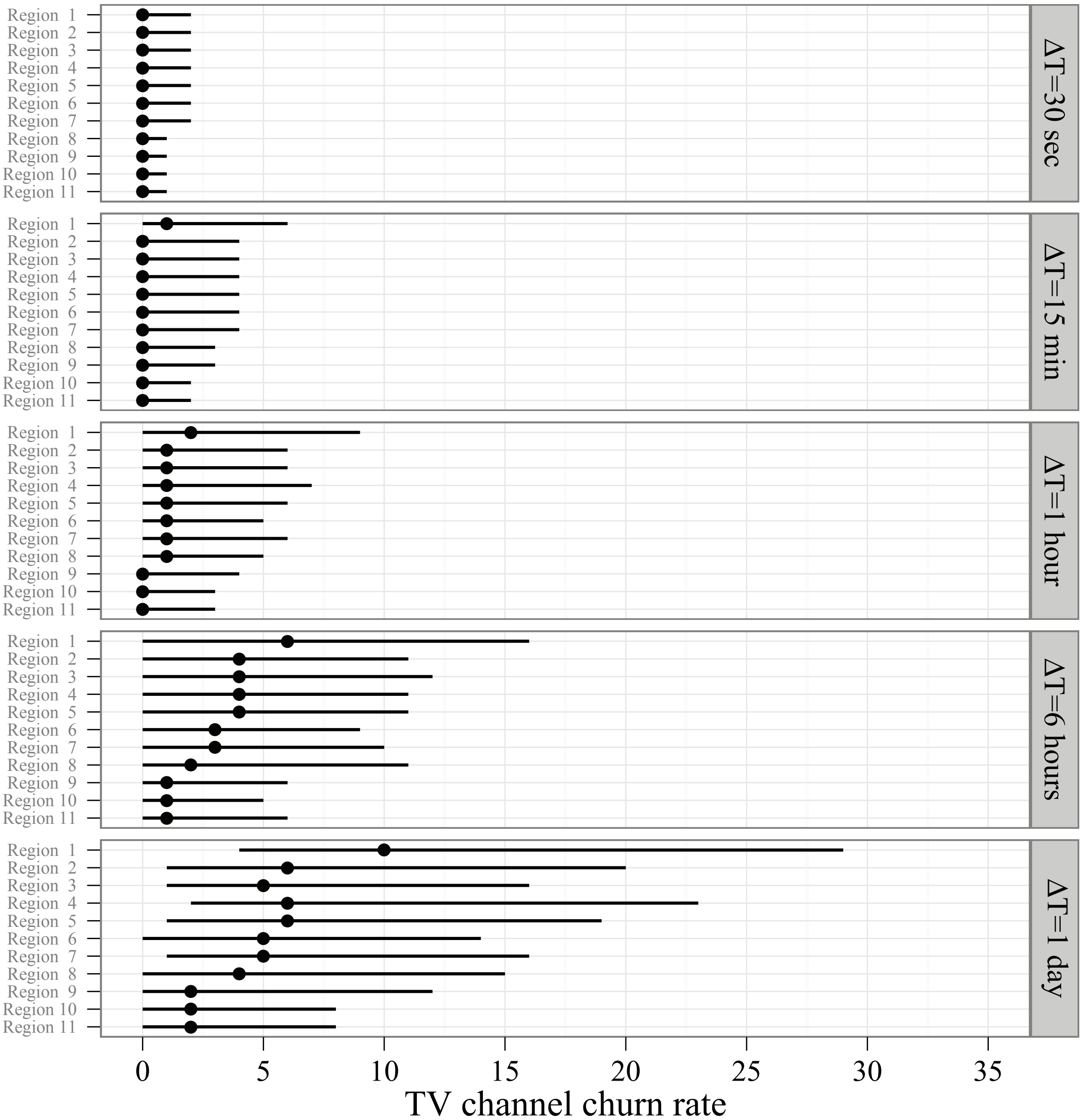


Figure 7.4: TV channel churn rate for all eleven regions, for five values of the update interval

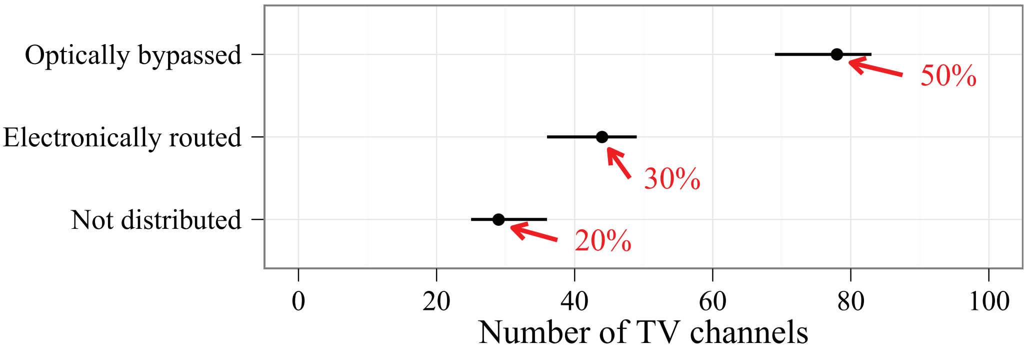


Figure 7.5: Average number of TV channels that are optically bypassed, electronically routed and not distributed, respectively

7.6 Impact on energy consumption

After understanding that by using the proposed protocol there are clear opportunities to introduce optical bypass in IPTV networks I now analyse the impact this has on energy consumption. By employing this technique energy savings are expected for two reasons:

1. Some traffic flows (the popular TV channels) bypass some routers. This reduces the number of bits requiring electronic processing, thus avoiding energy-expensive OEO conversions, buffering, and forwarding table lookups. The work is shifted to optical switches, which are at least two orders of magnitude more energy efficient when compared to its electronic equivalent [14].

2. As TV channels without viewers are not distributed, network load is reduced and even less bits require electronic processing in the routers.

7.6.1 Selective joining in core optical networks

Before presenting results from using the proposed scheme, it is important to return to the scheme proposed in the previous chapter, selective joining, considering an optical network scenario. In Chapter 6 I focused the analysis on power consumption of IP routers only. As now I consider a core optical network, I also need to include the power consumption of the optical layer components. This will allow a fair comparison with the proposal made in this chapter.

In optical networks, associated with each wavelength (port) is a transponder (OEO converter), as was shown in Figure 7.2. The transponder interfaces the router to a fibre optic cable. Its main function is to perform the required OEO conversions. Considering this, the power consumption model presented as Equation 6.1 is now updated, as in Equation 7.1.

$$P = P_{ch} + K_T P_T + \sum_{i=0}^{L} P_{l_i} \tag{7.1}$$

As can be seen, the only difference from equation 6.1 is the inclusion of the power consumption of the transponders. In this equation, K_T is the number of transponders (one per port) and P_T is the power per transponder. Every time a new port needs to be turned on, a new transponder is also activated. I assume the power consumption for each transponder to be 73 W, based on Alcatel-Lucent WaveStar OLS 1.6T ultra-long-haul systems [6]. This figure has been used in recent related work [104, 180].

In accordance to the results presented in the previous section (Figure 7.5), I assume that only 20% of the channels are not distributed to the core. The results I present in Figure 7.6 thus correspond to a reduction of IPTV traffic in the network core to 80%.

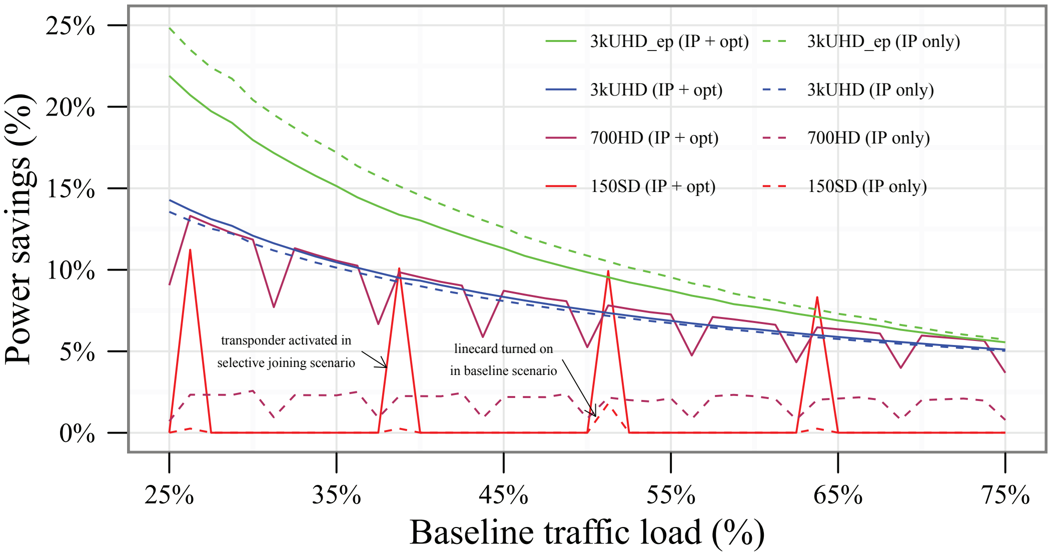


Figure 7.6: Power savings of using the selective joining scheme considering an optical IP network, as a function of the baseline traffic load in the node (core network)

As can be seen by comparing this plot with the one presented in the previous chapter, in Figure 6.8[1], the results change significantly. The main reason is the fact that the transponders are power hungry equipment. This results in an increased advantage in using the selective joining scheme in some scenarios, as reducing traffic load decreases the number of active transponders. It is particularly relevant to mention scenario `700HD`, which is typical in current networks (recall that this scenario is based on AT&T's IPTV service offering [160]). The use of the scheme proposed in the previous chapter increases the power savings to around 10% in normal traffic conditions. Similarly to Figures 6.7 and 6.8 in Chapter 6, the peaks in Figure 7.6, evident in both the `150SD` and `700HD` scenarios in this case, represent transition points. In the previous chapter, the most pronounced bumps represented new linecards being turned on in the baseline scenario. While the traffic load did not increase over those transition points the proposed scheme presented a higher-than-average power saving advantage. The same occurs in Figure 7.6. However, in this case the peaks represent the addition of another active transponder in the baseline scenario. As this component consumes more power than a linecard, the peaks are more pronounced when compared with those from the previous chapter. They also occur more frequently because an active transponder is activated every time a new port is turned on.

