UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE INFORMÁTICA

USER BEHAVIOR IMPACT ON IPTV PLATFORM PERFORMANCE

Carlos Alberto Catarino Ferreira da Silva

Mestrado em Segurança Informática

Dezembro 2010

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

USER BEHAVIOR IMPACT ON IPTV PLATFORM PERFORMANCE

Carlos Alberto Catarino Ferreira da Silva

Tese orientada pelo Prof. Dr Hyong Kim
e co-orientada pelo Prof. Dr Marcelo Pasin

Mestrado em Segurança Informática

Dezembro 2010

Resumo

As plataformas de IPTV têm sido introduzidas pelos prestadores de serviços de rede e de telecomunicações como forma de competir com os operadores de cabo e satélite, impulsionando a receita por utilizador e simultaneamente reduzindo, na medida do possível, os custos associados à exploração de infra-estrutura legada. Através da análise de métricas de desempenho relativas ao funcionamento interno de um sistema complexo procuramos revelar padrões e ocorrências isoladas que possam indicar a existência de problemas de desempenho. Usando esta informação, analisámos também as características da actividade dos utilizadores. A hipótese subjacente é que deve ser possível correlacionar o desempenho de um sistema ou sub-sistema tendo por base o comportamento dos utilizadores, mesmo que estes não interajam directamente com o referido sub-sistema. Adicionalmente, esta correlação deve seguir regras que possam ser usadas, por exemplo, para redefinir a arquitectura do sistema ou detectar anomalias proactivamente. Neste projecto analisámos métricas de desempenho dos Servidores de Distribuição e dos Servidores de VOD. De seguida estudámos os Registos de Actividade das STB de forma a caracterizar as acções dos utilizadores que têm maior influência no desempenho da plataforma de IPTV. Demonstrámos que estas análises em domínios distintos (utilizadores/STB e servidores internos) pode convergir e possibilitar a activação de alarmes quando a probabilidade de ocorrência de problemas de desempenho é elevada.

Palavras chave: IPTV, comportamento de utilizador, desempenho, logs.

Abstract

IPTV platforms are being introduced by many network operators in order to compete with cable and satellite operators, increasing the average revenue per user, while taking advantage and reducing as much as possible the losses associated with legacy infra-structure. By analyzing performance metrics regarding the inner works of complex systems we aimed at unveiling patterns and identifying outlier occurrences that indicate actual or potential problems. Using such information we analyzed user activity information seeking to match the identified outlier patterns with characteristic user activity. The broader underlying hypothesis is that it is possible to correlate a system, or system's performance problem based on the behavior of the end users even if they do not interact with it. Furthermore, we assume that this correlation follow rules which could thereafter be used for architecture redesign and proactive anomaly detection. In this project, we examined performance metrics of the Distribution Servers and VOD Servers. We then used STB activity logs to characterize the user actions with greater influence on the IPTV platform. We showed that these analyzes of multiple domains (user/STB and internal servers) can be merged enabling to raise alerts whenever there is high probability of occurring efficiency problems in the system.

Acknowledgments

I would like to thank Professor Hyong Kim for his assertive and objective guidance. Professor Marcelo Pasin for the enriching discussions. Eng João Ferreira for his availability, for the long distance discussions and brainstorming.

A special thank you to the Carnegie Mellon University, to the Faculdade de Ciências da Universidade de Lisboa, to the CMU|Portugal program and Portugal Telecom for providing the conditions for me to participate in this outstanding program.

To my family for all the support despite my intermittent presence.

And Joana, thank you, just for everything.

Table of Contents

List of Figures

List of Tables

Abbreviations

DVR	Digital Video Recording
DRM	Digital Rights Management
GUI	Graphical User Interface
HD	High Definition
ICC	Instant Channel Change
IPTV	Internet Protocol Television
QoS	Quality of Service
RDP	Remote Desktop Protocol
RTP	Real-time Transport Protocol
SD	Standard Definition
SMSR	Stream Management Stream Request
STB	Set Top Box
UDP	User Datagram Protocol
VOD	Video On Demand

Chapter 1
Introduction

The increasing capacity and capabilities of operator's networks has led to great developments in the products and services that can be offered. Triple play services that aggregate high speed Internet, Voice and Internet Protocol Television (IPTV) are being introduced by many network operators in order to compete with cable and satellite operators, increasing the average revenue per user, while taking advantage and reducing as much as possible the losses associated with legacy infra-structure [1]. Nevertheless, IPTV services have unique and stricter requirements than the basic Internet services such as Web browsing or e-mail [2].

The IPTV digital television service generally consists in commercial multicasting of TV and Video On Demand (VOD), bundled with voice over IP (VoIP), as well as Web and e-mail access. IPTV has a different architecture from the traditional TV broadcasting systems where all content is pushed to the users. IPTV users actively select the channel they want to watch and only that channel is forwarded to each set. This architecture enables a variety of additional services such as Digital Video Recording (DVR) with scheduler, Time Shift program availability, or Remote Desktop (RDP) applications. All these services enable an increasing personalized use of the IPTV platform effectively converging communications, computing and content [3].

The IPTV Service is inherently resource intensive with a high degree of demand unpredictability. The basic Internet service has modest peak bandwidth requirements and a high degree of oversubscription is possible by aggregating the traffic of multiple users that are not constantly downloading or uploading at the same time. Therefore, the average bandwidth in the network is lower than the peak bandwidth offered to the users. On the other hand, TV streaming requires about 3 Mb/s of sustained bandwidth for a Standard Definition (SD) stream or 8 Mb/s for a High Definition (HD) stream. The major issues related with the nature of IPTV services are [4]:

- Concurrency of broadcast channels can occur in prime time periods when many users request many broadcast channels concurrently. The amount of requests stresses the multicast routing protocol which is required to have high availability. The Internet Group Management Protocol (IGMP) must efficiently manage the broadcasting tree branches and handle the increasing number of channel change requests. These can happen in bursts, for example, during commercial blocks, increasing unicast traffic with TV caching servers that assist the channel change process.
- Concurrency can also occur for a single broadcasted channel in sporadic events such as a popular sports event. This scenario results in very large multicast trees, highlighting the importance of highly available multicast routing.
- Concurrency of unicast sessions (such as VOD) is a major variable to consider in network design since each VOD content request consists in a unicast connection between the STB and the VOD server. At first, peak concurrency may be low, but with increasing diversity of available content, the costumer adherence will increase and the unicast video stream

traffic will become more significant. Occasional situations, such as the release of a popular content can also trigger high unicast concurrency. In summary, policy and priority based admission control as well as the VOD server network architecture play fundamental roles in avoiding resource contention.

- Growth of demand for HD content increases the overall bandwidth requirements as HD video streaming requires about 8 Mb/s, compared to the 3 Mb/s of the Standard Definition (SD). HD content increases the channel broadcasting bandwidth budget but has potentially more impact when considered in the context of the VOD service due to its unicast nature. Furthermore, HD is ever more used, for example, as a commercial argument.
- Set Top Box (STB) proliferation with multiple STB per household used simultaneously by different people watching different channels, and advanced services such as picture in picture or multi-angle viewing further increasing bandwidth requirements. Once a channel is tuned, the bandwidth requirements are constant until the end of the program and therefore oversubscription is not possible, not even within a household, or the cost would be noticeable quality degradation with screen freezes, image pixelization or audio distortion.

Differentiated Services (DiffServ) is the best solution to ensure Quality of Service (QoS) in unicast applications. However, unicast is not scalable for video broadcasting. IP multicast is a better way to improve QoS, saving bandwidth in the core and access networks by transmitting information destined to multiple users as aggregated as possible through a distribution tree. A channel change process is triggered by the user's request, which needs to be accepted by an admission control mechanism, and after which a multicast tree is be built. There are, nevertheless, issues when implementing multicast networks. The three main problems include: heterogeneous trees in terms of QoS requirements per tree branch, scalability issues due to state transfer between routers, and the Neglected Reserved Sub-tree (NRS) problem which occurs because one data flow can be replicated in multiple egress nodes, making the admission control and resource sufficiency check more difficult [5].

Important technical challenges are to be addressed when deploying an IPTV infra-structure. In general, the success of the architectural choices and the platform's performance depends on the validity of the user behavior assumptions. Therefore, it is important to keep a close look at the user's behavior characteristics, particularly those more relevant to the platform's performance. For example, zapping through the available channels is a frequent activity and the user expects the television set to respond quickly to his requests. Therefore, keeping track of these actions and knowing their implications on the server's performance is a key metric which can improve the assessment of the platform's adequacy to the real demand.

1.1 Motivation

Typically, the system performance is measured using observations and defining thresholds or estimations to trigger alerts. In large and complex systems, any such observation is likely to be related to other occurrences in other parts of the system which are not described in those observations. In any given time, a monitoring system is observing a consequence of a given problem and a potential cause for subsequent problems in other parts of the system. By analyzing available performance metrics regarding the inner works of a complex system we try to unveil patterns and identify outlier occurrences which indicate actual or potential problems. Using this information we will analyze user activity information seeking to match the identified outlier patterns with characteristic user activity.

The broader underlying hypothesis is that it is possible to correlate a system's performance problem based on the behavior of the end users. Furthermore, we assume that this correlation follows known rules, which could thereafter be used, for instance, in architecture redesign or proactive anomaly detection.

This study focuses on the architectural elements that are involved in critical activities of the IPTV platform, namely on the STBs, video cache servers (D-Servers) and VOD servers, using available logging reports. The questions we tried to answer are:

- What is the most relevant information that allows detecting or inferring performance issues, given the very large amount of produced logs?
    - How are the D-Servers (packet retransmission server) performance influenced by their usage pattern? Is D-Server performance different when serving requests:
        - For same channel/multiple channels?
        - Concentrated /spread in time?
        - From distinct clients/same clients?
        - From clients from same/different regions?
    - How does the Costumer's Instant Channel Change (ICC) usage (zapping through channels, also designated as costumer migration) affect D-Server performance? Considering the channel change's:
        - Duration (e.g. is the D-Server performance unaltered/worsen in periods where zapping duration is larger?)
        - Amount of traversed channels (e.g., is the D-Server performance unaltered/worsen in period where many costumers end zapping in the same channel?)
        - Channels traversed (e.g. is there any group of channels, for which D-Server performance is worse when they are zapped through?)
    - What is the VOD usage pattern and is VOD platform architecture and configuration adequate for such usage?
- Given the very large amount of produced logs, what is the relevant user activity information that sufficiently characterizes the platform's usage so that it is possible to characterize the real demand?
- Is there a user activity pattern correlated with identified performance issues?
- Is it possible to take the information of a performance issue and establish its root cause?

1.2 Organization of the report

The rest of this report is organized as follows. Chapter 2 provides an overview of the architecture of the IPTV platform that was analyzed. Chapter 3 presents the analysis of the logging information, considering the three main categories of Distribution Servers, VOD Servers, and Activity Logs from the STBs, first separately and then combining them together.

Chapter 4 summarizes the conclusions, identifies open issues and proposes future work directions.

Chapter 2
Background

In this section we provide a brief overview of the architecture of the IPTV platform used in this project. We also provide an overview of the main features and mechanisms which are the focus of this study, namely the Instant Channel Change mechanisms and the Video On Demand feature. These features and mechanisms were chosen due to their relevance as sources of unicast traffic of video over IP, and to their importance in ensuring the user's quality of experience.

2.1 IPTV architecture overview

The platform on which this study is based is a large scale system with distributed components serving a large and country-wide spread user base. The supported services are live TV broadcasting, VOD, and network applications through Remote Desktop Protocol (RDP). Figure 2-1 represents a simplified overview of the platform.

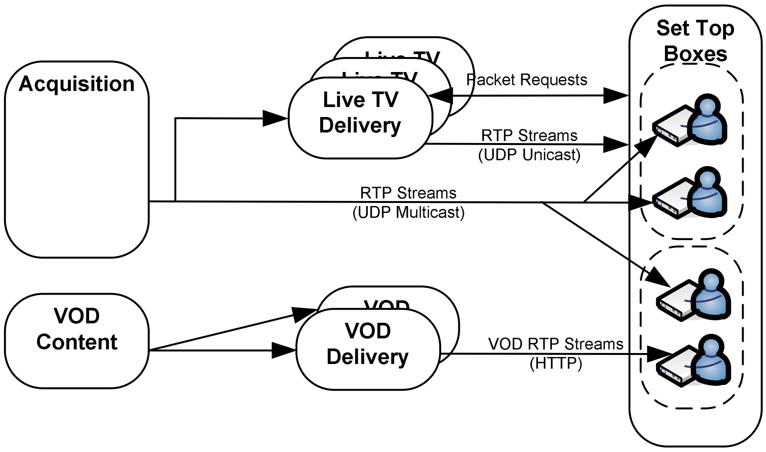

Figure 2-1 IPTV simplified architecture

The media contents are firstly acquired by the platform and then delivered to the set-top-boxes, located in the households, with which the costumers interact.

The live video acquisition is made from multiple sources depending business contracts with content producers. Because of this diversity, the platform contains a logical abstraction which normalizes the video content, namely the encoding, generates the digital video streams and encapsulates them in Real-time Transport Protocol (RTP) streams. The video streams are encrypted and protected with Digital Rights Management (DRM) technology, and are made available to the STBs using UDP multicast for Live TV and HTTP for VOD.

2.2 Packet retransmission

Although the multicast is done over UDP, the RTP encapsulation enables packet loss detection and reordering of packets in case the underlying IP infrastructure has delivered the packets out of sequence [6]. Therefore, when a STB is connected to a multicast stream receiving a Live feed, it periodically verifies whether there are missing packets in the stream. If so, it sends a request to a Delivery Server (D-Server), of the Live TV Delivery module (Figure 2-1), soliciting the missing packets.

The Live TV Delivery system also receives the video streams from the Acquisition module, and keeps a circular buffer with the most recent video content. Upon receiving a packet retransmission request from a STB, the Delivery server tries to retrieve the missing packets from the circular buffer. If it still holds the data related with the missing packets, it resends those packets to the STB in a unicast connection. If the packets are too old they may no longer exist in the buffer, and in such case the request fails.