[1] The dashed lines in Figure 7.6 are the same as the dashed lines in Figure 6.8.

7.6.2 Energy consumption model of the hybrid nodes

To be able to quantify the energy savings achieved by introducing optical bypass in an optical IPTV network, in this section I build a power consumption model of the hybrid node considered in this chapter. Such node is depicted in Figure 7.7.

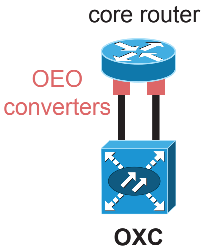


Figure 7.7: Hybrid node

Three factors affect the power consumption of an hybrid node:

1. The power consumption of the router.
2. The power consumption of the OXC.
3. The power consumption of the OEO converters (transponders).

Note that in this analysis I do not consider the power consumption of other optical equipment that is necessary in an optical network, such as the optical amplifiers, multiplexers and demultiplexers. I consider switching equipment and OEO converters only. Previous work [180, 220] as shown that switching equipment and transponders (OEO converters) are the main contributors for power consumption of optical IP networks (responsible for over 97% of total power consumption according to [180]). Based on the three variables above, I use the following model for the power consumption P of a hybrid node:

$$P = P_R + P_{OXC} + P_{OEO} \tag{7.2}$$

In equation 7.2 P_R is the power consumption of the router, P_{OXC} is the power consumption of the optical cross connect, and P_{OEO} is the power consumption of the OEO converters (transponders). For P_R I use the model based on real measurements [184] developed in Chapter 6 (Section 6.5.1). The power consumption of the OXC is given by equation 7.3.

$$P_{OXC} = K_{op} P_{op} \tag{7.3}$$

In this equation, K_{op} is the number of input/output optical switch ports and P_{op} is the power per input/output switch port. I assume the OXC switching fabric is realised using micro-electro-mechanical systems (MEMS) [76]. In a MEMS optical switch, a micro-mirror is used to reflect a light beam. The direction in which the light beam is reflected can be changed by rotating the mirror to different angles, allowing the input light to be connected to any output port. These MEMS have switching times of the order of milliseconds or hundreds of microseconds and for this reason can be used only for slow switching (i.e., circuit switching). For faster switching Semiconductor Optical Amplifiers (SOAs) could be used. But as MEMs consume less power [7], and as the OXC is not to be used for fast switching, MEMS are the option here. I assume 3D-MEMS [215] in particular. The power per input/output switch port of the OXC corresponds to the power consumption for its continuous control, which is equal to 107 mW per input/output port. This value is based on the power consumption of the MEMS controller circuitry of an 80 × 80 3D-MEMS switch implementation, reported in [215]. I am therefore assuming power consumption is proportional to the number of active input/output ports[1]. The experimental figure and this assumption were considered in previous related work [7, 76] and are also in agreement with studies from other researchers [14, 193].

Finally, the power consumption of the OEO converters is given by equation 7.4.

$$P_{OEO} = K_T P_T \tag{7.4}$$

In this equation, K_T is the number of transponders (one per wavelength that connects to the router) and P_T is the power per transponder. As in the previous subsection, I assume the power consumption for each transponder to be equal to 73 W.

7.6.3 Results

I now analyse how the introduction of optical bypass techniques in the IPTV network translate into energy savings. I consider the same three scenarios as in Chapter 6: `150SD`, an IPTV service offering of 150 SDTV channels; `700HD`, 700 HDTV channels; and `3kUHD`, 3000 UHDTV channels. For the router model I make the same assumptions as in Chapter 6. For the first scenario, I assume a router with 4 linecards with 4x1Gbps Ethernet ports each, as in Figure 6.6. For the other scenarios I just scale up the model by increasing the number and changing the type of linecards. This implies that each wavelength can carry 1Gbps in the first two scenarios, but it scales to 40 Gbps in the third. Note that in this scheme two sets of wavelengths are needed: one for the traffic that optically bypasses the routers, and another for the rest. This is considered in the analysis to calculate the number of active OXC ports. The number of active OEO converters is equal to the number of active ports in the router.

In accordance to the results presented in Figure 7.5, I assume that 50% of the IPTV traffic optically bypasses the routers, 30% is sent to the router for electronic processing, and 20% of the TV channels are not distributed. Considering this, the power savings for all three scenarios (plus the `3kUHD_ep` scenario, which is the same as `3kUHD` but considering routers with an energy-proportional power consumption profile) are presented in Figure 7.8. The dashed lines represent the results from using selective joining only. The solid lines represent the power savings using the optical bypass protocol proposed in the current chapter.

When compared with the selective joining scheme proposed in the previous chapter, the

[1] If I assume an on/off behaviour, i.e., a switch consuming its 8.5 W of total power independently of the number of active ports, all results I present in this chapter change by less than 1%. This stems from the fact that the OXC is the node component with the lowest power consumption by a good margin, in any case.

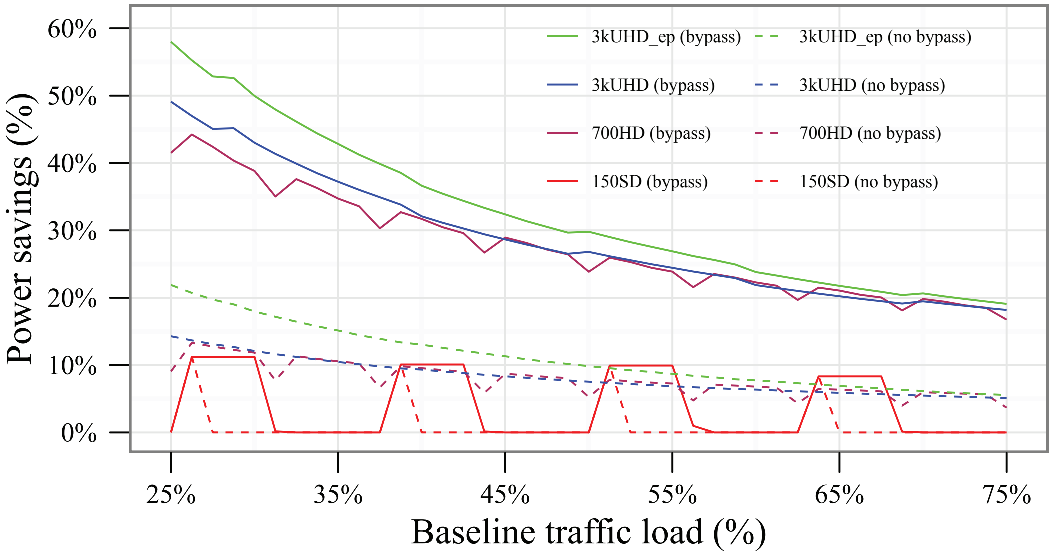


Figure 7.8: Power savings achieved by optically bypassing popular TV channels in the network core, as a function of the baseline traffic load in the node

introduction of optical bypass in the optical IPTV network core increases power savings substantially. At baseline traffic loads of around 30%, the power savings increase from 10% to 15% to over 40%. Considering EP routers, power consumption is halved. I conclude that the use of this technique is very effective in reducing power consumption, including in current IPTV service scenarios (such as 700HD).

7.7 Discussion: on the value of electronics

In the previous section I showed how optically switching *popular* IPTV traffic reduces power consumption significantly. How about optically switching *all* IPTV traffic? To answer this question, I invite the reader to look at Figure 7.9. This graph shows the result of optically switching all IPTV traffic in the network core (solid lines), against optically switching only the popular TV channels (dashed lines).

As can be observed, by optically switching all IPTV traffic the power savings increase even further. Considering a baseline traffic load of 25%, in the 700HD, 3kUHD and 3kUHD_ep scenarios an additional 20% power saving is achievable by all TV channels bypassing the routers.

So why not moving completely to optics in the future? In a scenario where all IPTV traffic is optically bypassed, to guarantee their availability for IPTV users, all TV channels need to be distributed continuously in the network core. This is because OXCs allow slow switching only[1]. The advantage of maintaining the electronic routing option is that, contrary to circuit-switched optical networks, with electronic routing it is possible not to distribute all TV channels. This

[1] By slow switching I mean switching technologies with speeds on a millisecond range. For instance, the 3D-MEMS switches considered in this chapter have a switching time of around 10ms [218], so an optical cross connect based on this technology is slow to reconfigure. This is in contrast to fast switching (nanosecond regime) as required for packet switching, for which no mature optical technology is yet available [218].

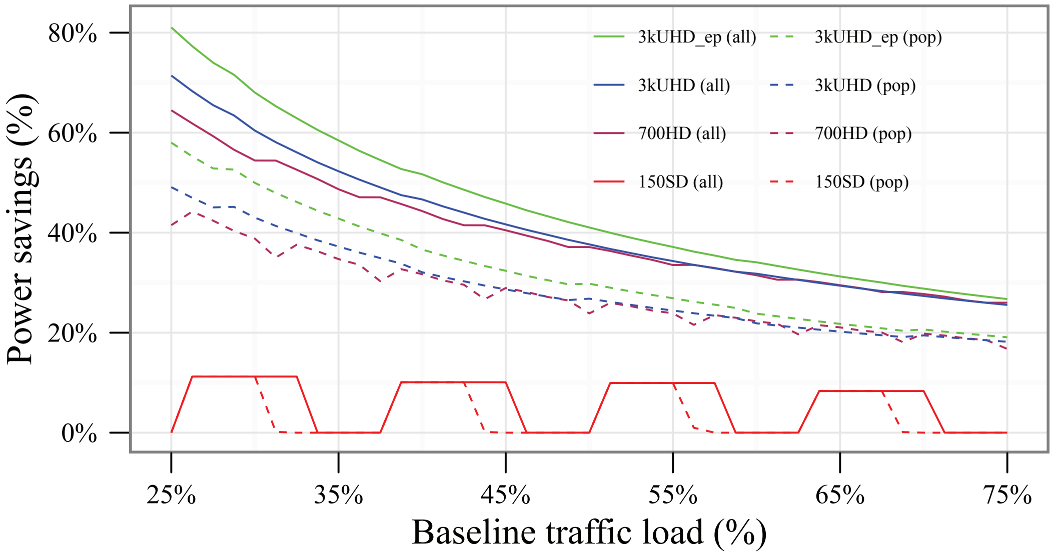


Figure 7.9: Power savings achieved by optically bypassing all TV channels in the network core (compared to popular TV channels only), as a function of the baseline traffic load in the node

added capability increases bandwidth efficiency. As explained in the introduction, with the increased popularity of narrowcasting services and niche channels, the number of unpopular channels (as defined in this chapter) may plausibly increase to the several hundreds or thousands in the near future. This trend offers an important argument for the maintenance of electronic routing as an option. A hybrid scheme as the one proposed in this chapter therefore offer a compromise between *energy* and *resource* efficiency IPTV service providers may want to consider.

7.8 Conclusions

In this chapter, I considered the introduction of energy-friendly optical technologies to reduce the energy consumption of IPTV distribution networks. I proposed an energy and resource-friendly protocol for the IPTV network core, blending electronic routing with all-optical switching. The main idea is to optically switch popular TV channels. This IPTV traffic bypasses the routers and therefore does not require any electronic processing (it is switched at the optical layer). The rest of the channels are sent to the routers for electronic processing (to be switched at the IP layer).

By analysing the IPTV dataset described in Chapter 4, I observed that by using the proposed protocol it is possible to switch 50% of the IPTV traffic all-optically. The energy savings obtained from optically bypassing this traffic are substantial. When compared with the selective joining scheme proposed in the previous chapter, the power savings in the network increase from 10% to 15% to over 40% under normal load conditions. The scheme is also bandwidth efficient as channels without viewers are not distributed. Finally, if all IPTV traffic is optically switched, instead of only the popular TV channels, the power savings increase even further. However, this comes with an increase of bandwidth inefficiency.

Chapter 8

Summary of contributions and future work

The closing chapter of the dissertation summarises the work upon which it is based and its original contributions. In addition, potential avenues for future research are proposed.

8.1 Summary of contributions

In this dissertation I studied and analysed three techniques to assist IPTV providers in the design of novel resource and energy efficient networks. These techniques addressed two relevant technological challenges currently faced by IPTV operators. The first such challenge is IPTV service's high channel change delay. Synchronisation and buffering of media streams can cause channel change delays of several seconds. The second is the question of how to maintain an operationally cost and energy efficient network in face of the evolution of IPTV services. Current static multicast solutions are inefficient, but dynamic multicast solutions also bring issues related to network scalability and service quality.

In face of these technological challenges, the first contribution of this dissertation was an empirical analysis of a particular solution to the channel change delay problem — predictive pre-joining of TV channels — using real IPTV usage data. In this scheme each Set Top Box simultaneously joins additional multicast groups (TV channels) along with the one requested by the user. If the user switches to any of these channels next, switching latency is virtually eliminated, and user experience is improved. Previous work on this subject used simple mathematical models to perform analytical studies or to generate synthetic data traces to evaluate these pre-joining methods. I demonstrated in this dissertation that these models are conservative in terms of the number of channel switches a user performs during zapping periods. They do not evidence the true potential of predictive pre-joining solutions, and were therefore an important motivation to perform such empirical analysis. The main conclusion of this study was that a simple scheme where the neighbouring channels (i.e., the channels adjacent to the requested one) are pre-joined by the Set Top Box alongside the requested channel, during zapping periods only, eliminates zapping delay for around half of all channel switching requests to the network.

Importantly, this result is achieved with a negligible increase of bandwidth utilisation in the access link.

The second contribution of this dissertation was related to the design and operation of IPTV networks. Current IPTV service providers build static multicast trees for the distribution of TV channels. This is justified to guarantee the quality of experience required by its customers. By distributing TV channels to as close to the users as possible, network latencies do not add significantly to the already high channel change delay. However, as particular channels have no viewers at particular time periods, this method is provably resource and energy inefficient. To reduce these inefficiencies, I proposed a semi-dynamic scheme where only a selection of TV multicast groups is distributed in the network, instead of all. This selection changes with user activity. This method was evaluated empirically by analysing real IPTV usage data. I demonstrated that by using the proposed scheme IPTV service providers can save a considerable amount of bandwidth while affecting only a very small number of TV channel switching requests. Furthermore, I also showed that although today the bandwidth savings would have reduced impact in energy consumption, with the introduction of numerous very high definition channels this impact will become significant.

To further increase the energy efficiency of IPTV networks, the third contribution of this dissertation was a novel energy and resource friendly protocol for core optical IPTV networks. The fundamental concept is to blend electronic routing and optical switching, thus gluing the low-power consumption advantage of circuit-switched all-optical nodes with the superior bandwidth-efficiency of packet-switched IP networks. The main idea is to optically switch popular TV channels. These can be categorised as long-lived flows, and are therefore perfect targets for this type of slow switching. With the use of this protocol, popular IPTV traffic optically bypasses the network nodes, i.e., this traffic avoids electronic processing. I evaluated this proposal empirically by performing a trace-driven analysis using real IPTV data. The main conclusion was that the introduction of optical switching techniques results in a quite significant increase in the energy efficiency of IPTV networks.

All the schemes studied in this dissertation were evaluated by means of trace-driven analyses using a dataset from an operational IPTV service provider. It is widely accepted that a thorough evaluation using real workloads enables the assessment of future network architectures with an increased level of confidence. This is particularly relevant in research fields that have relied heavily upon hypothetical user models which are different from the reality and can lead to incorrect estimation of system performance. Such is the case of IPTV systems research, which favours the use of evaluation methods as the one employed in this dissertation.

8.2 Future directions

In this dissertation I addressed two important technological challenges currently faced by IPTV operators: high channel change delay and network efficiency. The solutions proposed and analysed in this dissertation mitigate part of these problems. But many more persist. As such, I close this dissertation by suggesting possible directions for future research on these topics.

8.2.1 Improving channel change user experience

Most commercial solutions to the channel change delay problem attempt to ensure that an STB that is trying to join a new TV channel gets an auxiliary stream that starts with an I-frame and then offers some kind of mechanism to switch over to the main multicast stream (the boost stream solutions described in Section 3.2.2). This type of solution requires a dedicated zapping server to transmit a unicast burst when a channel change request is made. This is the most common fast channel change mechanism, and is used, for example, by the Windows Media Platform [141]. As the auxiliary stream starts with an I-frame, the zapping server maintains a delayed version of all TV streams. The unicast stream sent to the STB after a channel request is therefore a *delayed version* of the original multicast stream. To avoid glitches, when the STB switches to the multicast stream this one synchronises with the unicast stream, and is delayed for play out. The multicast stream in the STB is therefore out of synch with the original multicast stream that is being distributed in the network. In certain situations these delays may cause discomfort to the IPTV users (for example, your neighbour cheering a football goal before you see it), and hence is a challenge for IPTV network operators. A possible solution to this problem which may be worth investigating is to speed up the delayed multicast stream in order for it to catch-up the original stream. Informal subjective tests have shown that the variation of the playout speed is often unnoticeable by users [119, 187], so it may be a solution if performed in a controlled way. Kalman *et al.* [118, 119] have presented a similar idea in the past in order to buffer less video data, with the objective of reducing zapping delay.

8.2.2 Improving resource efficiency

IPTV services are bandwidth intensive. High definition TV requires bit rates on the tens of Mbps range, and future ultra high definition formats may increase this figure by orders of magnitude. So resource efficiency will continue to be a challenge for IPTV operators.

The type of solutions analysed in this dissertation to mitigate the high channel change delay problem of IPTV services assume several TV channels are sent simultaneously to the Set Top Box. This increases the bandwidth requirements of access networks. To alleviate this problem, a more efficient scheme would be to use Scalable Video Coding (SVC) techniques [208] with pre-joining solutions. SVC video streams contain one or more subset streams, or layers. A subset video stream is derived by dropping packets from the larger video to reduce the bandwidth required for the subset bitstream. The subset bitstream can represent a lower quality video signal, for instance. Sending the additional TV channels in lower quality (for example, by transmitting its base layer only) while transmitting all layers of the channel requested (to guarantee maximum quality for this particular channel) may offer an interesting tradeoff between switching latency and access network bandwidth cost worth investigating.

For the IP network core, constructing more efficient multicast distribution trees is another issue that deserves investigation. To build a multicast tree, PIM-SM [73], the most common multicast routing protocol [178], makes use of the unicast routing protocol topology information available through the routers forwarding table. This information, together with the group membership information, enables the construction of shortest-path trees (SPTs) from the source or from a core node (the Rendezvous Point in PIM). To be precise, in situations where paths are asymmetric, these are *reverse* SPTs because PIM uses unicast routing shortest-paths from the receiver to the source to build the branch of the tree from the source to the receiver [63].

In a recent paper, Xu *et al.* [213] have proved that optimal traffic engineering (TE) can be realised using link-state routing protocols with hop-by-hop forwarding. They presented a link-state protocol, PEFT, that provably achieves optimal traffic engineering while retaining the simplicity of hop-by-hop forwarding. By using PIM-SM with such unicast protocol it is therefore possible to obtain optimal TE *reverse* SPTs. But IPTV traffic consumes a very significant amount of bandwidth in the *forward* direction, from the source to its multiple destinations.

A solution to create optimal TE *forward* SPTs could thus be the following. A new source-initiated message would be added to PIM-SM. This message would be used to update the multicast routing table from the source to the receiver, allowing the construction of forward SPTs. This is a similar technique to the one used by the multicast protocol REUNITE [188]. By using this modified PIM-SM with a unicast protocol such as PEFT one would obtain optimal traffic engineered Shortest Path Trees in the forward direction.

8.2.3 Improving energy efficiency

I demonstrated in this dissertation that avoiding waste and opting for low-power switching are effective techniques to improve energy efficiency in IPTV networks. These solutions assumed little changes to current IPTV network architectures and topologies. An interesting future avenue would be the research on new network architecture designs and novel topologies with the objective of conceiving more environment-friendly IPTV networks.