2.3 Instant Channel Change

The channel change process is very important for the user's quality perception. If changing channels is not nearly instant, the user tends to get frustrated, his quality of experience decreases, and is likely to churn or file a complaint. There are multiple contributions for the channel change delay time such as multicast group joins, long key frame periods, audio-video synchronization delays or initial buffering delays. Several proposals have been made to deal with these problems using different approaches. One study proposes to adjust the encoding parameters, namely increasing the frequency of the intra-coded frames (I-Frames), with which the channel tune hooks, thus reducing the time to find the next I-Frame and reducing the channel change delay at the cost bandwidth [7]. Also on the encoding topic, Kalman et al. suggest reducing the initial buffering delay by using adaptive media playout (AMP), which is essentially a receiver driven rate limiter by accessing a different version of the content according to the connection's current capabilities [8]. Another proposal consists in having the home gateway, which is a multicast proxy for the STBs, pre join the multicast group of the channels to which the user is predicted to change. If the prediction is accurate, this mechanism decreases the multicast join delay associated with a channel change at the cost of bandwidth overhead [9]. Another approach consists in using a secondary "channel-change stream" associated with the multicast of regular quality stream for the requested channel. The STB joins the new multicast channel in the background while the play buffer is being filled by the "channel-change stream" resulting in smaller display latency [10]. Begen et al. [11] detail the best use of RTP and its control protocol in reducing the channel change times. Finally, to both reduce the effect of long key frame periods and initial buffer delays, a special server can be placed between the multicast source and the STB. The server holds a circular buffer with the most recent stream content. Upon receiving a join request, the server forwards the channel to the STB along with a supplementary unicast stream consisting of the content of the buffer. The unicast stream is sent in a burst to rapidly fill the receiver's buffer, which allows the STB to start decoding in a very short time because the initial buffering delay has been reduced [12,13]. This approach requires additional infra structure and also generates bursts of unicast traffic which can create a scaling problem, for example, when flash crowds occur. The IPTV platform in this study uses an approach similar to the later described approach. In Figure 2-1, the intermediate servers are represented as the Live TV Delivery Module, and they also accumulate the function of packet retransmission. This architectural element is interesting and relevant to analyze precisely because of the identified drawbacks.

2.4 Video On Demand

The Video On Demand service consists in making video content available such that users can browse through the video assets and rent them for a given time period and for a given fee. The sources of the video content vary as they are usually third party content providers. Therefore, the VOD system has a backed subsystem responsible for the acquisition and import of video content, supporting multiple methods and formats. This subsystem feeds the VOD delivery subsystem on which this study focuses.

Users can access the video content in any moment during the validity period, so the video assets have to be delivered in unicast streams whenever the user requests it. This is the reason why the delivery subsystem is interesting in this study. Even if two neighbors are accessing the same content, the traffic cannot be optimized in the network paths because each end point is an independent user controlling its content. Therefore, there is potential for network congestion. In an extreme case, if all users simultaneously requested VOD content, and the network was dimensioned assuming some aggregation gains due to multicast, the VOD traffic would congest the network.

To minimize the risk of network congestion or to deal with bandwidth constraints, the VOD delivery system can be geographically distributed. The most popular VOD assets should be placed nearer the STBs and the less popular in more central servers. This strategy also allows the system to provide failover capabilities, as the STB can first try to connect with the servers that are closer, but if the content is not available in that server or if the server is unavailable, they can try secondary servers.

The VOD servers execute a resource intensive activity, managing multiple unicast transfers of large amounts of data contained in large files. Another potential source for problems in performance is in the ability of the VOD servers to deal with the requests they receive. One main issue is whether video content is accessed from disk or from RAM. RAM based servers can provide higher egress capacity so, if available, they should hold the most popular contents. Disk based servers have lower egress capacity but they can hold a larger amount of data, being more appropriate for less popular assets.

Chapter 3
Data set analysis

In this section we analyze logging information from the Distribution Servers of the Live Delivery System, from the VOD Servers of the VOD Delivery System, and from the STB Activity Logs, which are the main components previously described in the IPTV architecture overview (see Chapter 2.1).

Our approach consisted in parsing the logs from their specific format into standard text files which were then imported to tables in a MySQL database. Since the logs were not known to represent a period where any particular problem had occurred, we firstly analyzed the server's logs searching for information that could prove or suggest that some performance issues may have happened. We then looked at the logs made available by the STBs and gave special attention to the information that represents the user's actions with most impact on the IPTV platform's workload. Finally, we brought these two perspectives together and correlated them, looking for combinations of user actions and system state that are more favorable to the occurrence of problems in the IPTV platform.

3.1 Distribution Servers

The D-Servers are the main components of the Live TV Delivery module (see Figure 2-1). They are responsible for serving packet retransmission requests and assisting the ICC process. D-Servers keep a circular buffer as cache of the most recent video content of every channel. If a packet retransmission request arrives from a STB, the D-Server retransmits the requested packets to the STB in unicast messages. If a STB performs a channel change, the D-Servers are also involved in providing an initial burst of video stream in unicast while the STB joins the multicast group. Additionally, when a user browses the channels, the D-Servers also provide the unicast video stream that is presented in a PIP window. Therefore, the workload on the D-Servers should be correlated with the aggregate activity of the STBs that consists in: retransmissions of lost packets regarding the tuned channel, browsing channels, and changing channels. Therefore, we try to find evidences of performance problems in the information provided by the D-Servers, for example, in the ratio of requested packets to serviced packets. This will enable a more meaningful correlation with other data sources, namely the Activity Logs.

3.1.1 Data set overview

The D-Servers provide cumulative information (every 5 minutes) containing the following information.

- Timestamp
- ClientID (16 bytes)
- ClientIP (4 bytes)
- ServiceID (16 bytes)
- Number of Retries Requests (1 bytes)

- Total Packet Requests (2 bytes)
- Total Packets Serviced (2 bytes)
- Max Hole size (1 bytes)

The raw data is organized in multiple files which are flushed hourly. Each file contains multiple reports. Each report is preceded with a line containing a timestamp for that report. The content of the report is similar to a hexadecimal representation of a memory dump. The first two "bytes" of the report represent the number of entries in it. The actual report immediately follows these two bytes. Each entry is fixed size and contains the information previously listed in 42 "bytes". Figure 3-1 represents a sample of this information where the report size is 0x00C3 (195 entries), and first entry is composed of the hexadecimal byte representation from position 0x0002 to 0x002C, that is, 42 "bytes". The fields of the first entry are represented in Figure 3-1 by interchanging the background color.

```
#PCS-DSVR-01
#2010:07:20 23:03:54:54 - 2010:07:21 00:00:00:00
2010:07:20 22:03:54:52|2010:07:20
23:03:54:52T1|16952|Information|Microsoft.TV2.Edge.Core.AVControl.SessionRetryReport|Mic
rosoft.TV2.Edge.Service.exe||4296|17|195 session retry reports for 190 unique clients.
SessionRetryReportData:
0000: 00 c3 b1 23 cf ca 7a c2 40 1e 84 5f b7 e9 d9 15 : ...#..z.@.._....
0010: 27 15 0a c3 41 ec a9 37 87 82 0b 70 4e 21 8b 69 : '...A..7...pN!.i
0020: 2a 2f 22 58 81 94 1f 00 30 00 30 04 3d 13 0c 5d : */"X....0.0.=..]
0030: 4f a1 4b a4 89 9d 9b 75 65 f5 23 5e 0a c3 5b c7 : O.K....ue.#\^..[.
0040: 9c 46 e3 42 c4 b9 4d f1 a1 47 4f 99 b7 a4 b9 13 : .F.B..M..GO.....
```

Figure 3-1 Sample of a D-Server session report. The characters in bold represent one entry and the fields are represented by interchanging the background color of the text

The information available for this study refers to 2 days and 79 D-Servers, which is slightly more than halve of the total amount of servers in the platform. The logs refer to July 20th and July 21st 2010, but since the Activity Log analysis will be focused on the July 20th, the analysis of the D-Servers is also focused on July 20th solely. A total of 3761 files were parsed into a conventional comma separated values text file and then imported into the database. Unless otherwise stated, all charts scaled from 0:00 to 24:00 refer to the full day of July 20th.

Each server provides the information grouped in reports with the same timestamp. The reports are separated by 5 minutes to which their information refers to. Nevertheless, the moments in which the statistics are recorded, that is, the timestamps, are not the same in every server. This 5 minute aggregation leads to a potential 5 minute time offset when we map the timestamps to time periods (e.g. to make a daily, hourly or 15 minutes aggregated analysis). This occurs because a given time period includes information from timestamps matching its first 5 minutes. Nevertheless, the information on such timestamp refers to its previous 5 minutes, which actually refers to outside the wider time period. The inverse occurs in the upper bound of the time period. Therefore, there is a forward offset of the information overlapping the time period's limit. This offset is different from server to server depending on how out of phase the pace of the timestamp is with the pace of the time period. Figure 3-2 represents these two scenarios.

Using a larger time period makes the offset bias less significant but reduces granularity. We will mostly use 15 minutes time periods which provide some granularity to understand trends within hourly periods and still reduce the influence of the referred offsets (on average the offset should be 2,5 minutes in a 15 minute period).

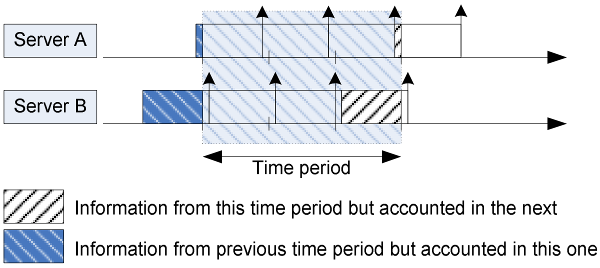

Figure 3-2 Offset caused by the lack of synchronized between the timestamps of the servers and between the timestamps and the analysis time period

Figure 3-3 represents the information reported by all servers grouped every 15 minutes. The count of distinct client IDs is also presented using the secondary vertical axe and allows a perception of the size of the audience with which the servers are connected. We can see that, as the day progresses, the total amount of retry requests reaches a rather constant value of around 110000 requests per 15 minutes by 8:00, while the amount of packets that are requested keeps increasing until it also reaches an apparent maximum value of approximately 260000 packets from 13:00 onward. From 13:00 to 16:00 occurs a period where both the number of retries and the number of packet requests oscillate reaching 4 distinctive peaks, until they stabilize. During this stable period, the ratio of requested packets per request is approximately 2,4 packets/request. It is also worth noticing that the first metric that increases in the first peak is the amount of packet requests and not the amount of retry requests. This means that there is a sudden increase in the amount of packets solicited in each request and not an increase in the overall amount of requests.

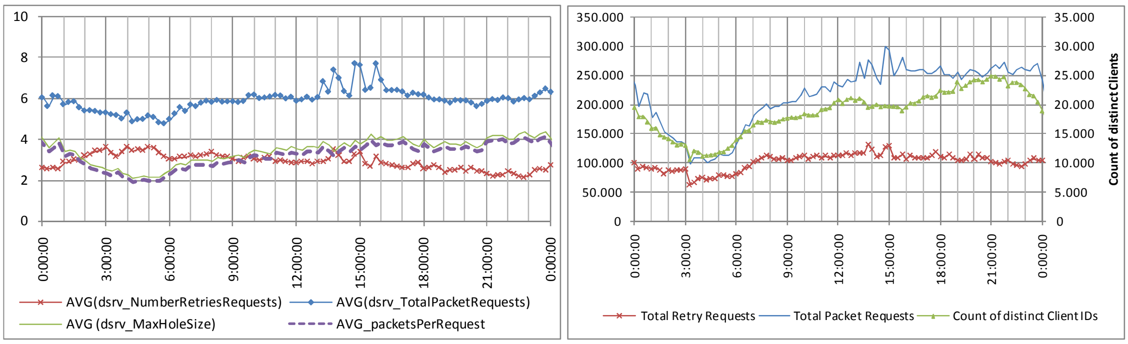

Figure 3-3 Average retry requests, packet requests, maximum hole size, and packets per request in 15 minutes aggregates (left), and overall amount of requests, packet requests, and count of distinct client IDs (secondary vertical axe) in 15 minutes aggregates (right)

The number of distinct client IDs follows the trend of the total requested packets until 13:00. Nevertheless, while the number of requested packets stops increasing at 13:00, the number of distinct clients resumes increasing around 16:00 until it peaks around 21:00. The distribution of the count of distinct clients reflects the expected daily distribution of a TV audience: low value during dawn, increasing during the morning, stabilizing or slightly decreasing after lunch, then increasing in the afternoon as people return home, and peaking after dinner which is the prime time. What is interesting is that despite this evolution regarding distinct clients, the total requested packets remains roughly constant throughout the afternoon and evening, barely evidencing the occurrence of the prime time period. This behavior may be due to load balancing mechanisms. As previously mentioned, this data set does not contain information of all the servers. Therefore, it is possible that the workload (requested packets) on these servers stabilizes, while increasing on the remaining servers. Simultaneously, since the overall active STBs increases, the number of distinct clients that connect to these servers also increases, but the amount of requests that each one of them does is naturally smaller.

The most relevant information to retain from this analysis is that from 13:00 to 16:00 appears to occur a transient period after which the global amount or packet requests on this set of servers stabilizes, while the amount of distinct STBs keeps increasing. This result suggests that a load balance mechanism may be increasing its activity from this period onward.

If we breakdown this information per server we can see that outstanding patterns are noticeable and are otherwise attenuated by the averaging effect of the large numbers involved in these statistics. This is what we explore in the next section.

3.1.2 Per Server analysis

Grouping the information per server unveils patterns in the different metrics. Figure 3-4 allows to visual identify a pattern in the metrics reported by D-Servers which consists in a band inside which most of the values are contained. It is also possible to visually identify outliers to the "normal" behavior consisting in peaks in the metrics of a subset of servers.