An idea for future work in this area is to investigate energy-friendly placement strategies for PIM-SM Rendezvous Points (RPs). For scalability reasons, multicast protocols such as PIM-SM contain the option of having a single node from which branches of the multicast tree emanate. Scalability is obtained by the possibility of having a single multicast tree per group as opposed to one tree per (source, group) pair [21]. In PIM-SM this core node is called the Rendezvous Point (RP). The selection of the RP directly affects the structure of the tree, and therefore the performance of the network. An important problem in the construction of shared multicast trees is hence to determine the position of the RP. The choice of the RP allows network providers to perform traffic engineering, as in the recent work by Wang *et al.* [204]. The authors proposed a new algorithm, based on tabu search [87], to find the optimal placement for RP nodes. In their work, the cost to minimise to achieve this optimum is the sum of the average throughput of all links. Instead of minimising such variable, one could explore similar strategies that instead minimise energy consumption without negatively impacting performance.

All these techniques reduce energy consumption, but a truly environment-friendly IPTV network should have has its goal to reduce or eliminate the emission of greenhouse gases. With such purpose, Dong *et al.* [68] recently proposed a novel approach to minimise CO_2 emissions in optical networks (considering unicast traffic). The authors achieved their goal by assuming some network nodes have access to renewable energy sources and by maximising its use in the network. Another interesting avenue of research is to investigate similar techniques assuming IPTV multicast scenarios.

Appendices

Appendix A

From electronics to optics: enabling techniques

In this appendix I present some work that, despite its orthogonality to the proposal presented in Chapter 7, is closely related. Namely, I address optical multicast, traffic grooming, and aggregated multicast. These are important techniques for IPTV operators that want to take full advantage of the opportunities offered by the inclusion of novel optical technologies in their networks.

A.1 Optical multicast

Since the seminal work by S. Deering [62], in 1989, the multicast problem has been extensively studied in the electrical domain. More recently the research focus has integrated multicast in the optical domain, which is the focus of this section. As explained in Chapter 3, in an all-optical network a lightpath is an optical point-to-point connection from a source to a destination. Switching at intermediate nodes is done at the optical layer, so the path from source to destination is all-optical. In [174] this concept was generalised to that of a light-tree which, unlike a lightpath, has multiple destination nodes. An important advantage of optical multicast is signal transparency with respect to traffic type, bit rates and protocols. In addition, Wang and Yang have shown that the use of optical multicast leads to a significant reduction in the number of wavelengths required in most networks, thereby increasing network efficiency [207].

Issues on optical multicast can be classified as data plane or control plane issues. At the data plane level, the fundamental issues are the architecture of multicast-capable optical cross connects and network topology design (in particular, the optimal placement of network equipment). Concerning the former, nodes with optical multicast capability are usually implemented by using optical splitters. A light splitter has the ability to split an input optical signal into multiple identical output optical signals. The only difference is power reduction of the output signals. Ideally, for a light splitter with a fan out of n, the power at each output of the splitter is $\frac{1}{n}$. The power constraints on optical networks are therefore exacerbated by the presence of optical splitting, and this has to be considered in the network design phase (for example, to define where to place the optical amplifiers in the network, a problem addressed by Hamad and Kamal in [96, 97]). Other technique to split the signal is WDM multicasting. This type of optical multicast can be done by taking advantage of the non-linear nature of optical fibre (by making clever use of Self-Phase Modulation, SPM, as in [84], or Four Wave Mixing, FWM[1], as in [78], for instance), by utilising pump modulated parametric amplifiers[2], as in [36], or by using an active vertical couplers-based optical switch[3] [217]. All the node architectures referred so far are of the Split-and-Delivery (SaD) type. A different type of multicast node architecture was proposed by Ali and Deogun [9]: Tap-and-Continue (TaC). Contrary to splitting the signal, as in SaD, in the TaC architecture the data proceeds strictly along a path but intermediate nodes on the path may access the data themselves by tapping a small fraction of the signal. This mechanism reduces splitting loss, but still has the inherent limit on the number of times a signal can be tapped before it loses integrity. As usual, hybrid architectures also exist. Fernandez et al. [75], for example, proposed a novel architecture that tries to combine the advantages of both tapping and splitting. Their *2-split-tap-continue* node is similar to a SaD-based node. This difference is that it not only switches but also taps a fraction of the input power to the local node. The architecture also includes a novel interconnection network which is the key for the improved efficiency over both SaD and TaC architectures alone.

The design and optimisation of an optical network is a difficult problem, in particular due to the heterogeneity of the equipment. The cost of hardware precludes full deployment in all nodes of optical splitters and wavelength converters. Wavelength conversion (WC), the ability to convert an input signal received on one wavelength into an output signal on a different wavelength, is a desirable capability for an optical node, as it can help improve wavelength utilisation in the network. However, the costs of wavelength converters are still very high [222], and for that reason only a small subset of network nodes may realistically be WC-capable. Besides this, these nodes usually are capable of converting only from specific input wavelengths to another wavelength within a finite waveband, so they have limited wavelength-conversion capabilities. A multicast node is also expensive to implement due to the complexity of fabrication and the large number of amplifiers required. For this reason, current networks not only have sparse limited wavelength-conversion capabilities (not all nodes are WC-capable, and the ones that are have limited conversion capabilities), but also have sparse limited multicast capabilities (not all nodes are multicast-capable, and as light splitters have a finite fan-out they are limited on the number of outputs). If properly designed, however, these limitations are not a serious problem. Networks with just a few power splitters and wavelength converters have efficiencies close to that of a full WC- and multicast-capable network, as shown by Yang and Liao's work [214].

Besides network heterogeneity, other constraints need to be taken into account when designing and optimising an optical network. First, each link can have multiple fibres and multiple

[1] SPM and FWM are non-linear effects that arise in optical communication systems due to the dependency of the optical fibre refractive index on the intensity of the applied electric field. In highly non-linear fibre SPM broadens the signal's electromagnetic spectrum. FWM occurs in WDM systems. The mix of several wavelengths in one fibre gives rise to new (usually undesirable) signals at new frequencies.

[2] An optical amplifier capable of offering spectrally wide gain in any band of interest.

[3] Such optical switch allows optical multicast without excess optical splitting loss due to the optical gain available in active vertical coupler switch cells.

wavelengths per fibre. Second, light-trees that share a common physical link cannot be assigned the same wavelength (the wavelength-clash constraint [185]). Third, the power level of the signal on any wavelength must not degrade below a certain lower bound [143] (namely, the sensitivity of the receivers and of the optical amplifiers), and simultaneously should not exceed an upper bound due to the non-linearity effects [162]. Finally, other physical-layer impairments, such as dispersion, have to be considered [212]. The topology design problem of optimally placing the network nodes taking into account all these constraints is therefore extremely complex. The research on this issue has considered the optimal placement of splitters alone [8, 135] or jointly with the wavelength converters [41, 48, 216], and also assuming nodes with both splitting and wavelength-conversion capabilities [66].

Besides the above data plane issues, optical multicast requires algorithmic support from the control plane. On the control plane, the main issue is to solve the Multicast Routing and Wavelength Assignment (MC-RWA) problem. The MC-RWA problem involves establishing the multicast routes on the network, and determining the appropriate wavelength to be assigned to them, minimising the resources required (usually, the number of wavelengths). The combined problem is NP-complete, as proved by Ali and Deogun in [9]. For this reason, the RWA problem is often decoupled into two separate sub-problems: the routing sub-problem and the wavelength assignment sub-problem. The routing sub-problem is still NP-complete as it involves the construction of a Steiner Minimum Tree, but wavelength assignment can be solved in polynomial time, as shown in [47] and [134]. RWA problems are usually formulated as a Mixed Integer Linear Programming (MILP) problem [125, 183] and solved using optimisers, such as CPLEX [107]. It is only possible to solve MILP problems to optimality for very small networks. For realistic-sized networks, scalable heuristics as those proposed in [124, 125, 183, 185] are always necessary.

For the interested reader references [95, 173, 221] present detailed surveys of the literature on optical multicast.

A.2 Traffic grooming

Most applications have bandwidth requirements that are far less than that provided by a single optical wavelength (or lightpath). It is therefore economical to use a lightpath to concurrently support multiple connections. The process of allocating sub-wavelength traffic demands to specific lightpaths such that the resources are shared is known as the traffic grooming problem [70, 106]. Traffic grooming refers to techniques used to aggregate low-speed traffic streams onto high-speed wavelengths. As explained in Chapter 3, these lightpaths can then be all-optically switched in the intermediate nodes (optical bypass), and thus save energy. Wang et al. [206], for instance, show that traffic grooming mechanisms together with optical bypass are a feasible solution to reduce electrical ports consumption in IP routers.

There has been some work on the design and operation of optical networks to support traffic grooming of multicast applications. Most work emphasises the reduction of the required number of wavelength channels, as [120], but there is also investigation on minimising the number of higher layer electronic equipment (for example, IP ports) [198]. In the context of IPTV, most grooming is usually done at the source, with the TV head-end injecting all TV channels into one set of wavelengths and delivering it to the IP network core. Grooming can also be relevant to add local TV channels at the network edge, inserting local channels to be distributed to specific regions only.

A.3 Aggregated multicast

The capacity of a light-tree in core optical networks is much higher than the bandwidth required by most multicast flows. Therefore, it is not efficient to directly map a single multicast flow into one light-tree. To increase network efficiency, Zhu et al. [223, 224] have studied the problem of aggregating multicast flows in IP over optical networks. In [223] the authors show the problem is NP-complete, propose an optimal Integer Linear Programming (ILP) solution and an efficient heuristic. In [224] Zhu and Jue extend this work by separating pay-per-view and other secure channels from the rest.

The optical multicast flow aggregation problem is related to the aggregated multicast problem in IP networks, as studied by Cui et al. [49, 59]. However, while the objective of the former is to increase network efficiency, the motivation of the latter is scalability. The key idea of aggregated multicast is to force multiple IP multicast sessions to share a single distribution tree to reduce multicast state in routers.

References

[1] A. Abramson. *The History of Television, 1880 to 1941*. McFarland & Company, 1987. 19

[2] A. Adams, J. Nicholas, and W. Siadak. Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol specification (revised). In *RFC 3973*, 2005. 34

[3] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta. Somniloquy: Augmenting network interfaces to reduce PC energy usage. In *NSDI*, Boston, MA, Apr. 2009. 51

[4] C. Aggarwal, J. Wolf, and P. S. Yu. On optimal piggyback merging policies for video-on-demand systems. In *SIGMETRICS*, Philadelphia, PA, May 1996. 43

[5] K. Ahmad and A. Begen. IPTV and video networks in the 2015 timeframe: The evolution to medianets. *IEEE Communications Magazine*, 47(12):68–74, 2009. 22, 38

[6] Alcatel. Alcatel-Lucent WaveStar OLS 1.6T product specification. http://tinyurl.com/AlcatelWavestar. [Online; accessed 16-04-2012]. 119

[7] S. Aleksic. Analysis of power consumption in future high-capacity network nodes. *IEEE/OSA Journal of Optical Communications and Networking*, 1(3):245–258, 2009. 56, 121

[8] M. Ali. Optimization of splitting node placement in wavelength-routed optical networks. *IEEE Journal on Selected Areas in Communications*, 20(8):1571–1579, 2002. 135

[9] M. Ali and J. S. Deogun. Cost-effective implementation of multicasting in wavelength-routed networks. *Journal of Lightwave Technology*, 18(12):1628–1638, 2000. 134, 135

[10] AMD. White paper: Magic packet technology, 1998. 51

[11] G. Anastasi, I. Giannetti, and A. Passarella. A Bittorrent proxy for green Internet file sharing: Design and experimental evaluation. *Computer Communications*, 33(7):794–802, 2010. 51

[12] C. Anderson. *The Long Tail: Why the Future of Business is Selling Less of More*. Hyperion, 2006. 88

[13] M. Andrews, A. F. Anta, L. Zhang, and W. Zhao. Routing and scheduling for energy and delay minimization in the powerdown model. In *INFOCOM*, San Diego, CA, Mar. 2010. 53

[14] J. Baliga, R. Ayre, K. Hinton, W. V. Sorin, and R. S. Tucker. Energy consumption in optical IP networks. *Journal of Lightwave Technology*, 27(13):2391–2403, 2009. 36, 49, 88, 108, 111, 118, 121

[15] J. Baliga, R. Ayre, K. Hinton, and R. S. Tucker. Photonic switching and the energy bottleneck. In *PS*, San Francisco, CA, Aug. 2007. 49, 55, 111

[16] J. Baliga, R. Ayre, K. Hinton, and R. S. Tucker. Architectures for energy-efficient IPTV networks. In *OFC*, San Diego, CA, Mar. 2009. 53

[17] J. Baliga, R. Ayre, W. V. Sorin, K. Hinton, and R. S. Tucker. Energy consumption in access networks. In *OFC*, San Diego, CA, Feb. 2008. 49, 52

[18] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker. Green cloud computing: Balancing energy in processing, storage, and transport. *Proceedings of the IEEE*, 99(1):149–167, January 2011. 52

[19] J. Baliga, K. Hinton, and R. S. Tucker. Energy consumption of the Internet. In *COIN-ACOFT*, Melbourne, Australia, June 2007. 49

[20] A. Ballardie. Core Based Trees (CBT version 2) multicast routing. In *RFC 2189*, 1997. 34

[21] T. Ballardie, P. Francis, and J. Crowcroft. Core based trees (CBT). In *SIGCOMM*, San Francisco, CA, Sept. 1993. 34, 128

[22] A. Banerjee, L. Drake, L. Lang, B. Turner, D. Awduche, L. Berger, K. Kompella, and Y. Rekhter. Generalized multiprotocol label switching: an overview of signaling enhancements and recovery techniques. *IEEE Communications Magazine*, 39(7):144–151, 2001. 57

[23] D. Banodkar, K. K. Ramakrishnan, S. Kalyanaraman, A. Gerber, and O. Spatscheck. Multicast instant channel change in IPTV systems. In *COMSWARE*, Bangalore, India, Jan. 2008. 45

[24] L. A. Barroso and U. Holzle. The case for energy-proportional computing. *IEEE Computer*, 40(12):33–37, 2007. 50, 90, 97

[25] BBC. BBC iPlayer. http://www.bbc.co.uk/iplayer/. [Online; accessed 16-04-2012]. 22

[26] BBC. Niche TV growing quickly online. http://news.bbc.co.uk/2/hi/programmes/click_online/5108980.stm. [Online; accessed 16-04-2012]. 88

[27] A. C. Begen, N. Glazebrook, and W. V. Steeg. Reducing channel-change times in IPTV with Real-time Transport Protocol. *IEEE Internet Computing*, 13/3(3):40–47, 2009. 30, 43

[28] A. C. Begen, N. Glazebrook, and W. V. Steeg. A unified approach for repairing packet loss and accelerating channel changes in multicast IPTV. In *CCNC*, Las Vegas, NV, Jan. 2009. 28, 30, 43

[29] A. C. Begen, C. Perkins, and J. Ott. On the use of RTP for monitoring and fault isolation in IPTV. *IEEE Network*, 24(2):14–19, 2010. 20, 35, 38

[30] Y. Bejerano and P. Koppol. Improving zap response time for IPTV. In *INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009. 31, 45

[31] A. P. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier. A survey of green networking research. *CoRR*, abs/1010.3880, 2010. 54

[32] A. P. Bianzino, A. K. Raju, and D. Rossi. Apples-to-apples: a framework analysis for energy-efficiency in networks. *SIGMETRICS Performance Evaluation Review*, 38:81–85, January 2011. 54

[33] N. Bila, E. de Lara, M. Hiltunen, K. Joshi, H. A. Lagar-Cavilla, and M. Satyanarayanan. The case for energy-oriented partial desktop migration. In *HotCloud workshop*, Boston, MA, June 2010. 49, 51

[34] J. Blackburn and K. Christensen. A simulation study of a new green BitTorrent. In *ICC*, Desden, Germany, June 2009. 53

[35] J. Boyce and A. M. Tourapis. Fast efficient channel change. In *ICCE*, Singapore, Nov. 2005. 46

[36] C.-S. Bres, N. Alic, E. Myslivets, and S. Radic. Scalable multicasting in one-pump parametric amplifier. *Journal of Lightwave Technology*, 27(3):356–363, 2009. 134

[37] E. V. Breusegern, J. Cheyns, D. D. Winter, D. Colle, M. Pickavet, F. D. Turck, and P. Demeester. Overspill routing in optical networks: a true hybrid optical network design. *IEEE Journal on Selected Areas in Communications*, 24(4):13–25, 2006. 56

[38] T. W. Cable. TWCable TV for iPad. http://www.timewarnercable.com/nynj/learn/apps/twctv/. [Online; accessed 16-04-2012]. 20

[39] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan. Internet Group Management Protocol, version 3. In *RFC 3376*, 2002. 29, 32

[40] J. Caja. Optimization of IPTV multicast traffic transport over next generation metro networks. In *NETWORKS*, New Delhi, India, June 2006. 89

[41] Y. Cao and O. Yu. Optimal placement of light splitters and wavelength converters for multicast in WDM networks. In *ICCAS*, Busan, Korea, Sept. 2005. 135

[42] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain. Watching television over an IP network. In *IMC*, Vouliagmeni, Greece, Oct. 2008. 19, 35, 42, 60, 66, 75, 77, 78, 82, 88, 105, 111

[43] M. Cha, P. Rodriguez, S. Moon, and J. Crowcroft. On next-generation telco-managed P2P TV architectures. In *IPTPS*, Tampa Bay, FL, Feb. 2008. 20, 42

[44] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power awareness in network design and routing. In *INFOCOM*, Phoenix, AZ, Apr. 2008. 54, 97, 99

[45] P. Chanclou, S. Gosselin, J. F. Palacios, V. L. Alvarez, and E. Zouganeli. Overview of the optical broadband access evolution: a joint article by operators in the IST network of excellence e-Photon/One. *IEEE Communications Magazine*, 44(8):29–35, 2006. 37

[46] C. S. Chang, D. S. Lee, and Y. S. Jou. Load balanced Birkhoff-von Neumann switches. In *Workshop on High Performance Switching and Routing*, Dallas, TX, May 2001. 55

[47] B. Chen and J. Wang. Efficient routing and wavelength assignment for multicast in WDM networks. *IEEE Journal on Selected Areas in Communications*, 20(1):97–109, 2002. 135

[48] S. Chen, T. H. Cheng, and G.-S. Poo. Placement of wavelength converters and light splitters in a WDM network using the generic graph model. *Computer Communication*, 33(7):868–883, 2010. 135

[49] J.-H. Chi, M. Gerla, K. Boussetta, M. Faloutsos, A. Fei, J. Kim, and D. Maggiorini. Aggregated multicast: A scheme to reduce multicast states. Technical report, Internet Engineering Task Force, 2002. 136

[50] L. Chiaraviglio, M. Mellia, and F. Neri. Energy-aware backbone networks: A case study. In *ICC*, Desden, Germany, June 2009. 50

[51] C. Cho, I. Han, Y. Jun, and H. Lee. Improvement of channel zapping time in IPTV services using the adjacent groups join-leave method. In *ICACT*, Gangwon-Do, Korea, Feb. 2004. 47, 71

[52] K. J. Christensen, C. Gunaratne, B. Nordman, and A. D. George. The next frontier for communications networks: power management. *Computer Communications*, 27(18):1758–1770, 2004. 50, 51

[53] M. Christensen, K. Kimball, and F. Solensky. Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) snooping switches. In *RFC 4541*, 2006. 33

[54] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini. An energy saving routing algorithm for a green OSPF protocol. In *INFOCOM*, San Diego, CA, Mar. 2010. 53

[55] J. M. Cioffi, H. Zou, A. Chowdhery, W. Lee, and S. Jagannathan. Greener copper with dynamic spectrum management. In *GLOBECOM*, New Orleans, LA, Nov. 2008. 52

[56] Cisco. Cisco visual networking index: Forecast and methodology 2008-2013. 20

[57] E. Commission. Growth of the number of TV channels and multi-channel platforms in Europe continues despite the crisis. 88

[58] J. Crowcroft, M. Handley, and I. Wakeman. *Internetworking Multimedia*. Morgan Kaufmann, 1999. 32