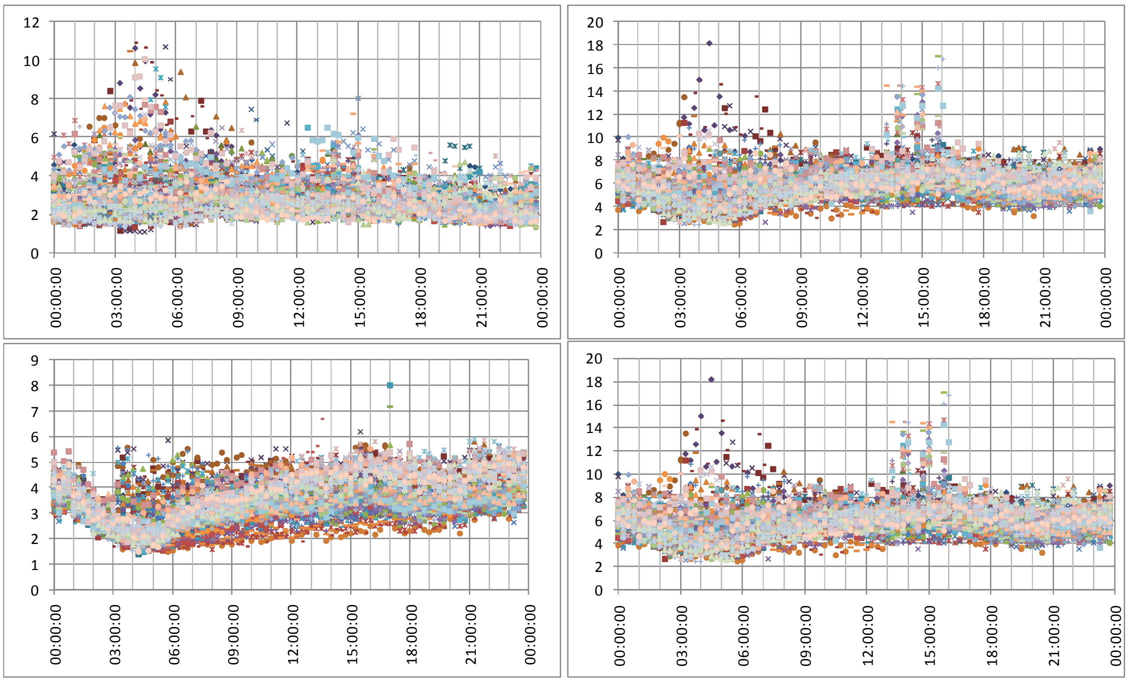

Figure 3-4 Average retry requests (top left), average requested packets (top right), average max hole size (bottom left), and average packets serviced (bottom right) per server, in 15 minutes aggregates

This first analysis allows verifying that the average packet requests and average packets serviced do not have a noticeable different behavior and show similar pattern in what concerns the band of "normal" values as well as the moments where outliers occur. This result was already expected since most of the time, the amount of served packets should be equal to the amount of requested packets. Significant differences in these metrics would indicate severe efficiency problems. The computation and analysis of the relation between these two metrics is made in section 3.1.3.

The noticeable outliers on the average requested packets are located during dawn and in the beginning of the afternoon. During the dawn period, the outliers of the average packet requests per server are sparse and each outlier belonging to a server is isolated, that is, there seems to be no correlation between each server's behaviors.

During the beginning of the afternoon, we verify the occurrence of 4 peaks which correspond to those previously identified in the global analysis of Figure 3-3. The disaggregation per server allows discovering that these peaks are actually a result of a subset of servers.

One of the servers in this subset is server 65. Focusing on this server and on the requested packets metric, we see (Figure 3-5) that the average value is 6,3 packets. Four peaks can be indentified at 13:15, 14:00, from 14:45 to 15:00, and from 15:45 to 16:00. The probability distribution function (pdf) of the packet requests data points of server 65 approximately follows a normal distribution (Figure 3-5).

In a normal distribution, about 95% of the values are within 2 standard deviations of the mean. We computed the distance between each data point and the average, and then identified the distances that are larger than twice the standard deviation. These points are marked in blue in Figure 3-5. This simple method allowed identifying the previously referred outlier data points. These outlier values suggest the occurrence of workload peaks but are not by themselves indicators of errors or failures.

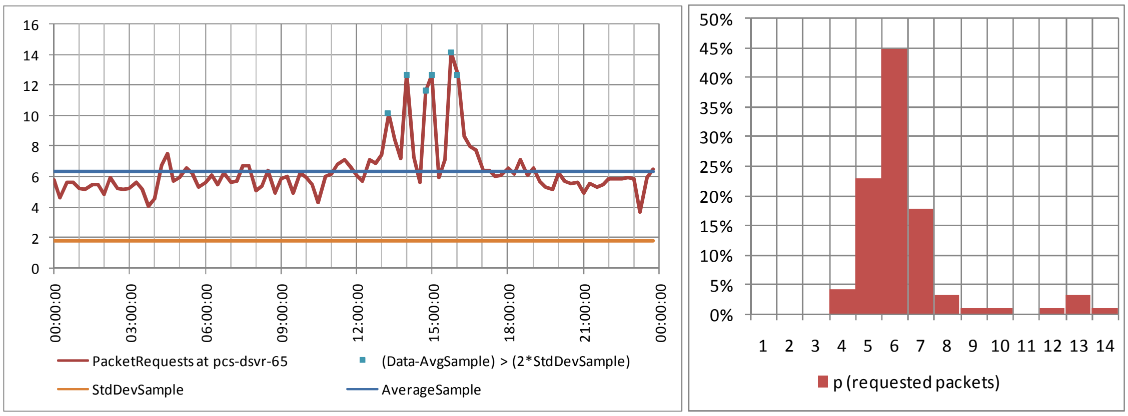

Figure 3-5 Detail of the packet requests regarding server 65 which participates in the afternoon peaks (left) and probability distribution function of the packet request values for the same server

Extending this analysis for the remaining servers, we obtained, for 1 day of information and 79 servers, a total of 395 outliers, which means 5 outliers on average per server. The amount of information may seem relatively large but it allows a clearer perspective of the information which is represented in Figure 3-6. We can now identify different patterns of the distribution of the outliers themselves.

During the dawn there is an area below 5 packet requests which is identified as outlier. This is justified by the smaller workload requested by users as this period is where most people are sleeping. Such small amount of requested packets is actually significantly lower than average thus it is signaled as outlier. For this reason they can be ignored.

During the same period (dawn) there are also sporadic large outlier values for individual servers. This may occur as consequence of increased internal server workload due to batch jobs or maintenance tasks.

Finally, the most distinctive characteristic is the confirmation of the peaks that were visually identified in Figure 3-4. A simple query with the count of outliers per time period clearly identifies the 4 peaks at 13:15, 13:45, from 14:45 to 15:00, and from 15:45 to 16:00. Furthermore, we can now easily determine which servers have outlier values in these moments and use this information for further investigation. Another interesting aspect of these results is that from 18:00 to 23:00 the amount of distinct connected clients increases but there are very few identified outliers. This result is coherent with the fact that the overall amount of requested packets in these servers did not increase during prime time (Figure 3-3). It also means that the increasing number of distinct unicast connections that the servers had to deal with did not cause any outstanding peaks to the amount of requested packets. If this was the case, it could be an indication of problems dealing with larger amount of unicast connections.

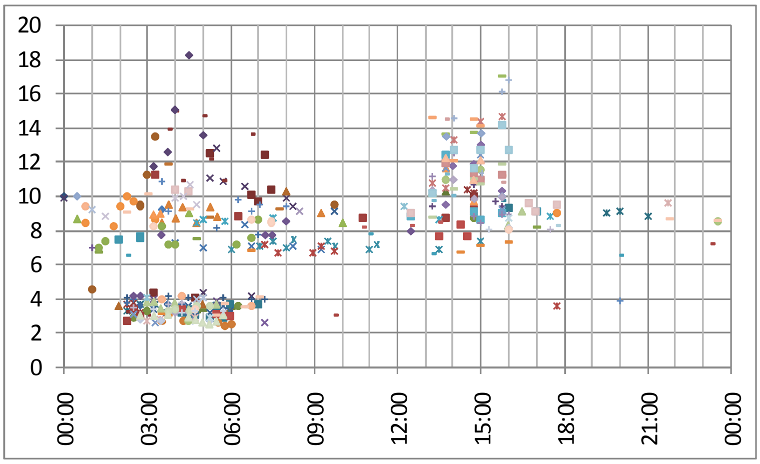

Figure 3-6 Average amount of requested packets in 15 minutes aggregates that are considered outliers in each server's distribution

The number of retry requests (regardless of the amount of requested packets) does not show clear outliers. As multiple packets are solicited in a single request, an abnormal peak in packet requests may not increase proportionally the amount of distinct requests. This leads to the relevance of analyzing the distribution of the amount of packets per request, for each server.

Figure 3-7 shows the average amount of requested packets in each retransmission request using a different symbol for every server. These values represent the sum of requested packets divided by the sum of received request, that is, it aggregates in each timestamp the information from all the channels that were serviced by each server. By analyzing Figure 3-7 we can observe that the normal value for this metric is clearly bounded between approximately 1 and 4 packets per request, with slight oscillation throughout the day. Some outstanding occurrences were then identified. Around 3:15 of July 20th there are multiple servers reporting much larger values of packets per request, suggesting that a problem may have occurred in some element that is common with those servers. Similarly, around 17:00 there are also multiple servers, although less than at 3:15, reporting very larger than average values for amount of packets per request. In this case there are approximately 4 servers involved (while at 3:15 there were 13: servers 6, 16, 18, 23, 33, 41, 14, 20, 4, 43, 2, 28, 39). Another interesting feature is that from 3:15 until 12:00 of July 20th, multiple servers reported above average packet per request ratio (see the less dense area above the band drawn by the average behavior in Figure 3-7).

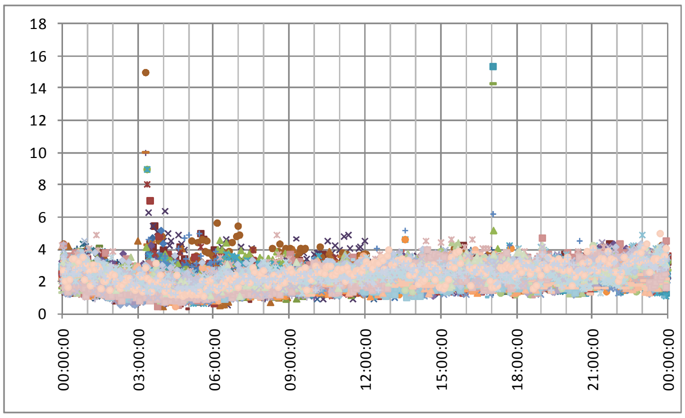

Figure 3-7 Amount of packets per each request for every server

In order to look into these occurrences, we filtered out all servers' information except the one from those involved in the 3:15 and 17:00 peaks, and plotted them in Figure 3-8.

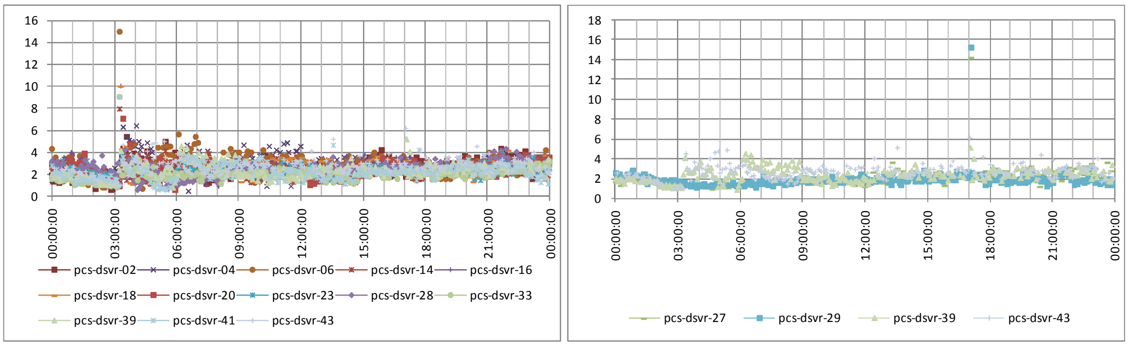

Figure 3-8 Amount of packets per each request for the servers involved in the 3:00 peak (left) and 17:00 peak (right)

The two situations prove to be different. In the 3:15 occurrence we see that, simultaneously with the peak, the average packets per request also increases and for a small period, the lower bound of all events breaks the trend and increases. It is also noticeable that the band drawn by the plots has a disruption simultaneously with the peak. It then gradually converges to a cohesive band around 12:00. The 17:00 peak does not appear to have caused such a disruption in the overall behavior of the systems. Although it just involves 4 servers, there is not a very significant change in the packets per request in the subsequent records for these servers. Since servers 43 and 39 are also involved in the 3:15 event, it is clear (from the graph on the right in Figure 3-8) the different consequences that the 3:15 event had on these servers when compared with the 17:00 event.

In summary, from the available metrics (number of retry requests, number of packet requests, number of packets serviced, and maximum hole size), the number of packet requests showed to explicitly reflect unusual behavior. By looking into the pdf of the number of packet requests of a specific server during one day, we visually verified that it resembles a normal distribution. Therefore, we used the rule that applies to normal distributions where approximately 95% of the data points are within 2 standard deviations of the mean. Computing this limit allowed the detection of outlier data points matching the same points that were visually identified. Extending this analysis to all servers we obtained a framework allowing the identification of periods of time and specific servers with unusual amounts of packet requests from STBs. This phenomenon can be indicators of problems, for example, on the multicast group management, or on the access network. We also observed that the overall amount of packets requested to this set of servers peaked and stabilized roughly from 16:00 onward, despite the amount of distinct clients kept increasing. This suggests that there is a load balancing mechanism using other servers that are not on this set, and that the increasing number of distinct unicast connections did not cause any problem that would provoke a peak in the amount of requests.

3.1.3 D-Server's efficiency

In order to individually analyze each server in terms of its actual response to the requests it receives, we computed the D-Server's efficiency as the ratio between served packets and requested packets. Using the full data sample, we realized that there are samples for which this ratio is greater than 1 (Figure 3-9). This means that there are records showing more served packets than requested as the one shown in Table 3.1.

dsrv_ time	dsrv_ server	dsrv_ clientID	dsrv_ IP	dsrv_ service ID	dsrv_ Number Retries Requests	dsrv_ Total Packet Requests	dsrv_ Total Packet Serviced	dsrv_ Max Hole Size	DServer Efficiency
20-07-2010 2:06	--	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx	a.b.c.d	yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy	255	1	231	7	231

Table 3.1 Example of one D-Server statistic record showing more served packets than requested packets

The reason for this occurrence is not clear. One hypothesis is that some requests accounted as "packet requests" in one 5 minute report could be served only on the following 5 minute report, causing efficiency to be smaller than 100% in the first period and greater than 100% in the following. Nevertheless, by querying the raw data this hypothesis was denied. An alternative hypothesis considers the gaps in the 5 minute reports of some servers (as we will see in section 3.1.4). These gaps could mean that the servers were somehow overloaded and producing incoherent information. Nevertheless these occurrences of over 100% efficiency do not coincide with those gaps and by querying the raw data we also saw that there was no apparent data incoherence across the gaps, there was just data missing. Finally, another possibility is that some requests are proxyed from other servers for which we do not have the statistical information and are accounted as "packets serviced" but not accounted as "packet requests" in the final server. This last hypothesis was not confirmed.

One way to narrow this problem was to make the D-Server's data coherent with the available data from the Activity Logs of the STBs. For this purpose we filtered out from the original set of D-Server statistics all the records from users that are not accounted in the Activity Logs. This filtering reduced the amount of the D-Server records from 6880261 to 182799, and the amount of distinct users from 135265 to 3019. The 3019 users (STB) compares with the 10746 distinct STBs available on the Activity Logs for July 20th. The remaining STBs in those Activity Logs where apparently connected with the remaining D-Servers or did not communicate with any during the period.

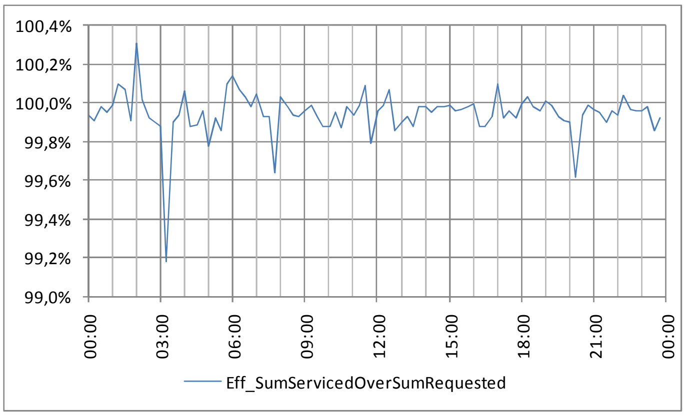

Figure 3-9 Server's efficiency in 15 minute aggregates

After filtering out these records, we recomputed the efficiency in the same way (see Figure 3-10). We no longer obtained values for efficiency greater than 100% and smaller than 99,4%. Around 3:15, when we also detected an overall increase in the average amount of requested packets per request, a 0,5% overall drop in D-Server's efficiency is also observed. There are more moments where the efficiency is not 100% despite being close to 100%. This global statistics attenuates the significance of the underlying information due to the large numbers involved. To deal with this problem we analyzed the information from multiple perspectives: per server, per channel, and per geographical region.

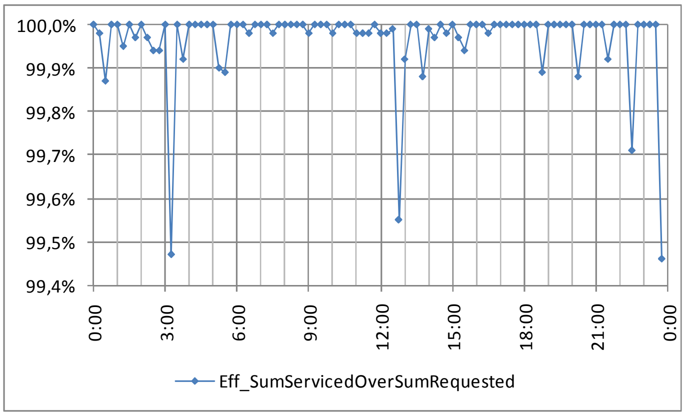

Figure 3-10 D-Server's efficiency in 15 minute aggregates after filtering out records regarding users that do not appear in the Activity Logs

Analysis per server

Initially, we looked at the efficiency ratio grouping the information by server. We verified that there is a correspondence between the global efficiency dropping and some server, or multiple servers, reporting an efficiency drop. Furthermore, where the global efficiency computation results in efficiency drops of merely 0,5% (by 13:00), we see that, from the perspective of the affected servers, the efficiency achieves values of 90%, which is more significant. Additionally, there are other moments in the day where the efficiency of some servers drops below 80% while the global statistics indicated a mere 0,1% efficiency decrease. This result is shown in

Figure 3-11 where we only plotted the data points when the efficiency for a server was not 100%, that is, for some channel or multiple channels, there were requests from one or multiple STBs that were not fully satisfied.

From Figure 3-10 and Figure 3-11 we can see that there is correspondence between the global efficiency dropping and some server, or multiple servers, reporting an efficiency drop.

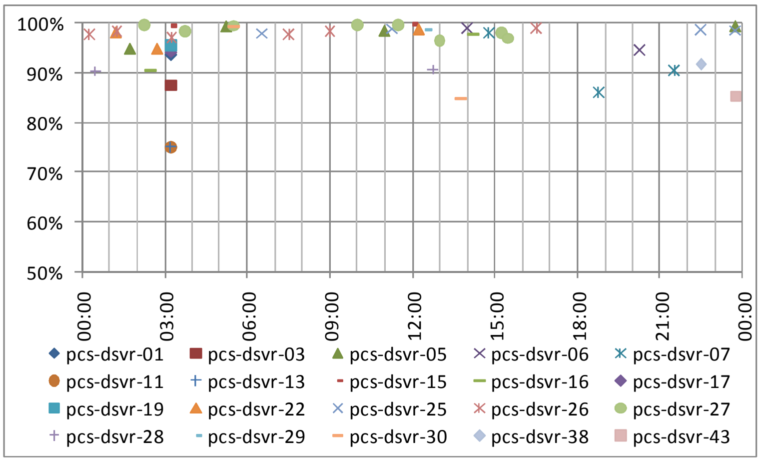

Figure 3-11 Plot of D-Server's efficiency in 15 minutes aggregates, per server, where it is different from 100%

Nevertheless, Figure 3-11 alone does not provide information on the actual importance of the reported efficiency drop. For example, the 25% efficiency reported by server 13 at 3:15 corresponds to 1 packet not served in a total of 4 requested packets in the 15 minutes period. We should thus be able to determine whether a certain efficiency level is relevant considering the amount of data involved. Figure 3-12 represents the D-Server's efficiency in the vertical axe whenever it was different from 100%. Additionally, each bubble’s radius is proportional to the sum of the requested packets in the period. Therefore, the most significant occurrences of efficiency drops are those where a bubble is low in the vertical axe and simultaneously presents a large radius. Some bubbles have very small radius as the one corresponding to the previous example. On the other hand, bubbles such as the one from server 28 at 12:45 representing 91% efficiency in a total of 279 packets are therefore more relevant.

Efficiency problems were detected in 20 out of the 79 analyzed servers, which by itself is already an interesting result. The periods from 11:00 to 13:00 and from 14:00 to 16:00 show higher density of efficiency problems. The later period overlaps the moments where outlier amounts of packet requests were identified in section 3.1.2. The period from 19:00 onward also shows several moments with efficiency around 90%. This can be related with the increasing amount of connected clients during prime time. Nevertheless, given the known increase in the amount of connected clients, it is relevant to notice that the occurrence of efficiency problems is less frequent in the night than during the afternoon.

Furthermore, most of the episodes where the efficiency is not 100% involve a single server. The exceptions are at 3:15, where 8 servers are involved, 23:45 with 3 servers, and 1:15, 5:30, and 22:30 with 2 servers. These facts suggest that the causes for the efficiency problems may be internal to the servers. Nevertheless, the existence of periods where these occurrences are more frequent also indicates that there may be external conditions that facilitate their occurrence in multiple servers.

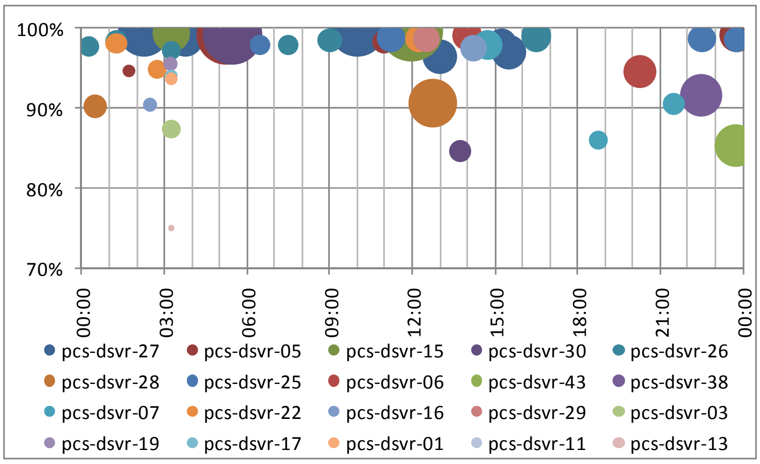

Figure 3-12 Bubble graph representing the D-Server's efficiency in the vertical axe and the bubble's radius being proportional to the sum of the requested packets to that server

Analysis per channel

Another perspective of analyzing this data is in what regards the distribution of the retransmission requests by the channels corresponding to the requests. Such analysis could evidence issues specific to a given channel, such as problems on the multicast group management or on the encoding procedures.

To assess the relative importance of the events that inform of efficiency drops we used a bubble graph where the radius of the bubble is proportional to the requested packets for that channel. Nevertheless, we must take into account the amount of users that are tuned to each channel because it is natural that a more popular channel has larger amount of packet requests without that being necessarily an indication of problems. Therefore, for the bubble radius we used the amount of requested packets per active viewer on the channel. A combination of this value with the server's efficiency per channel provided a metric of how much the lost packets affected the connected users.

As the channel audience information is not contained in these D-Server reports, we queried the Activity Logs of the STBs in order to estimate the amount of users that are connected to a given channel. We used the amount of Channel Tune events that were occurring in a given moment counting the amount of such events that started before and ended after each instant. As we are considering 15 minutes aggregates, we compute this metric using a 15 minutes interval. With this information we were able to determine the average amount of requested packets per channel viewer in the moments where the efficiency was not 100%. The results from this computation are graphically represented in Figure 3-13.

The results show the same distribution of moments where efficiency was not 100%, and likewise with the analysis per server, show that most of the occurrences affect a single channel simultaneously. The only moments where more than 1 channel was affected was at 3:00 (7 channels), 23:45 (3 channels) and 1:15, 3:45, and 22:30 (2 channels). This result matches almost perfectly with the analysis per server, meaning that most of the times the efficiency problems that occur are limited to one channel in one server. In fact, we identified 20 servers with efficiency problems, and we now found 20 channels (out of 294). Nevertheless, since there are periods where these occurrences are more frequent, there may be external conditions that facilitate their occurrence in a channel and server.

Finally, Figure 3-13 also shows a very prominent efficiency decrease to 62,5% concerning channel 234 at 18:45. This occurrence corresponded to an efficiency of 86,1% in server 7, meaning that it was able to process requests regarding other channels.

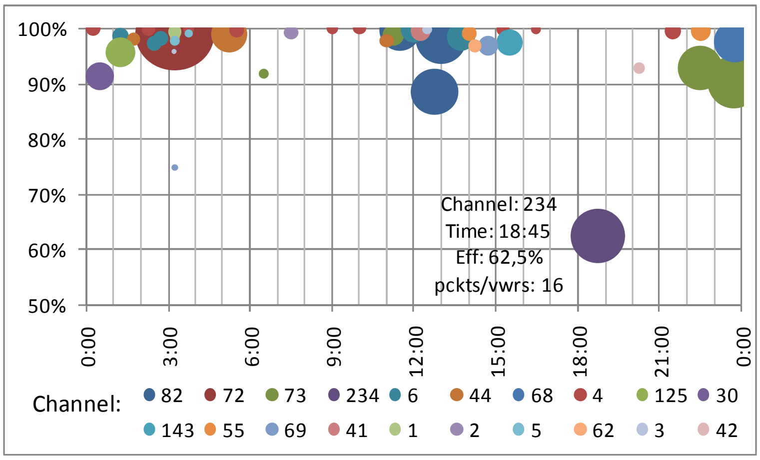

Figure 3-13 Bubble graph representing in the vertical axe the D-Server's efficiency where it is different from 100%, and the bubble's radius being proportional to the average requested packets per active viewer of each channel, in 15 minutes aggregates

Analysis per geographical region

The D-Server's report is also categorized by client IP. In this study, the IP of each STB is static and their geographic region can be looked up. It is possible to determine if the efficiency drops in the D-Servers are correlated with regions by repeating the query but grouping the data by the geographic region associated with each IP. This would suggest that a common problem in that region might have caused the efficiency drop, for example, a problem in the access network or edge router. The events where the efficiency is different of 100% in a region-aggregated perspective are plotted In Figure 3-14. The bubble's radius is proportional to the sum of requested packets related with that region per time slice. The larger the bubble, the more significant is the efficiency drop.

Our results suggested that the 3:15 efficiency drop is related with multiple regions, namely BHO1, PG1, VFX1 and CHE1, but most of the remaining moments are related with single regions. Nevertheless, of the 202 possible regions, only 17 come up in this analysis. We found that region VFX1 and PBL1 are the most frequent ones in this day, and PBL1 displays the most significant efficiency drops, around 12:45, 22:30 and 23:45.

Since there is little diversity in the geographical areas involved in each event, we conclude that most of the times, when the system provides evidence of efficiency problems, it is mainly characterized by a single set of server, channel, and geographical region.

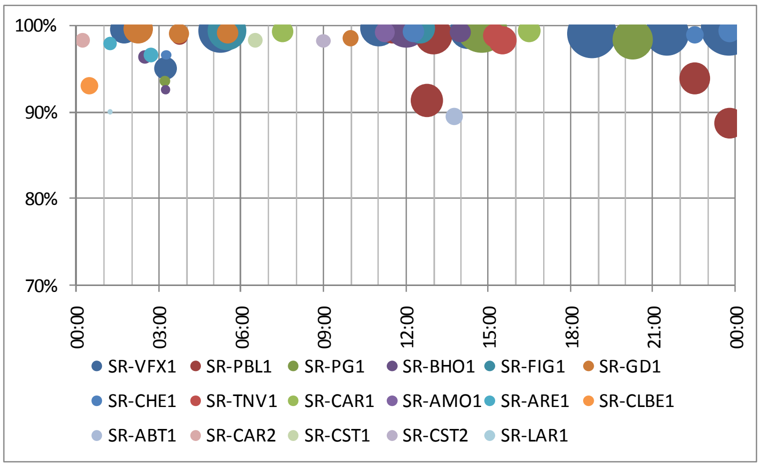

Figure 3-14 Bubble graph representing the D-Server's efficiency per geographical region in the vertical axe, where efficiency is different from 100%, and the bubble's radius being proportional to the sum of the requested packets in that region, in 15 minutes aggregates

Summary

By analyzing the packet retransmission efficiency from multiple perspectives we intended to discover what were the common elements when the efficiency was smaller than 100%. We discovered that most of these occurrences are isolated in terms of timestamp, server, channel and geographical area. The reports that can be aggregated in time and provide diverse information for efficiency drops are listed in Table 3.2.