[59] J.-H. Cui, J. Kim, D. Maggiorini, K. Boussetta, and M. Gerla. Aggregated multicast - a comparative study. *Cluster Computing*, 8(1):15–26, January 2005. 136

[60] C. da Manha (Portuguese newspaper). Meo ultrapassa os dez mil canais (*translation: Meo already has over ten thousand channels*). http://tinyurl.com/MeoTenK. [Online; accessed 16-04-2012]. 88

[61] T. Das, P. Padala, V. N. Padmanabhanand, R. Ramjee, and K. G. Shin. LiteGreen: saving energy in networked desktops using virtualization. In *USENIX ATC*, Boston, MA, June 2010. 49, 51

[62] S. Deering. Host extensions for IP multicasting. In *RFC 1112*, 1989. 31, 32, 133

[63] S. Deering, D. L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei. The PIM architecture for wide-area multicast routing. *IEEE/ACM Transactions on Networking*, 4(2):153–162, 1996. 128

[64] S. E. Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Stanford University, 1991. 31

[65] N. Degrande, K. Laevens, D. D. Vleeschauwer, and R. Sharpe. Increasing the user perceived quality for IPTV services. *IEEE Communications Magazine*, 46(2):94–100, 2008. 27, 28, 46

[66] D.-R. Din and C.-Y. Li. A genetic algorithm for solving virtual source placement problem on WDM networks. *Computer Communication*, 32(2):397–408, 2009. 135

[67] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment issues for the IP multicast service and architecture. *IEEE Network*, 14(1):78–88, 2000. 38

[68] X. Dong, T. El-Gorashi, and J. M. H. Elmirghani. IP over WDM networks employing renewable energy sources. *Journal of Lightwave Technology*, 29(1):3, 2011. 48, 54, 128

[69] R. Doverspike, K. Ramakrishnan, and C. Chase. Structural overview of ISP networks. In *Guide to Reliable Internet Services and Applications*. Springer, 2010. 99

[70] R. Dutta and G. N. Rouskas. Traffic grooming in WDM networks: past and future. *IEEE Network*, 16(6):46–56, 2002. 135

[71] Economist. In praise of television: the great survivor. http://www.economist.com/node/16009155. [Online; accessed 16-04-2012]. 20, 21

[72] A. Feldmann, A. Gladisch, M. Kind, C. Lange, G. Smaragdakis, and F.-J. Westphal. Energy trade-offs among content delivery architectures. In *CTTE*, Ghent, Belgium, June 2010. 53

[73] B. Fenner, M. Handley, H. Holbrook, and I. Kovelas. Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol specification (revised). In *RFC 4601*, 2006. 34, 37, 127

[74] W. Fenner. Internet Group Management Protocol, version 2. In *RFC 2236*, 1997. 32

[75] G. M. Fernandez, C. Vazquez, P. C. Lallana, and D. Larrabeiti. Tap-and-2-split switch design based on integrated optics for light-tree routing in WDM networks. *Journal of Lightwave Technology*, 27(13):2506–2517, 2009. 134

[76] M. Fiorani, M. Casoni, and S. Aleksic. Performance and power consumption analysis of a hybrid optical core node. *IEEE/OSA Journal of Optical Communications and Networking*, 3(6):502–513, 2011. 120, 121

[77] W. Fisher, M. Suchara, and J. Rexford. Greening backbone networks: Reducing energy consumption by shutting off cables in bundled links. In *SIGCOMM workshop on green networking*, New Delhi, India, Aug. 2010. 50

[78] M. P. Fok and C. Shu. Performance investigation of one-to-six wavelength multicasting of ASK-DPSK signal in a highly nonlinear bismuth oxide fiber. *Journal of Lightwave Technology*, 27(15):2953–2957, 2009. 134

[79] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S. C. Diot. Packet-level traffic measurements from the Sprint IP backbone. *IEEE Network*, 17(6):6–16, 2003. 35, 36, 92

[80] H. Fuchs and N. Farber. Optimizing channel change time in IPTV applications. In *BMSB*, Las Vegas, NV, Apr. 2008. 30, 31, 70

[81] C.-H. Gan, P. Lin, and C.-M. Chen. A novel prebuffering scheme for IPTV service. *Computer Networks*, 53(11):1956–1966, 2009. 47, 59, 69

[82] C. M. Gauger, P. J. Kuhn, E. V. Breusegem, M. Pickavet, and P. Demeester. Hybrid optical network architectures: bringing packets and circuits together. *IEEE Communications Magazine*, 44(8):36–42, 2006. 56

[83] M. Ghanbari, D. Crawford, M. Fleury, E. Khan, J. Woods, H. Lu, and R. Ravazi. Future performance of video codecs. Technical report, University of Essex, 2006. 27, 38, 88, 95

[84] M. Ghandour, S. Liu, D. K. Hunter, D. Simeonidou, R. Nejabati, and P. Petropoulos. Ultra high performance media multicasting scheme over wavelength-routed networks. In *OFC*, San Diego, CA, Mar. 2010. 134

[85] N. Ghani, S. Dixit, and T.-S. Wang. On IP-over-WDM integration. *IEEE Communications Magazine*, 38(3):72–84, March 2000. 110

[86] G. Gilder. *Life after Television*. W. W. Norton & Company, 1992. 21

[87] F. Glover. Tabu search — a tutorial. *Interfaces*, 20(4):74–94, 1990. 128

[88] A. Greenberg, G. Hjalmtysson, and J. Yates. Smart routers-simple optics a network architecture for IP over WDM. In *OFC*, Baltimore, MD, Mar. 2000. 56

[89] GreenTouch. Greentouch website. http://www.greentouch.org/. [Online; accessed 16-04-2012]. 88

[90] C. Gunaratne, K. Christensen, and B. Nordman. Managing energy consumption costs in desktop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed. *International Journal of Network Management*, 15(5):297–310, 2005. 50

[91] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen. Reducing the energy consumption of Ethernet with adaptive link rate (ALR). *IEEE Transactions on Computers*, 57(4):448–461, 2008. 50

[92] M. Gupta, S. Grover, and S. Singh. A feasibility study for power management in LAN switches. In *ICNP*, Berlin, Germany, Oct. 2004. 50

[93] M. Gupta and S. Singh. Greening of the Internet. In *SIGCOMM*, Karlsruhe, Germany, Aug. 2003. 48, 49, 50

[94] M. Gupta and S. Singh. Using low-power modes for energy conservation in Ethernet LANs. In *INFOCOM*, Anchorage, AK, May 2007. 50

[95] A. Hamad, T. Wu, A. E. Kamal, and A. K. Somani. On multicasting in wavelength-routing mesh networks. *Computer Networks*, 50(16):3105–3164, 2006. 135

[96] A. M. Hamad and A. E. Kamal. Optimal power-aware design of all-optical multicasting in wavelength routed networks. In *ICC*, Paris, France, June 2004. 134

[97] A. M. Hamad and A. E. Kamal. Optical amplifiers placement in WDM mesh networks for optical multicasting service support. *IEEE/OSA Journal of Optical Communications and Networking*, 1(1):85–102, 2009. 134

[98] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown. ElasticTree: saving energy in data center networks. In *NSDI*, San Jose, CA, Apr. 2010. 52

[99] K. Hinton, J. Baliga, R. Ayre, and R. Tucker. The future Internet — an energy consumption perspective. In *OECC*, Hong Kong, China, July 2009. 55

[100] K. Hinton, G. Raskutti, P. M. Farrell, and R. S. Tucker. Switching energy and device size limits on digital photonic signal processing technologies. *IEEE Journal of Selected Topics in Quantum Electronics*, 14(3):938–945, 2008. 55

[101] G. Hjalmtysson, J. Yates, S. Chaudhuri, and A. Greenberg. Smart routers-simple optics: an architecture for the optical Internet. *Journal of Lightwave Technology*, 18(12):1880–1891, 2000. 56

[102] H. Hlavacs and S. Buchinger. Hierarchical video patching with optimal server bandwidth. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 4(1):1–23, 2008. 43

[103] H. Holbrook, B. Cain, and B. Haberman. Using Internet Group Management Protocol version 3 (IGMPv3) and Multicast Listener Discovery Protocol version 2 (MLDv2) for Source-Specific Multicast. In *RFC 4604*, 2006. 32

[104] W. Hou, L. Guo, J. Cao, J. Wu, and L. Hao. Green multicast grooming based on optical bypass technology. *Optical Fiber Technology*, 17(2):111–119, 2011. 111, 119

[105] H. Huang and J. A. Copeland. Optical networks with hybrid routing. *IEEE Journal on Selected Areas in Communications*, 21(7):1063–1070, 2003. 56

[106] S. Huang and R. Dutta. Dynamic traffic grooming: the changing role of traffic grooming. *IEEE Communications Surveys and Tutorials*, 9(1):32–50, 2007. 135

[107] IBM. ILOG CPLEX optimizer. http://tinyurl.com/IBM-CPLEX. [Online; accessed 16-04-2012]. 135

[108] F. Idzikowski, S. Orlowski, C. Raack, H. Woesner, and A. Wolisz. Saving energy in IP-over-WDM networks by switching off line cards in low-demand scenarios. In *ONDM*, Copenhagen, Denmark, May 2010. 50

[109] ISO/IEC. Generic coding of moving pictures and associated audio information. In *ISO/IEC 13818*, 1995. 27

[110] ISO/IEC. Coding of audio-visual objects. In *ISO/IEC 14496*, 1998. 27

[111] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot. An approach to alleviate link overload as observed on an IP backbone. In *INFOCOM*, San Francisco, CA, Mar. 2003. 100

[112] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard. Networking named content. In *CoNEXT*, Rome, Italy, Dec. 2009. 52

[113] R. Jain. *The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation, and modeling*. Wiley-Interscience, 1991. 59, 65, 68

[114] U. Jennehag and S. Pettersson. On synchronization frames for channel switching in a GOP-based IPTV environment. In *CCNC*, Las Vegas, NV, Jan. 2008. 46

[115] U. Jennehag, T. Zhang, and S. Pettersson. Improving transmission efficiency in H.264 based IPTV systems. *IEEE Transactions on Broadcasting*, 53(1):69–78, 2007. 46