Group of events	Time	Server	Channel	Area	Req.Packets	5 minute Report entry efficiency
1	20-07-2010 1:21	pcs-dsvr-26	125	SR-LAR1	1	0,0%
1	20-07-2010 1:24	pcs-dsvr-22	6	SR-ARE1	4	75,0%
2	20-07-2010 3:16	pcs-dsvr-13	1	**SR-VFX1**	2	50,0%
2	20-07-2010 3:17	pcs-dsvr-11	5	**SR-VFX1**	1	0,0%
2	20-07-2010 3:18	pcs-dsvr-01	69	**SR-PG1**	1	0,0%
2	20-07-2010 3:19	pcs-dsvr-17	60	**SR-BHO1**	2	50,0%
2	20-07-2010 3:20	**pcs-dsvr-03**	5	**SR-VFX1**	4	75,0%
2	20-07-2010 3:20	**pcs-dsvr-03**	3	**SR-VFX1**	4	25,0%
2	20-07-2010 3:20	**pcs-dsvr-03**	1	**SR-BHO1**	1	0,0%
2	20-07-2010 3:20	pcs-dsvr-19	55	**SR-VFX1**	1	0,0%
2	20-07-2010 3:21	pcs-dsvr-26	44	SR-CHE1	3	66,7%
2	20-07-2010 3:22	pcs-dsvr-15	72	**SR-PG1**	1	0,0%
3	20-07-2010 3:56	**pcs-dsvr-27**	4	SR-GD1	46	97,8%
3	20-07-2010 3:56	**pcs-dsvr-27**	5	SR-PBL1	10	90,0%
4	20-07-2010 5:35	pcs-dsvr-30	**4**	SR-FIG1	11	72,7%
4	20-07-2010 5:36	pcs-dsvr-27	**4**	SR-GD1	34	97,1%
5	20-07-2010 11:29	pcs-dsvr-25	73	SR-AMO1	7	85,7%
5	20-07-2010 11:31	pcs-dsvr-27	82	SR-PBL1	16	93,8%
6	20-07-2010 23:50	pcs-dsvr-05	6	SR-VFX1	19	94,7%
6	20-07-2010 23:59	pcs-dsvr-25	68	SR-CHE1	1	0,0%
6	20-07-2010 23:59	pcs-dsvr-43	73	SR-PBL1	54	42,6%

Table 3.2 Groups of D-Server report entries separated by less than 5 minutes and with efficiency inferior to 100%. The text in bold signals the entities that are repeated within a group of events.

We identified 6 groups of reports separated by less than 5 minutes. 3 of those groups have no common characteristics other than the timestamp being separated by less than 5 minutes.

There is one group with 2 events at 3:56 where the server 27 is the common element. By 5:35 there are 2 reports regarding channel 4. Around 3:20 there is a group with 10 reports where Server 3 appears 3 times, while the geographical area VFX1 appears 5 times, and BHO1 and PG1 appear 2 times, and CHE1 1 time. The commonalities within these 3 groups (server 27, channel 4 and geographical regions) can be an indication of the reason for the corresponding efficiency problem. From the groups without common characteristics, the only conclusion that can be derived is that an external condition to those servers/channels/areas might have been appropriate to cause efficiency problems.

We have thus defined the D-Server's efficiency as the ratio between served packets and requested packets. Although this ratio should not be greater than 1, we identified that when looking to a subset of servers the efficiency can be seen greater than 1. We speculate that this may occur because of requests that are proxyed from other servers (of which the data is not available in this study), for example, due to load balancing mechanisms, which are accounted as served but not as requested. We identified the moments where the efficiency dropped and measured their actual relevance considering the amount of packets involved. We concluded that these events are more frequent between 11:00 and 16:30 and then more sporadic but expressive from 18:45 onward. We investigated the underlying characteristics of the efficiency drops computing the efficiency from the perspective of the server, channel, and geographical area. We verified that most of the occurrences are isolated in terms of these characteristics, suggesting that they may not have been the reason for the efficiency problems. The events that are not isolated and can be groups were analyzed in the beginning of this sub-section.

3.1.4 Information gaps

We detected that there are occasional gaps in the 5 minute pace of the reports produced by the D-Servers. We do not know the cause of these gaps but one likely possibility is that the logging tasks have low priority and are not executed in periods of high workload. Therefore, we established the hypothesis that the occurrence of these gaps is correlated with periods of high workload, which would also be correlated with periods where the server's performance is likely to be degraded. If this is true, identifying these gaps could signal performance problems on a specific server.

We computed the time difference between the statistical information of each server and plotted the points where it was greater than the normal 5 minutes. The results are plotted in Figure 3-15 where each dot represents a moment in time where information is missing from a server. The vertical axe represents the elapsed time since the last known statistical information for that server. We additionally plotted the number of gaps per hour depending on the gap size. These gaps occur systematically throughout the day. The 10 minutes gaps, which correspond to 1 missed report, appears to have an increasing trend until 19:00. The 15 minute gap also appears to have an increasing trend but only until 16:00. Wider gaps only occur in some periods of the day. There is also one outstanding peak around 16:00 for one server (server 51) which was 50 minutes without producing statistics (that is from 15:10 to 16:00).

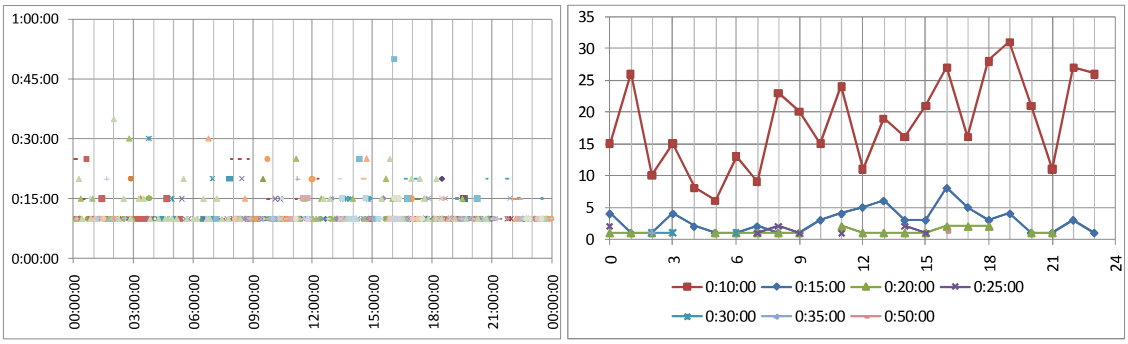

Figure 3-15 Value of the gap (vertical axe) between each server's report, greater than or equal to 10 minutes (left). Number of gaps per hour depending on the gap size (right)

There is no clear correlation with the previously analyzed information. The causes for these gaps can be internal to the server and should be further analyzed.

3.2 VOD Servers

The Video On Demand Servers (VOD-Servers) are media servers that provide the video stream of VOD content which has been previously purchased by the user and still has a valid expiration date. These servers are the main components of the VOD Delivery module depicted in Figure 2-1. The communication between the VOD-Servers and the STBs is supported on HTTP. The available logs belong to the web services that are exposed to the STBs and are in the W3C Extended log format (see section 0).

A first analysis of the logs reveals that the overwhelming majority of network traffic is related to the HTTP GET of the resource "/Media/" (Figure 3-16).

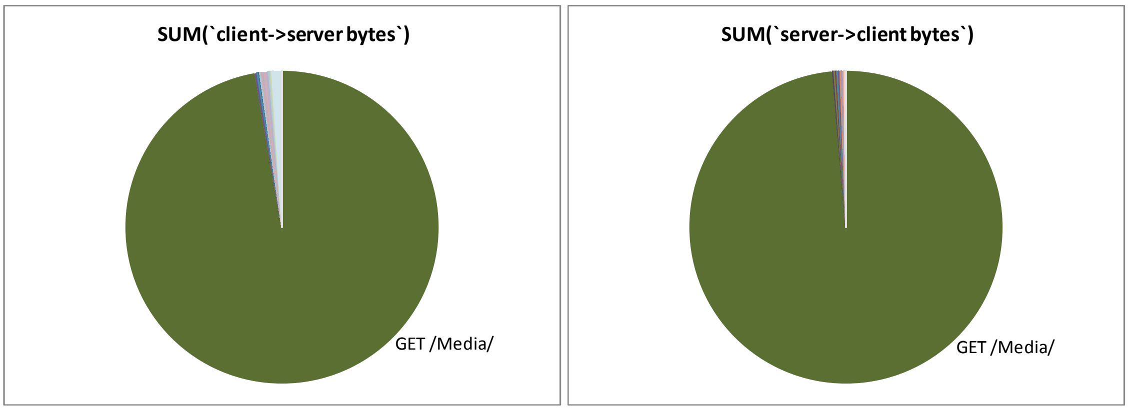

Figure 3-16 Relative weight of each web service in what concerns the traffic sent from client to server (left) and the traffic sent from server to the client (right)

We analyzed the HTTP status code of the "GET /Media/" requests and despite the majority of status 200 (which means "OK"), there is a small amount of status 404 ("not found") and a residual amount of status 500 ("server internal error"). These results and their daily distribution on July 20th are depicted in Figure 3-17.

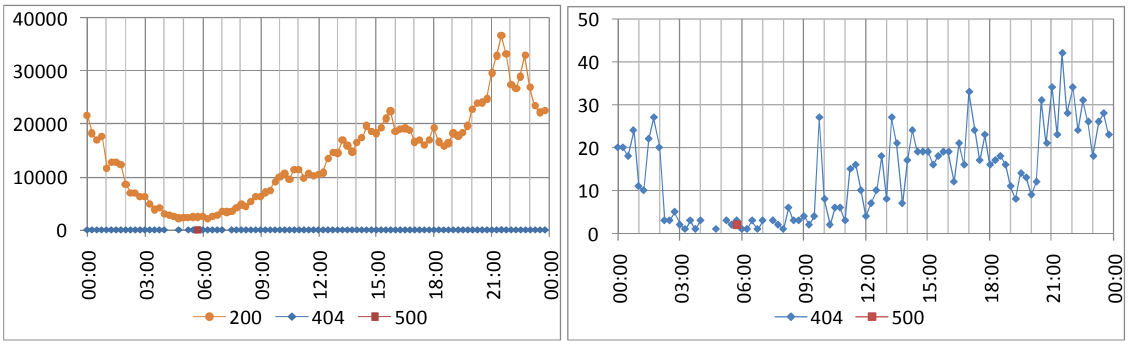

Figure 3-17 Count of HTTP GET /Media/ grouped by the resulting HTTP status codes in 15 minutes aggregates. Status 200 means "OK", 404 is "Not found", and 500 is "Server internal error"

The amount of requests with status 200 follows a daily distribution that is coherent with a daily distribution of a TV audience: low value during dawn, increasing during the morning, stabilizing or slightly decreasing after lunch, then increasing in the afternoon as people return home, and peaking after dinner which is the prime time. There are very few occurrences of HTTP error conditions.

Looking in detail to the logs, there are multiple HTTP requests per client with very little time difference between each other and with large amounts of transferred data, meaning that the process of streaming video over HTTP involves multiple HTTP requests, and not just one very large request. To better know the audience of VOD content, we should use the amount of distinct clients that are connected at each time (Figure 3-18).

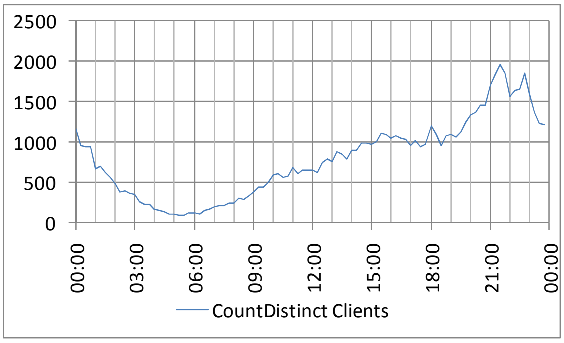

Figure 3-18 Count of distinct clients logged on the VOD Servers in 15 minutes aggregates

Grouping per server the daily distribution of connected clients, we confirm that there are more popular servers than others. Some servers are expected to hold the most popular media contents, thus receiving more requests and more distinct client connections. These servers should be as geographically disperse as possible, reducing the distance to the STB and alleviating the network backbone. From Figure 3-19, 2 groups of servers with very distinct load profile can be differentiated. Servers 5, 6, 7, 8, 9, and 12 (left figure) serve significantly more distinct users than the remaining servers suggesting that these servers hold the most popular media contents. With this information it should be possible to determine whether the VOD asset allocation is correctly distributed through the VOD Server architecture.

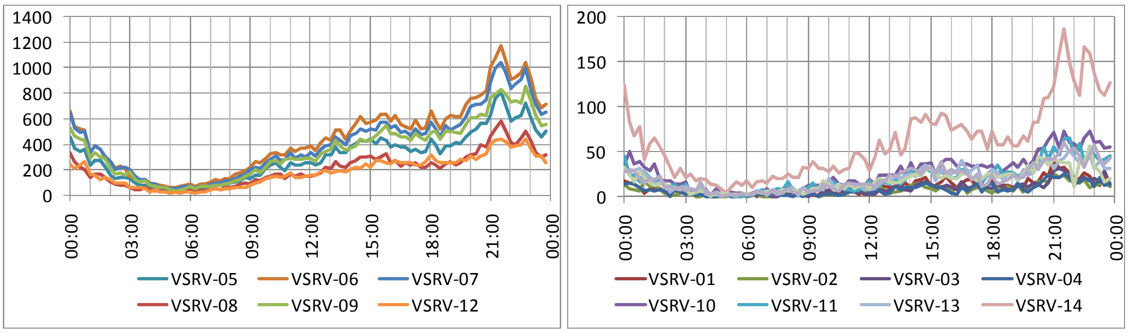

Figure 3-19 Count of distinct clients that were served by each server in each time period, in 15 minutes aggregates. The most popular servers are on the left chart, and lest popular servers are on chart on the right

Taking advantage of the information regarding the time taken by the VOD Server to reply to the HTTP requests and the amount of bytes transferred from server to client, we can determine the average bandwidth of the HTTP responses. We computed and plotted these values in Figure 3-20 separating the previously identified popular servers from the remaining. In average, the download bandwidth is similar between all servers. Only server 8 shows a slightly irregular pattern with less average bandwidth then the remaining servers. The less popular servers show larger variation of the download bandwidth because there is less statistical information.

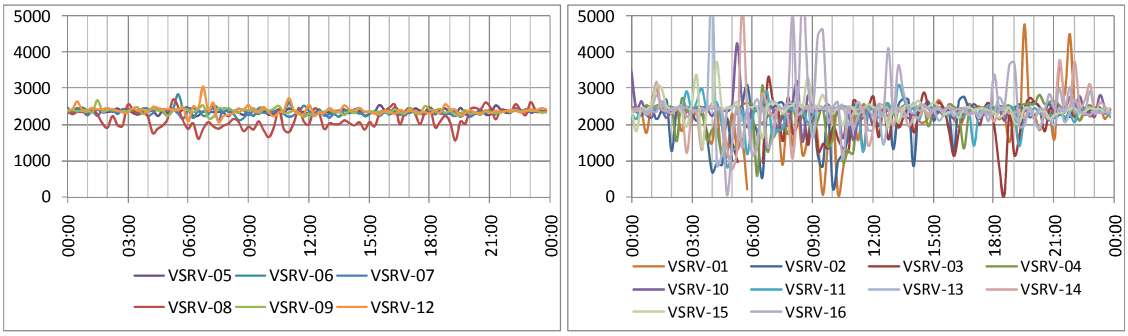

Figure 3-20 Average bandwidth (kbit/s) per server in 15 minutes aggregates. The most popular servers are on the left chart, and lest popular servers are on chart on the right.

The amount of download traffic from each server also reflects the existence of 2 different groups of servers, as can be verified in Figure 3-21. Nevertheless, server 8 now belongs to the group with less traffic despite belonging to the group with more distinct client connections. This server sustains as many client connections as the most popular servers but serves much less download traffic. The configuration of this server should be verified because it may be inadvertently configured as primary server of too many STBs.

These patterns show that, between the more popular servers, the load is evenly shared, because the daily distribution of download traffic and connected clients is identical. On the other hand, the less popular servers are more, and appear to have significantly less workload. Nevertheless, there is no evidence of performance issues such as HTTP error codes. These results suggest that the less popular servers are significantly under loaded.

It should also be mentioned that the total daily download traffic regarding VOD totals approximately 9,3 Tb. This volume regards VOD purchases as well as free content, such as trailers. This volume as well as the load distribution between servers should be monitored as the VOD content and user's adherence is expected to increase in the future.

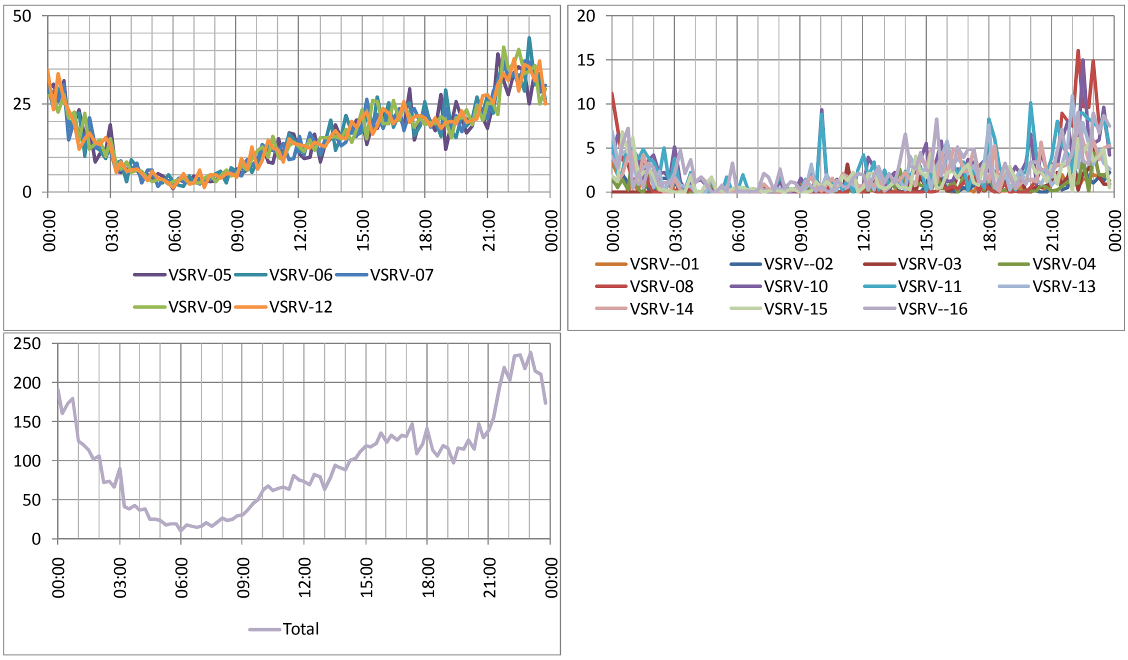

Figure 3-21 Download traffic (GB) per server in 15 minutes aggregates. The most popular servers are on the top left chart, the less popular servers are on the top right, and the total download traffic is on the bottom left.

3.3 Activity Logs

The STBs have the capability of keeping track of some user actions logging them as activity events, for example when a user changes the channel. Alongside with explicit user actions, the activity logs also contain events triggered by the STB itself, for example, when a scheduled recording automatically starts, or when the Electronic Program Guide (EPG) of the tuned channel indicates the ending of the program being watched. The typical purpose of this kind of information is to enable service providers to better understand which and how services and features are being used by subscribers. This information can prove useful for a variety of purposes such as analysis of marketing campaigns or determining advertising revenues.

We are interested in the identification and analysis of the information that represents explicit user actions and simultaneously contributes significantly to the IPTV platform’s critical workload. For example, we will look into events that indicate that the user tunes to a new channel because this activity involves multiple orchestrated actions inside the platform, namely a unicast video burst triggered by the Distribution Servers (section 3.1).

Each STB uploads its activity logs to a central database whenever a specified amount of locally stored events is reached or when a specified time period has elapsed since the last upload. If a box is turned off by the user before one of these thresholds is reached, those logs are not uploaded to the central database until the STB is reconnected. The central database does not have the capacity to store all events for multiple days so it only stores the information received within roughly the last 4 hours. Nevertheless, this information can report to several days back as in the case where a box is shut down for several days with some events locally stored.

3.3.1 Data structure

Each entry of the activity log contains 4 main fields:

- clientID: client identifier
- envetID: event identifier
- originTime: timestamp
- data: XML structure containing the attributes for the correspondent eventID

The “data” field content is, therefore, dependant of the type of event, for example, the event 101 (Box Power) will have in the “data” field a XML structure with the attributes "ClientType", "PowerState", "BuildFlavor", and "BuildNumber" as depicted in Figure 3-22.


```
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx; 101; 2010-05-27 18:29:53.193; "<d><nv
n=""ClientType"" v=""MOTOROLAVIP1200E_CE"" /><nv n=""PowerState"" v=""ON"" /><nv
n=""BuildFlavor"" v=""RELEASE"" /><nv n=""BuildNumber"" v=""25310"" /></d>"
```


Figure 3-22 Sample of an Activity Log entry

We developed a parsing script to read the Activity Logs and prepare a text file to import into the MySQL database. The script assumes that the amount of distinct fields is dynamic and unknown because from the sample we may not observe all possible different fields. This is relevant because not all documented events were observed, and in fact one undocumented event was detected. It is also worth considering that as the platform evolves new events may appear and should be accounted for.

The strategy of the parsing script is to learn the different attributes as they occur while simultaneously indexing them. The index is used to find the position in an array where the correspondent value is stored. The array is then dumped to a file and imported to the database.

3.3.2 Data set overview

The available activity log dataset comprises 3 snapshots of the database, exported roughly on July 20th 2010 11:00, July 21st 2010 11:00, and July 21st 2010 15:00.

The Activity Log data set contains a total of 2954486 records (events) and 12791 distinct STBs. This corresponds to just 1 of the 11 Service Groups of the whole IPTV platform. A Service Group is a logical aggregation of Clients and is created for scalability as the overall amount of Clients increases.

The distribution of event occurrence is described in Table 3.3. Nearly 90% of the events are of just five different event IDs: Stream Management Stream Request (17785), Browse Panel (105), Program Watched (114), Application (106), and Trick State (104). If we also consider the Channel Tune (100) and Box Power (101) events they represent 97% of all logged events.

EventID	Event Description	Count	%
17785	Stream Management Stream Request	1248190	42,25%
105	Browse Panel	424923	14,38%
114	Program Watched	345594	11,70%
106	Application	336568	11,39%
104	Trick State	299887	10,15%
100	Channel Tune	163371	5,53%
101	Box Power	42597	1,44%
115	DVR Start Recording	16008	0,54%
117	DVR Playback Recording	15837	0,54%
107	Menu Selection	15734	0,53%
138	unknown	14848	0,50%
119	DVR Delete Recording	14791	0,50%
118	DVR Schedule Recording	5222	0,18%
17784	Stream Management Contention	5022	0,17%
116	DVR Abort Recording	1288	0,04%
120	DVR Cancel Recording	1016	0,03%
17781	Stream Management Detune	934	0,03%
17786	Stream Management TV Interrupt	934	0,03%
17783	Stream Management Reboot	883	0,03%
109	RDP Application Disconnect	280	0,01%
110	RDP Application Navigate Away	199	0,01%
108	RDP Application Launch	153	0,01%
111	RDP Application MCE Tune	117	0,00%
102	VDP Purchase	90	0,00%

Table 3.3 Amount of occurrences of each event in the sample

The event ID 138 is not described in the available documentation.

The elapsed time period covered between the first and last available log entry is from July 8th 2010 15:16:38 to July 21st 2010 13:00:09, that is nearly 13 days. Nevertheless, for the reasons previously explained (see section 3.3), the events are not evenly distributed.

Each STB transfers the Activity Logs to the central database in one of two situations: (i) when a specified amount of locally stored events is reached or (ii) when a specified time has elapsed since the last event transfer. Figure 3-23 and Figure 3-24 show the distribution of each STB's first and last event, respectively, in 15 minute aggregates. The chart on the right zooms the detail from July 19th to July 21st. We can observe that the distribution of the first available event of each box is more disperse than their last event. This occurs because when a STB is powered off it keeps its events locally stored. If the box remains powered off longer than the referred time threshold, it will transfer the events as soon as it is powered up again. When that happens, it will communicate events that were locally stored which may refer to several hours or days back. We also observe in Figure 3-24 that there is a large amount of STB's for which their last available event is near 20:30 of July 20th. This is important because we will be looking at the data concerning this particular day. In Figure 3-25 and Figure 3-26 each dot represents a distinct STB, the vertical axe represents the amount of events available for each box (in logarithmic scale for clearer representation) and the horizontal axe represents the first (Figure 3-25) and last (Figure 3-26) available timestamp for each box. In the vertical axe, the concentration of points around the 500-multiple events is due to the value that is configured in the boxes as threshold to upload the event logs.

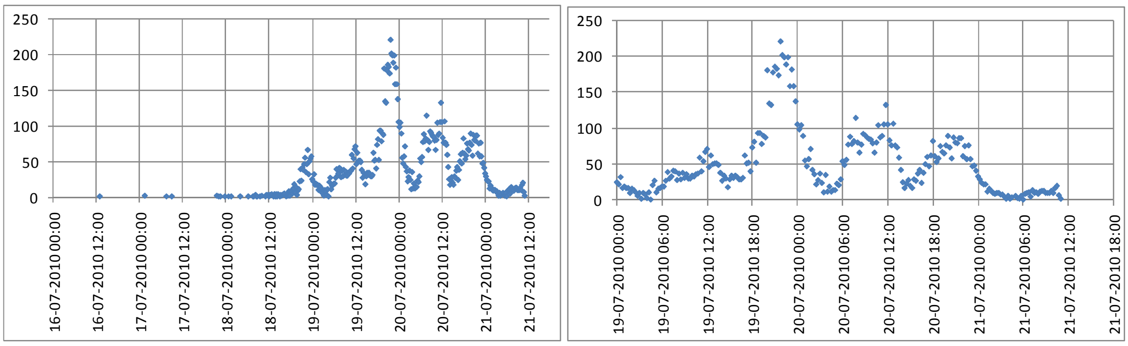

Figure 3-23 Count of first STB events in 15 minute aggregates

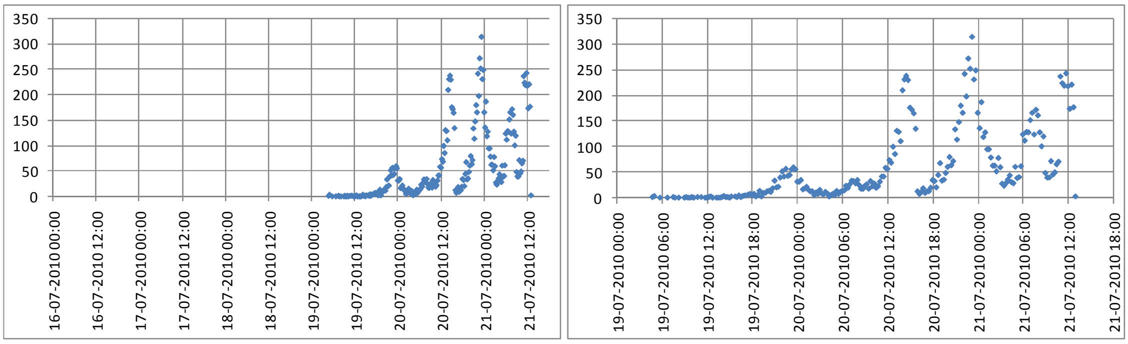

Figure 3-24 Count of last STB events in 15 minute aggregates

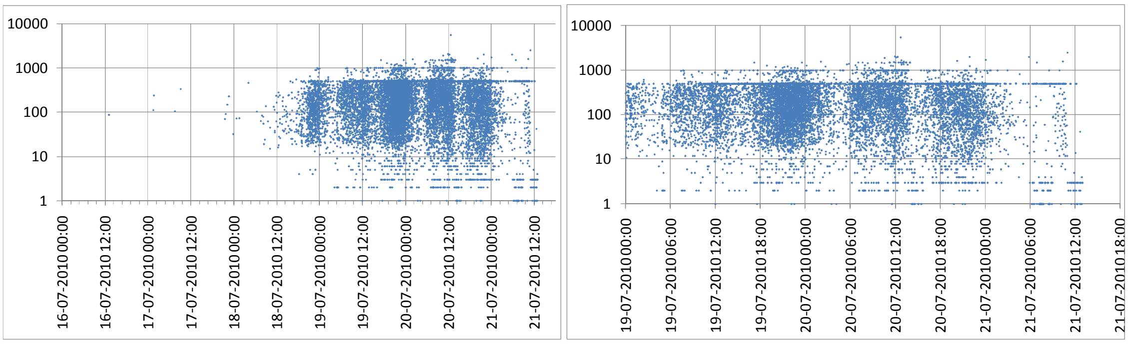

Figure 3-25 Amount of events per STB's, plotted on the moment of the STB's first event, full (left) and detail (right)

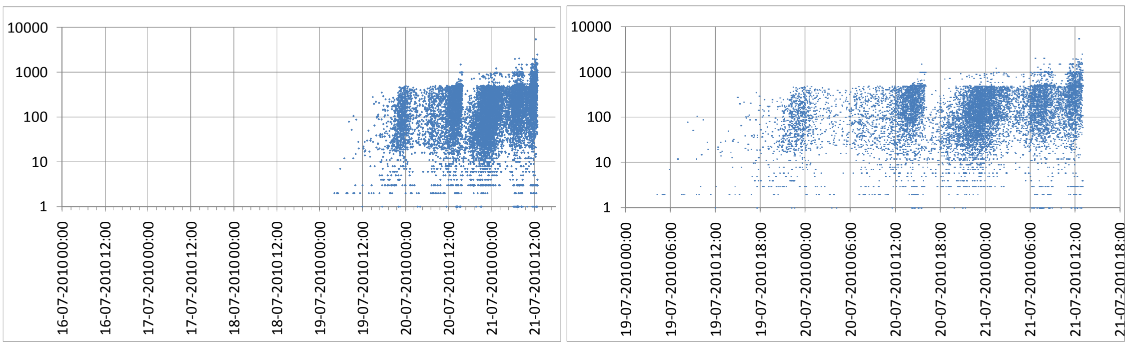

Figure 3-26 Amount of events per STB's, plotted on the moment of the STB's last event, full (left) and detail (right)

As the amount of STBs that reports their first or last event increases in Figure 3-23 and Figure 3-24, the density of dots in Figure 3-25 and Figure 3-26 also increases. The peaks in Figure 3-24 represent moments beyond which we no longer have information regarding a significant amount of boxes. The occurrence of these peaks suggests that these are periods of increased activity where multiple STBs accumulate enough Activity Logs to trigger their transfer.