[116] M. Jimeno and K. Christensen. A prototype power management proxy for Gnutella peer-to-peer file sharing. In *LCN*, Dublin, Ireland, Oct. 2007. 51

[117] H. Joo, H. Song, D.-B. Lee, and I. Lee. An effective IPTV channel control algorithm considering channel zapping time and network utilization. *IEEE Transactions on Broadcasting*, 54(2):208–216, 2009. 46

[118] M. Kalman, S. T. Eckehard, and B. Girod. Adaptive playout for real-time media streaming. In *ISCAS*, Scottsdale, AZ, May 2002. 43, 127

[119] M. Kalman, E. Steinbach, and B. Girod. Adaptive media playout for low delay video streaming over error-prone channels. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(6):841–851, 2004. 43, 127

[120] A. E. Kamal. Algorithms for multicast traffic grooming in WDM mesh networks. *IEEE Communications Magazine*, 44(11):96–105, 2006. 135

[121] M. Karczewicz and R. Kurceren. The SP- and SI-frames design for H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):637–644, 2003. 46

[122] R. H. Katz. Tech titans building boom. *IEEE Spectrum*, 46(2):40–54, 2009. 48

[123] I. Keslassy, S.-T. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard, and N. McKeown. Scaling internet routers using optics. In *SIGCOMM*, Karlsruhe, Germany, Aug. 2003. 55

[124] F. Koksal and C. Ersoy. A flexible scalable solution for all-optical multifiber multicasting: SLAM. *Journal of Lightwave Technology*, 25(9):2653–2666, 2007. 135

[125] F. Köksal and C. Ersoy. Multicasting for all-optical multifiber networks. *Journal of Optical Networking*, 6(2):219–238, 2007. 135

[126] K. Kompella and Y. Rekhter. OSPF extensions in support of Generalized Multi-Protocol Label Switching (GMPLS). In *RFC 4203*, 2005. 57

[127] K. Kompella and Y. Rekhter. Routing extensions in support of Generalized Multi-Protocol Label Switching (GMPLS). In *RFC 4202*, 2005. 57

[128] R. Kooij, K. Ahmed, K. Brunnström, and K. Acreo. Perceived quality of channel zapping. In *CSN*, Palma de Mallorca, Spain, Aug. 2006. 29, 70

[129] I. Kopilovic and M. Wagner. A benchmark for fast channel change in IPTV. In *BMSB*, Las Vegas, NV, Apr. 2008. 43

[130] C. Y. Lee, C. K. Hong, and K. Y. Lee. Reducing channel zapping time in IPTV based on user's channel selection behaviors. *IEEE Transactions on Broadcasting*, 56(3):321–330, 2010. 48, 59

[131] E. Lee, J. Whang, U. Oh, K. Koh, and H. Bahn. Popular channel concentration schemes for efficient channel navigation in Internet Protocol televisions. *IEEE Transactions on Consumer Electronics*, 55(4):1945–1949, 2009. 48

[132] U. Lee, I. Rimac, and V. Hilt. Greening the Internet with content-centric networking. In *e-Energy*, Passau, Germany, Apr. 2010. 52

[133] F. Legendre, V. Lenders, M. May, and G. Karlsson. Narrowcasting: an empirical performance evaluation study. In *CHANTS*, San Francisco, CA, Sept. 2008. 88

[134] R. Libeskind-Hadas and R. Melhem. Multicast routing and wavelength assignment in multihop optical networks. *IEEE/ACM Transactions on Networking*, 10(5):621–629, 2002. 135

[135] H.-C. Lin and S.-W. Wang. Splitter placement in all-optical WDM networks. In *GLOBECOM*, St. Louis, MO, Nov. 2005. 135

[136] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan. A power benchmarking framework for network devices. In *IFIP*, Buenos Aires, Argentina, June 2009. 54, 96

[137] A. Mahimkar, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and Q. Zhao. Towards automated performance diagnosis in a large IPTV network. In *SIGCOMM*, Barcelona, Spain, Aug. 2009. 42

[138] J. Maisonneuve, M. Deschanel, J. Heiles, W. Li, H. Liu, R. Sharpe, and Y. Wu. An overview of IPTV standards development. *IEEE Transactions on Broadcasting*, 55(2):315–328, 2009. 20

[139] G. Malkin. RIP version 2. In *RFC 2453*, 1998. 34

[140] F. Mannie. Generalized Multi-Protocol Label Switching (GMPLS) architecture. In *RFC 3945*, 2004. 57

[141] Microsoft. Delivering IPTV with the Windows Media Platform. Technical report, Microsoft, 2003. 19, 20, 43, 127

[142] P. V. Mieghem. *Performance Analysis of Communications Networks and Systems*. Cambridge University Press, 2009. 111

[143] Y. Morita, H. Hasegawa, K.-I. S. Y., Sone, K. Yamada, and M. Jinno. Optical multicast tree construction algorithm considering SNR constraint and 3R regeneration. In *ECOC*, Brussels, Belgium, Sept. 2008. 135

[144] S. Nedevschi, J. Chandrashekar, J. Liu, B. Nordman, S. Ratnasamy, and N. Taft. Skilled in the art of being idle: reducing energy waste in networked systems. In *NSDI*, Boston, MA, Apr. 2009. 51

[145] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall. Reducing network energy consumption via sleeping and rate-adaptation. In *NSDI*, San Francisco, CA, Apr. 2008. 50

[146] NetFPGA. Netfpga website. http://netfpga.org/. [Online; accessed 16-04-2012]. 54

[147] Nielsen. The Nielsen company. http://www.nielsen.com/. [Online; accessed 16-04-2012]. 41, 77

[148] Nielsen. Nielsen's Q1 2010 three screen report. Technical report, The Nielsen Company, 2006. 20

[149] A. Odlyzko. Data networks are lightly utilized, and will stay that way. *Review of Network Economics*, 2(3), 2003. 50

[150] U. Oh, S. Lim, and H. Bahn. Channel reordering and prefetching schemes for efficient IPTV channel navigation. *IEEE Transactions on Consumer Electronics*, 56(2):483–487, 2010. 47, 59, 79

[151] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey. Extended RTP profile for Real-time Transport Control Protocol (RTCP)-based feedback (RTP/AVPF). In *RFC 4585*, 2006. 44

[152] F. Palmieri. GMPLS control plane services in the next-generation optical internet. *The Internet Protocol Journal*, 11(3):2–18, 2008. 57

[153] C. Panarello, A. Lombardo, G. Schembra, L. Chiaraviglio, and M. Mellia. Energy saving and network performance: a trade-off approach. In *e-Energy*, Passau, Germany, Apr. 2010. 52

[154] D. Papadimitriou and A. Farrel. Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) extensions. In *RFC 3473*, 2003. 57, 113

[155] M. Pervilä and J. Kangasharju. Running servers around zero degrees. In *SIGCOMM workshop on green networking*, New Delhi, India, Aug. 2010. 48

[156] B. Piper. United states IPTV market sizing: 2009-2013. Technical report, Strategy Analytics, 2009. 20

[157] O. B. Portal. Average advertised download speeds, by country. 72, 84

[158] J. Postel. User Datagram Protocol. In *RFC 768*, 1980. 30

[159] PT. Meo customers may now have their own TV channel. http://tinyurl.com/MeoOwnTV. [Online; accessed 16-04-2012]. 88

[160] T. Qiu, Z. Ge, S. Lee, J. Wang, J. Xu, and Q. Zhao. Modeling user activities in a large IPTV system. In *IMC*, Chicago, IL, Nov. 2009. 42, 47, 59, 60, 65, 68, 71, 82, 88, 95, 119

[161] T. Qiu, Z. Ge, S. Lee, J. Wang, Q. Zhao, and J. Xu. Modeling channel popularity dynamics in a large IPTV system. In *SIGMETRICS*, Seattle, WA, June 2009. 42, 65, 66, 67, 78, 82, 88, 89, 105, 109, 111, 116

[162] R. Ramaswami, K. Sivarajan, and G. Sasaki. *Optical Networks: A Practical Perspective*. Morgan Kaufmann, 2009. 108, 135

[163] F. M. V. Ramos, J. Crowcroft, R. J. Gibbens, P. Rodriguez, and I. H. White. Channel smurfing: minimising channel switching delay in IPTV distribution networks. In *ICME*, Singapore, July 2010. 23

[164] F. M. V. Ramos, J. Crowcroft, R. J. Gibbens, P. Rodriguez, and I. H. White. Reducing channel change delay in IPTV by predictive pre-joining of TV channels. *Signal Processing: Image Communication*, 26(7):400–412, 2011. 23

[165] F. M. V. Ramos, R. J. Gibbens, F. Song, P. Rodriguez, J. Crowcroft, and I. H. White. Reducing energy consumption in IPTV networks by selective pre-joining of channels. In *SIGCOMM workshop on green networking*, New Delhi, India, Aug. 2010. 23

[166] F. M. V. Ramos, F. Song, P. Rodriguez, R. Gibbens, J. Crowcroft, and I. H. White. Constructing an IPTV workload model. In *SIGCOMM Poster Session*, Barcelona, Spain, Aug. 2009. 47, 78

[167] P. Ranganathan. Recipe for efficiency: principles of power-aware computing. *Communications of the ACM*, 53(4):60–67, 2010. 87, 105

[168] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). In *RFC 4271*, 2006. 22

[169] J. C. C. Restrepo, C. G. Gruber, and C. M. Machuca. Energy profile aware routing. In *ICC*, Desden, Germany, June 2009. 53

[170] Reuters. Going niche is the future for Indian television. http://in.reuters.com/article/2010/10/20/idINIndia-52320320101020. [Online; accessed 16-04-2012]. 88

[171] J. Rey, D. Leon, A. Miyazaki, V. Varsa, and R. Hakenberg. RTP retransmission payload format. In *RFC 4588*, 2006. 44

[172] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. In *RFC 3031*, 2001. 57

[173] G. N. Rouskas. Optical layer multicast: Rationale, building blocks, and challenges. *IEEE Network*, 17(1):60–65, 2003. 135

[174] L. H. Sahasrabuddhe and B. Mukherjee. Light trees: optical multicasting for improved performance in wavelength routed networks. *IEEE Communications Magazine*, 37(2):67–73, 1999. 133