In our data, the peak in Figure 3-23 (distribution of STB's first event on the log) that occurs before 0:00 of July 20th suggests that that the log contains information regarding a significant amount of STBs from then onward. In Figure 3-24, (distribution of STB's last event on the log), we see that the second peak is before the end of July 20th, suggesting that the amount of available information decreases from then onward. Nevertheless, the first peak in Figure 3-24, which occurs roughly between 13:00 and 15:30 implies that there is a significant amount of STBs for which we stop having information. This occurs because by the time the database snapshots were taken, these STBs had not yet uploaded their most recent Activity Logs.

The described characteristic suggests that an analysis of a single day may be biased due to this asymmetry. In these logs, before 15:30 of July 20th there is information regarding some amount of STBs, and there is a presumably arbitrary subset of these STBs for which we stop having information, not because they stopped their activity but because they were locally retaining their logs by the time the database snapshots were taken. Furthermore, the second peak in Figure 3-24 occurs around 21:00 also suggesting some bias from this moment onward.

Figure 3-27 represents the amount of STBs that are active at each point in time, defining as "active" a STB for which we have log entries before and after each point in time. From this chart we can see that the most significant activity period is the full day of July 20th, always containing more than 5000 active boxes. Furthermore we can visualize the effect of the significant amount of STBs for which we stop having information between 13:00 and 15:30 (roughly 2000), which corresponds to the sum of the points in Figure 3-24.

It is also noticeable that the amount of active STBs decreases again from 21:00 onward confirming the previous analysis.

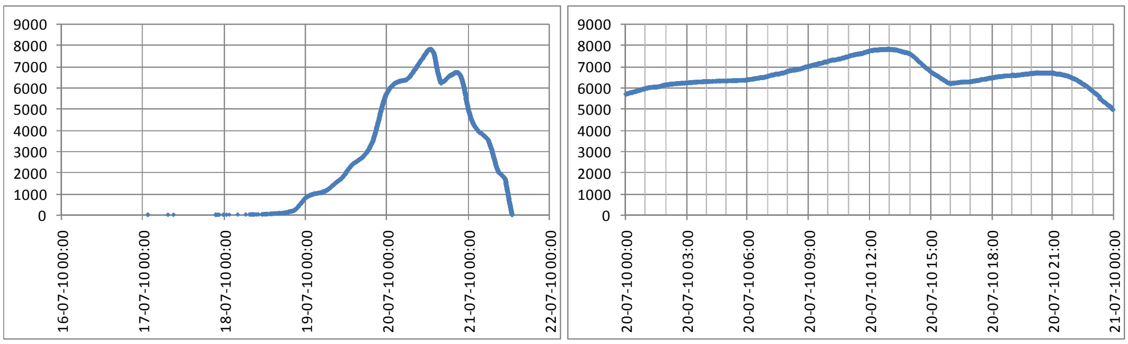

Figure 3-27 Distribution of active boxes, full (left) and detail for July 20th

We conclude that the best full day period for analysis is July 20th as most of the STBs have some record in this period. From the 12791 distinct STBs, 10746 (84%) reported events during July 20th. There are some characteristics from the dataset which we need to be aware of, such as the fact that many STBs have their last reported event occurring almost simultaneously in two moments of that day. This does not occur because the STBs or their users changed their behavior, but because the STBs had not yet uploaded the Activity Logs by the time the database snapshots were taken.

3.3.3 Events representing user behavior

Not all types of events recorded by the STBs in the Activity Logs directly reflect user actions. Some events are created by the STB and other by Management mechanisms. Therefore it is first necessary to study and identify the meaning of the events so that we can focus on those that directly represent or are a direct consequence of user actions. Next we need to make a qualitative appreciation of the contribution of each user action to the platform's workload. As mentioned in section Chapter 2 , we are interested in identifying sources of unicast traffic, such as those involved in channel changes, packet retransmissions or VOD, while at the same time taking in consideration the size of the multicast groups, that is, the global audience. We studied all the events and present in Table 3.4 a justified summary of the ones that fit in these categories. In the Event Description column some events are already desegregated by an attribute representing the "stream type", which is import to differentiate between events related with Live TV, VOD or DVR.

Event ID	Event Description	Originator	Network impact	%	Notes
105	Browse Panel	User	Unicast starts (channel browse, PIP)	14,00%	User browses through PIP version of Live channels
100	Channel Tune.LIVE	User	Unicast + Multicast starts	5,20%	User tunes to a Live channel. ICC occurs: unicast burst plus multicast join.
100	Channel Tune.VOD	User	Unicast starts	0,40%	User tunes to VOD channel
104	Trick State.VOD	User	Unicast pauses, resumes, etc	0,30%	User acts on VOD stream (pause, play, rewind, etc)
17785	Stream Management Stream Request.VOD	STB	Unicast is ongoing	0,10%	STB requests while user is tuned on a VOD channel
17781	Stream Management Detune.VOD	User	Unicast stops	<0,01%	User tunes away from stream. Has small statistical relevance
17785	Stream Management Stream Request.LIVE	STB	Multicast is ongoing	42,50%	STB requests while user is tuned on a Live channel
104	Trick State.NEWLIVE	User	Multicast is ongoing	2,00%	User acts on Live stream (pause, play, rewind, etc) while STB continues connected to the multicast group
115	DVR Start Recording.ByUser	User	Multicast starts/continues	<0,01%	STB joins multicast if not already connected (Live). ICC does not occur.
115	DVR Start Recording.BySystem	STB	Multicast starts/continues	0,60%	STB joins multicast if not already connected (Live). ICC does not occur.
17781	Stream Management Detune.LIVE	User	Multicast stops	0,03%	User tunes away from stream. Has small statistical relevance

Table 3.4 Summary of the most relevant Activity Log events

The events that are considered are the *Channel Tune, Browse Panel*, *Trick State*, *Stream Management Stream Request, Stream Management Detune, and DVR Start Recording*.

The attributes of the Channel Tune event allow identifying if it regards the tuning of a Live TV channel (and which), a VOD content or a (local) DVR recording. Only the first two imply new video data streams on the network. If the event regards a Live TV channel, then the ICC procedure has occurred before the STB joins the multicast group for that channel. If the event regards a VOD channel, then the whole stream is unicast. The Channel Tune event also contains the duration of channel tune, providing a way to compute the global adherence to a channel, that is, to the multicast group of the channel. Nevertheless, this event is only logged by the STB if the user stays tuned in the same channel more than 60 seconds.

The Browse Panel event occurs as the user navigates between the PIP versions of the Live channels in the GUI's browse panel. This information is very important as it shows the user actively scrolling through channels, triggering requests to the Distribution Servers and expecting an immediate response. The details regarding this event contain only the media descriptor, allowing the identification of the browsed channel, and the button pressed (up or down).

The Stream Management Stream Request event allows differentiating requests for Live stream, from VOD and DVR playbacks. As mentioned, Live streams are provided in multicast and VOD in unicast. In what regards DVR, if the household only has one STB, a DVR playback is actually a local video stream request made from the STB to itself. In households with multiple STBs, the DRV content is stored in one of the STBs acting as a media server. Therefore, some stream requests regarding DVR content are actually stream requests made from one STB to another within the household local network.

The Trick State event reflects user actions (e.g., pause, rewind, play) over video content. When a Trick State action is performed within the context of a live TV channel or DVR, the action is in fact being performed over video content that has already been locally stored in the STB or in the household's STB with storage capabilities. Therefore, Trick State events are expected to be closely associated with DVR content but should not be correlated with performance issues of the broadcasting platform as it does not introduce additional requests to the broadcasting systems. The exception is the Trick State event regarding VOD which has a direct influence on the unicast connection with the VOD Server.

The DVR Start Recording event indicates that the STB has started recording some channel. This means that if the STB was not already tuned to the channel, it performs a new join to the correspondent multicast group. In this situation the STB does not perform ICC because the group join is being performed in the background. Therefore, when this event occurs there are no new unicast connections, just a new join to the multicast group. This event can be triggered by the platform's scheduling system, if a schedule has been previously defined or on demand by the user through the GUI.

3.3.4 Distribution of events

The daily distribution of the different types of events is represented in Figure 3-28. The events occur in different orders of magnitude, so to plot them in the same chart we used a logarithmic vertical axe. From Figure 3-28 we can immediately verify that the statistically most significant events are the Stream Management Stream Request, Browse Panel, Trick State, Application, Channel Tune, Program Watched, and Box Power, which account for 97% of all events (Table 3.3). We observe that most events have similar daily patterns. There are two peak periods that occur roughly between 13:00 and 15:00 and between 21:00 and 23:00. This is perceivable in Figure 3-29 where the 5 minute aggregate count of events was plotted in different charts with different vertical scales for better legibility.

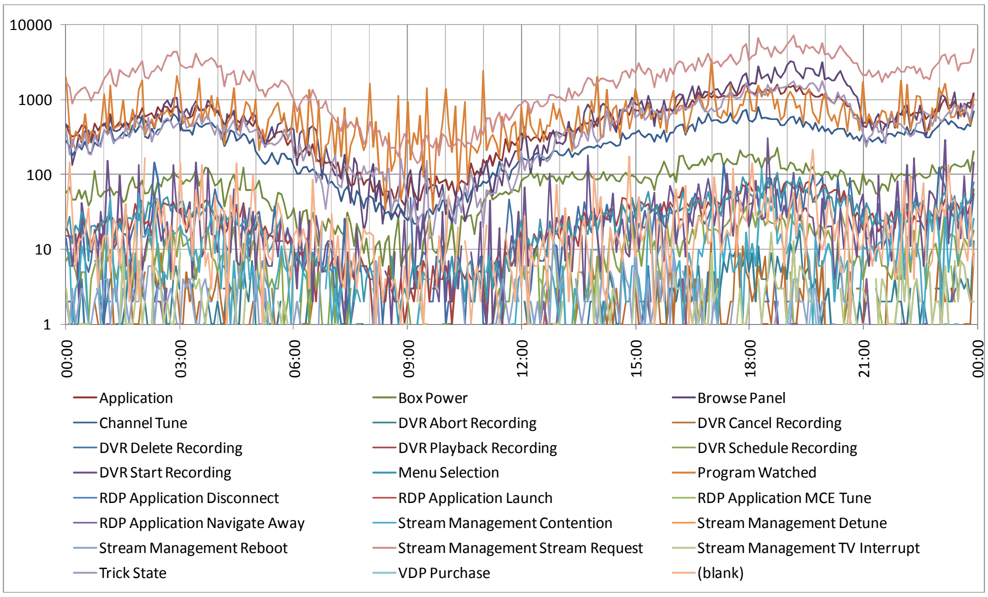

Figure 3-28 Daily distribution of events (5 minute aggregate with vertical logarithmic scale)

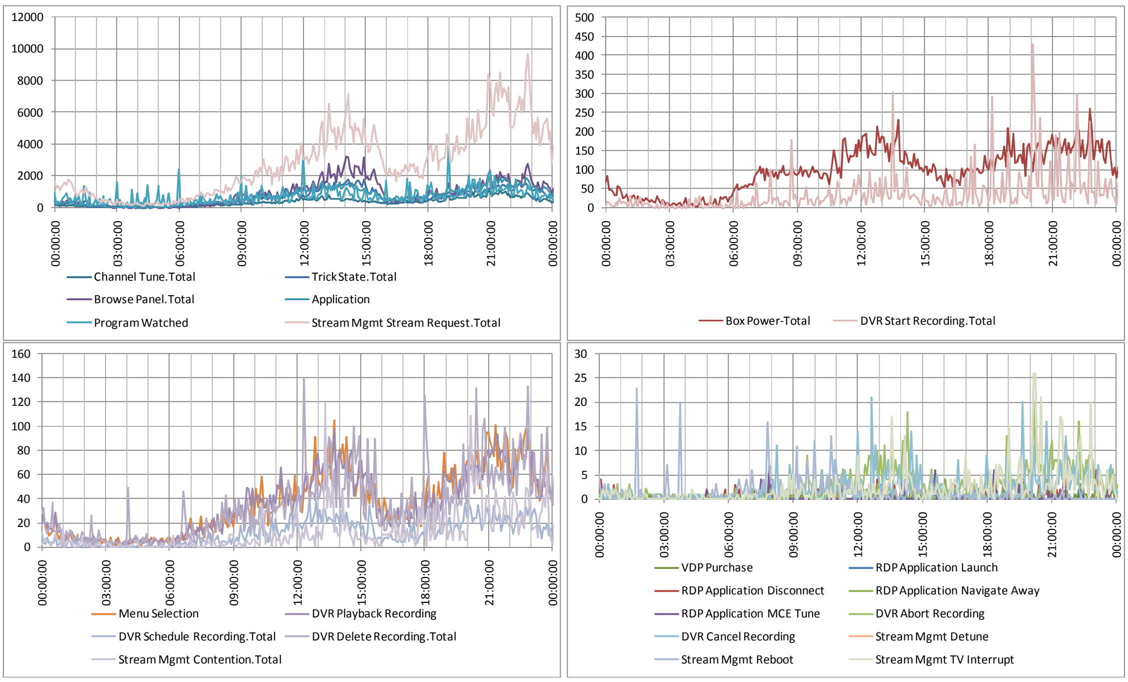

Figure 3-29 Daily distribution of events. From left to right and top to bottom, the represented events occur with decreasing frequently, and are plotted in different scales for better legibility

3.3.5 Channel Tune

The Channel Tune event represents the user's adherence to a specific channel. It is triggered when a user tunes to a channel and keeps connected to it for more than a minimum amount of time (1 minute in the studies environment). Therefore it represents one of the main user activities in a live TV distribution system and one of the system's activities that we are interesting in studying, the ICC. Additionally, because its details also contains the duration of the tune, it additionally allows estimating each channel's audience ratings.

The specific attributes in the XML "data" field in this event are described in Table 3.5.