[175] A. Saleh and J. Simmons. Evolution toward the next-generation core optical network. *Journal of Lightwave Technology*, 24(9):3303–3321, 2006. 56, 108

[176] C. Sasaki, A. Tagami, T. Hasegawa, and S. Ano. Rapid channel zapping for IPTV broadcasting with additional multicast stream. In *ICC*, Beijing, China, May 2008. 29, 45, 70

[177] K. Sato, N. Yamanaka, Y. Takigawa, M. Koga, S. Okamoto, K. Shiomoto, E. Oki, and W. Imajuku. GMPLS-based photonic multilayer router (Hikari router) architecture: an overview of traffic engineering and signaling technology. *IEEE Communications Magazine*, 40(3):96–101, 2002. 56

[178] P. Savola. Overview of the Internet multicast routing architecture. In *RFC 5110*, 2008. 34, 37, 127

[179] G. Shen and R. S. Tucker. Translucent optical networks: the way forward. *IEEE Communications Magazine*, 45(2):48–54, 2007. 56

[180] G. Shen and R. S. Tucker. Energy-minimized design for IP over WDM networks. *IEEE/OSA Journal of Optical Communications and Networking*, 1(1):176–186, 2009. 54, 56, 108, 110, 111, 119, 120

[181] S. W. Sherman and J. C. Browne. Trace driven modeling: Review and overview. In *ANSS*, 1973. 65

[182] P. Siebert, T. N. M. V. Caenegem, and M. Wagner. Analysis and improvements of zapping times in IPTV systems. *IEEE Transactions on Broadcasting*, 55(2):407–418, June 2009. 28, 30, 31, 70

[183] N. K. Singhal, L. H. Sahasrabuddhe, and B. Mukherjee. Optimal multicasting of multiple light-trees of different bandwidth granularities in a WDM mesh network with sparse splitting capabilities. *IEEE/ACM Transactions on Networking*, 14(5):1104–1117, 2006. 135

[184] V. Sivaraman, A. Vishwanath, Z. Zhao, and C. Russell. Profiling per-packet and per-byte energy consumption in the NetFPGA Gigabit router. In *INFOCOM Workshop on Green*

Communications and Networking, Shanghai, China, Apr. 2011. 54, 90, 96, 97, 99, 109, 120

[185] N. Skorin-Kapov. Multicast routing and wavelength assignment in WDM networks: a binpacking approach. *Journal of Optical Networking*, 5(4):266–279, 2006. 135

[186] D. Smith. IPTV bandwidth demand: Multicast and channel surfing. In *INFOCOM*, Anchorage, AK, May 2007. 44

[187] E. Steinbach, N. Farber, and B. Girod. Adaptive playout for low latency video streaming. In *ICIP*, Thessaloniki, Greece, Oct. 2001. 127

[188] I. Stoica, T. S. E. Ng, and H. Zhang. REUNITE: a recursive unicast approach to multicast. In *INFOCOM*, Tel-Aviv, Israel, Mar. 2000. 128

[189] W. Sun, K. Lin, and Y. Guan. Performance analysis of a finite duration multichannel delivery method in IPTV. *IEEE Transactions on Broadcasting*, 54(3):419–429, 2008. 30, 31, 47, 59, 69, 70, 74

[190] A. S. Tanenbaum and M. V. Steen. *Distributed Systems: Principles and Paradigms*. Pearson Prentice Hall, 2007. 62

[191] G. Thompson and Y.-F. R. Chen. IPTV: Reinventing television in the internet age. *IEEE Internet Computing*, 13(3):11–14, 2009. 19, 20, 37

[192] P. Tsiaflakis, Y. Yi, M. Chiang, and M. Moonen. Green DSL: Energy-efficient DSM. In *ICC*, Desden, Germany, June 2009. 52

[193] R. S. Tucker. The role of optics and electronics in high-capacity routers. *Journal of Lightwave Technology*, 24(12):4655–4673, 2006. 55, 121

[194] R. S. Tucker. Green optical communications Part I: Energy limitations in transport. *IEEE Journal of Selected Topics in Quantum Electronics*, 17(2):245–260, 2011. 49

[195] R. S. Tucker. Green optical communications Part II: Energy limitations in networks. *IEEE Journal of Selected Topics in Quantum Electronics*, 17(2):261–274, 2011. 48, 49

[196] R. S. Tucker, K. Hinton, and G. Raskutti. Energy consumption limits in high-speed optical and electronic signal processing. *Electronics Letters*, 43(17):906–908, 2007. 55

[197] R. S. Tucker, R. Parthiban, J. Baliga, K. Hinton, R. W. Ayre, and W. V. Sorin. Evolution of WDM optical IP networks: A cost and energy perspective. *Journal of Lightwave Technology*, 27(3):243–252, 2009. 49, 88

[198] R. Ul-Mustafa and A. E. Kamal. Design and provisioning of WDM networks with multicast traffic grooming. *IEEE Journal on Selected Areas in Communications*, 24(4):37–53, 2006. 135

[199] H. Uzunalioglu. Channel change delay in IPTV systems. In *CCNC*, Las Vegas, NV, Jan. 2009. 29

[200] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez. Greening the Internet with nano data centers. In *CoNEXT*, Rome, Italy, Dec. 2009. 52

[201] S. Vanhastel and R. Hernandez. Enabling IPTV: What's needed in the access network. *IEEE Communications Magazine*, 46(8):90–95, 2008. 20, 37, 40

[202] N. Vasić and D. Kostić. Energy-aware traffic engineering. In *e-Energy*, Passau, Germany, Apr. 2010. 53

[203] D. Waitzman, C. Partridge, and S. Deering. Distance Vector Multicast Routing Protocol. In *RFC 1075*, 1988. 34

[204] H. Wang, X. Meng, M. Zhang, and Y. Li. Tabu search algorithm for RP selection in PIM-SM multicast routing. *Computer Communications*, 33(1):35–42, January 2010. 128

[205] H. Wang, A. Wonfor, K. A. Williams, R. V. Penty, and I. H. White. Demonstration of a lossless monolithic 16x16 QW SOA switch. In *ECOC*, Vienna, Austria, Sept. 2009. 108

[206] X. Wang, W. Hou, L. Guo, J. Cao, and D. Jiang. Energy saving and cost reduction in multi-granularity green optical networks. *Computer Networks*, 55(3):676–688, 2011. 135

[207] Y. Wang and Y. Yang. Multicasting in a class of multicast-capable WDM networks. *Journal of Lightwave Technology*, 20(3):350–359, 2002. 133

[208] T. Wiegand, L. Noblet, and F. Rovati. Scalable video coding for IPTV services. *IEEE Transactions on Broadcasting*, 55(2):527–538, June 2009. 127

[209] Wikipedia. Paul Gottlieb Nipkow. http://en.wikipedia.org/wiki/Paul_Gottlieb_Nipkow. [Online; accessed 16-04-2012]. 19

[210] Wired. CES 2008: Panasonics enormous 150" plasma TV dwarfs all competitors. http://www.wired.com/gadgetlab/2008/01/ces-2008-keynot/. [Online; accessed 16-04-2012]. 11, 38, 39

[211] Y. Xiao, X. Du, J. Zhang, F. Hu, and S. Guizani. Internet Protocol television (IPTV): The killer application for the next-generation Internet. *IEEE Communications Magazine*, 45(11):126–134, 2007. 19

[212] Y. Xin and G. Rouskas. Multicast routing under optical layer constraints. In *INFOCOM*, Hong Kong, China, Mar. 2004. 135

[213] D. Xu, M. Chiang, and J. Rexford. Link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering. In *INFOCOM*, Phoenix, AZ, Apr. 2008. 128

[214] D.-N. Yang and W. Liao. Design of light-tree based logical topologies for multicast streams in wavelength routed optical networks. In *INFOCOM*, San Franciso, CA, Mar. 2003. 134

[215] M. Yano, F. Yamagishi, and T. Tsuda. Optical MEMS for photonic switching-compact and stable optical crossconnect switches for simple, fast, and flexible wavelength applications in recent photonic networks. *IEEE Journal of Selected Topics in Quantum Electronics*, 11(2):383–394, 2005. 121

[216] K.-M. Yong, G.-S. Poo, and T.-H. Cheng. Optimal placement of multicast and wavelength converting nodes in multicast optical virtual private network. *Computer Communications*, 29(12):2169–2180, 2006. 135

[217] S. Yu, S.-C. Lee, O. Ansell, and R. Varrazza. Lossless optical packet multicast using active vertical coupler based optical crosspoint switch matrix. *Journal of Lightwave Technology*, 23(10):2984–2992, 2005. 134

[218] G. S. Zervas, M. D. Leenheer, L. Sadeghioon, D. Klonidis, Y. Qin, R. Nejabati, D. Simeonidou, C. Develder, B. Dhoedt, and P. Demeester. Multi-granular optical cross-connect: Design, analysis, and demonstration. *IEEE/OSA Journal of Optical Communications and Networking*, 1(1):69–84, 2009. 56, 122

[219] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner. Theoretical and practical limits of dynamic voltage scaling. In *DAC*, San Diego, CA, June 2004. 48

[220] Y. Zhang, P. Chowdhury, M. Tornatore, and B. Mukherjee. Energy efficiency in telecom optical networks. *IEEE Communications Surveys and Tutorials*, 12(4):441–458, 2010. 54, 108, 111, 120

[221] Y. Zhou and G.-S. Poo. Optical multicast over wavelength-routed WDM networks: A survey. *Optical Switching and Networking*, 2(3):176–197, 2005. 135

[222] Y. Zhou and G.-S. Poo. Multicast wavelength assignment for sparse wavelength conversion in WDM networks. In *INFOCOM*, Barcelona, Spain, Apr. 2006. 134

[223] Y. Zhu, Y. Jin, W. Son, W. Guo, W. Hu, W.-D. Zhong, and M.-Y. Wu. Multicast flow aggregation in IP over optical networks. *IEEE Journal on Selected Areas in Communications*, 25(5):1011–1021, 2007. 136

[224] Y. Zhu and J. Jue. Multi-class flow aggregation for IPTV content delivery in IP over optical core networks. *Journal of Lightwave Technology*, 27(12):1891–1903, 2009. 136