Event Fields	Meaning	Values
MediaDesc	ID of the media content to which the event refers to	n/a
ChannelNumber	Number of the channel tuned	n/a
Duration	Duration of the channel tune in milliseconds.	n/a
IsTunnedToService	Indicates if channel was successful tuned to.	True, False
StreamSelection	Type of stream to which the user tuned (full screen or PIP) and service type (primary or secondary) of the media session	FULLSCREEN, PIP, FULLSCREEN_PRIMARY, FULLSCREEN_SECONDARY, PIP_PRIMARY, PIP_SECONDARY
ChannelType	Type of channel to which the user tuned.	DVRAPPMEDIACHANNEL, LIVETVMEDIACHANNEL, NEWLIVE, RDPAPPMEDIACHANNEL, RDPMEDIACHANNEL, VODAPPMEDIACHANNEL
TuneID	Link to "Program Watched" event	n/a

Table 3.5 Attributes of the Channel Tune event

Relation with Program Watched event

We started by comparing the Channel Tune event with the Program Watched event because they are linked. The Program Watch event works as a time mark identifying changes of the current program being broadcasted by a channel. Therefore, multiple Program Watched events are expected to occur during the same Channel Tune as the user stays tuned during multiple programs of the same channel. Furthermore, one such event is expected to occur almost simultaneously across all the STBs tuned on the same channel. Consequently, peaks of this event are expected to occur in specific moments in time.

This result is verified in Figure 3-30 as the Program Watched event always occurs in larger number than the Channel Tune. Since we are not interested in querying the database to find the audience of a specific program, the Program Watched event does not bring additional relevant information. Its daily trend is the same as the one of the Channel Tune event, and its peaks, which are proportional to a specific TV program's audience, are also proportional to the correspondent channel's audience, which can be estimated from the Channel Tune event and its duration. Therefore, for the purpose of this study, the Program Watch event, which accounts for 11,7% of all events, can be disregarded reducing the overall amount of analyzed data.

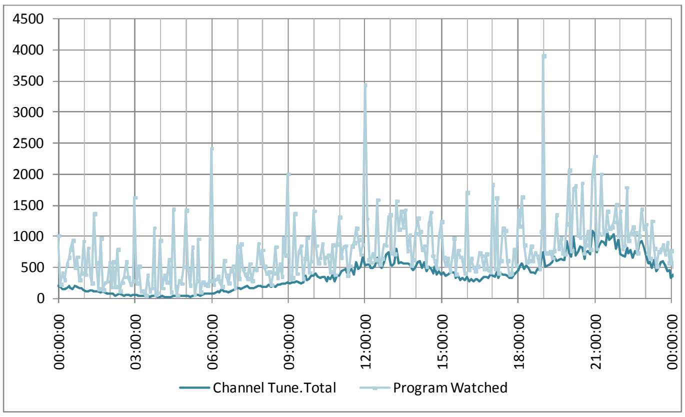

Figure 3-30 Comparison between Channel Tune and Program watched events. As example of two events with similar daily distribution but significantly different variance (quantities for every 5 minutes)

Daily distribution

Despite the daily pattern of the Channel Tune event, its average duration tends to a value of around 40 minutes as the day progresses. The Channel Tunes that start during the dawn of July 20th have very large values suggesting that those STBs are inadvertently left with the power on. As expected, the amount of events during dawn is also very small suggesting a much lower usage. The probability distribution of the duration of the Channel Tune event shows that 50% of the events are shorter than 8 minutes and 90% are shorter than 1,5 hours. During the day, the global usage is characterized by two daily peaks around 13:00 and between 20:00 and 22:00 (see Figure 3-31 and Figure 3-32).

We also observe that the most significant amount of Channel Tune events regards Live channels. In Figure 3-31 the events regarding Live channels are plotted using the secondary vertical axe on the right which is more than one order of magnitude higher. While in peak period the Live channels account for about 1000 events every 5 minutes, DVR channels account for nearly 80, and VOD channels for less than 20. Nevertheless, the daily pattern is roughly similar.

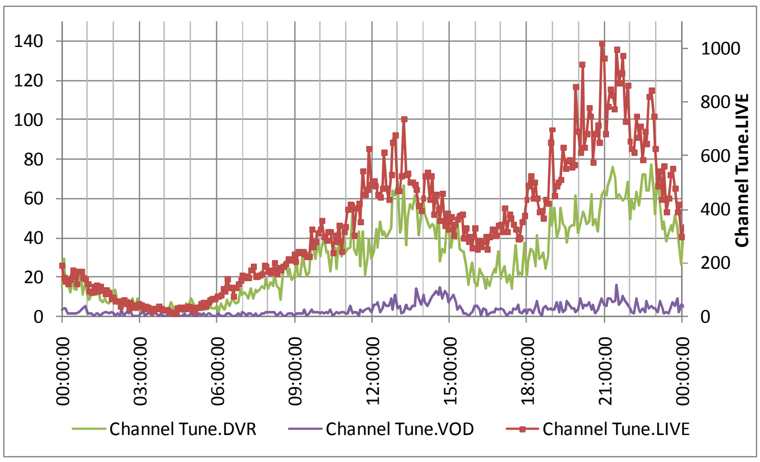

Figure 3-31 Channel Tune event distribution breakdown by channel type, in 5 minutes aggregates

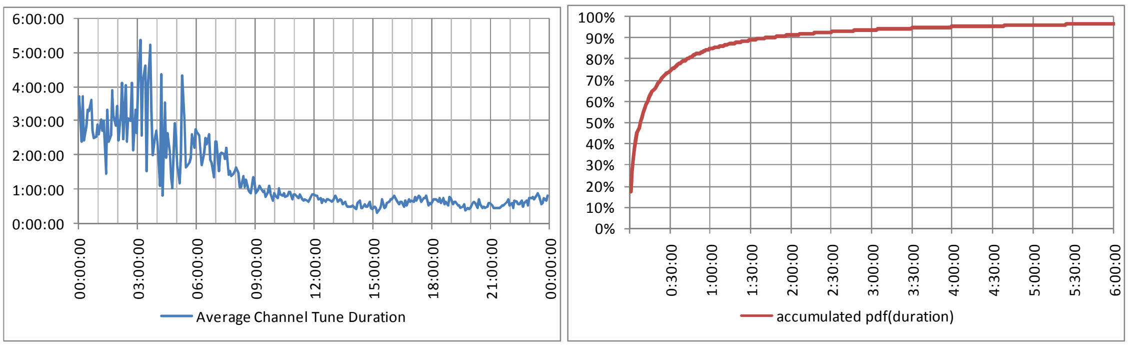

Figure 3-32 Average Channel Tune (Live events) duration in 5 minutes aggregates (left), and cumulative probability distribution of the duration of the Channel Tune (Live and VOD) event (right)

User channel change frequency

In this section we analyze the Channel Tune considering that it means that the STB has concluded a channel change and went through the ICC process described in section 2.3. Nevertheless, we also know that a Channel Tune event is only created if the user stays tuned in the same channel more than 1 minute. This means that this event disregards the channel changes that occur with less than 1 minute apart. Nevertheless, we investigate the time separation between Channel Tune events trying to find how the logged channel changes are distributed throughout the day. For this purpose we computed the distance between each Channel Tune event of each STB during each power on session. We use the Box Power event (ID 101) and its Power State attribute (On/Off) to assign a new "power session" identifier to the events on the STB that are subsequent to a Power On event. This allows us to compute the distance between Channel Tune events inside the same "power session" and ignore the distances that are caused because the user has turned off the STB.

The resulting distribution using 1 minute resolution is represented in Figure 3-33. The mode of the distance between Channel Tune events of each STB is in the range of 1 to 2 minutes with 12,1% of the occurrences. Since the pdf increases exponentially as the distance tends to 0, and since we know that the system does not log channel tunes separated by less than 1 minute apart, it is likely that a significant amount of not logged channel tunes exist and are separated by less than 1 minute from other Channel Tune events but are not logged by the STB.

What we then tried to understand is the distribution of the consecutive Channel Tune events of each STB, and how simultaneous they occur across all STBs. Despite the most frequent distance (1 minute), the cumulative probability increases relatively slowly representing that the average distance between Channel Tune events is much greater than 1 minute (chart on the right of Figure 3-33). Nevertheless, since a significant amount of events is separated by less than 2 minutes, we re-compute the distance between these events and assign a distinct "cluster" IDs to Channel Tune events of the same STB and the same "power session" that are separated by less than 2 minutes.

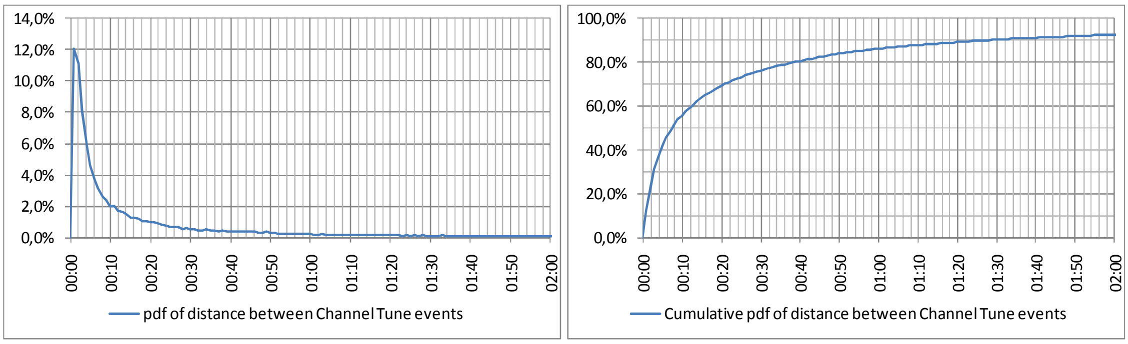

Figure 3-33 Probability distribution function (left) and cumulative pdf (right) of the distance between Live Channel Tune events of the each STB during the same power on session

The average size of the clusters is small and very constant throughout the day, approximately 2,1 events per cluster (Figure 3-34) when disregarding isolated Channel Tune events. There is just one peak during dawn, but this is a period of small statistical importance. This means that the cluster distribution throughout the day follows the same daily pattern as the distribution of the Channel Tune Event which can be graphically estimated in Figure 3-34. The computation of the ration between these two metrics also confirmed that the relationship is approximately linear.

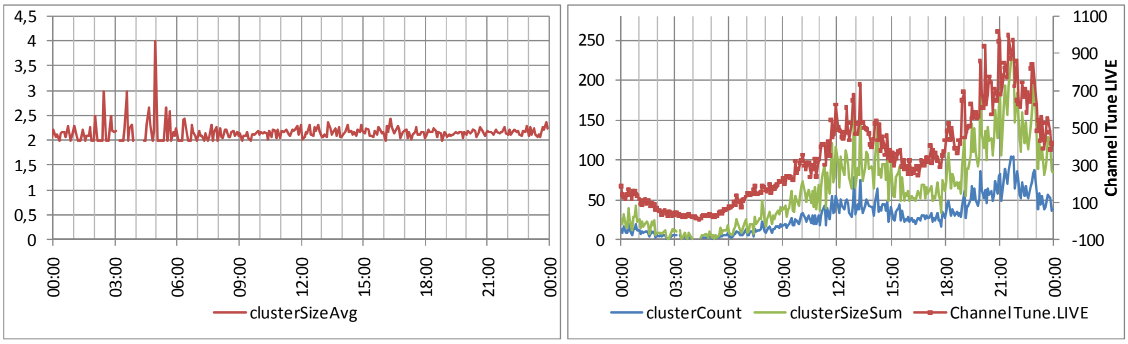

Figure 3-34 Channel Tune average cluster size (left), and comparison of the distribution of the Channel Tune event with the distribution of the clusters and the sum of events in the clusters (the relationship is aproximatly linear)

We could have increased the average size of the resulting clusters by increasing the distance that entitles two events to belong to the same cluster. However, we considered that if two Channel Tune events are separated by 2 minutes they already represent a somehow sparse workload on the platform. Nevertheless, given the pdf of the distance between events, it is reasonable to assume that in between the logged events of the same cluster, other channel tune actions occurred but were not logged.

A different approach could estimate the time between the end of a Channel Tune (unlike the previous analysis which uses the start time) and the start of the next Channel Tune, and assume that in that period other channel tune actions could occur. Nevertheless, there would be no data to support the decision whether those additional actions occurred and in what quantity. Furthermore, the start time of the first event, which is the moment when additional workload is requested from the platform, could be too far away in the past to be relevant.

The conclusion from this analysis is that it is not advantageous to make an event distance based analysis on the Channel Tune event because of the filtering that the STB makes on the short duration channel tunes. We realized that this analysis did not bring significant additional information any different from what was obtained from the plain distribution of the events throughout the day.

Daily distribution per channel

So far the analysis has considered the aggregated information regarding all TV channels. Nontheless, it is reasonable to expect that the behavior differs from one channel to another. Figure 3-35 represents the distribution of Channel Tune events and their duration regarding generalist channels, while Figure 3-36 represents the same type of information but for two specialized channels. In both charts, the period between 0:00 and 8:00 is not completely covered by the scale because those values have a very large variation and the focus is the period from 8:00 onward. The reason for such behavior during dawn is the small statistical information combined with STBs being left on during the night, generating very large average durations.

The specialized channels (e.g., movies and TV series) do not follow the same daily pattern characterized by two daily peaks. They present multiple peaks, increasing in size during the day, and reaching a maximum around 21:00. The average duration of the Channel Tune also reaches maximum by the end of the day in depending of the channel’s characteristics.

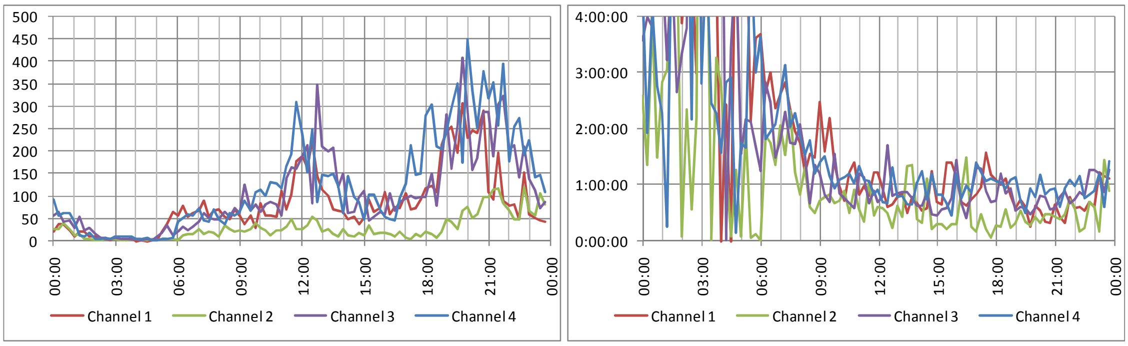

Figure 3-35 Amount of Channel Tune events (left) and average Channel Tune duration of 4 generalist channels

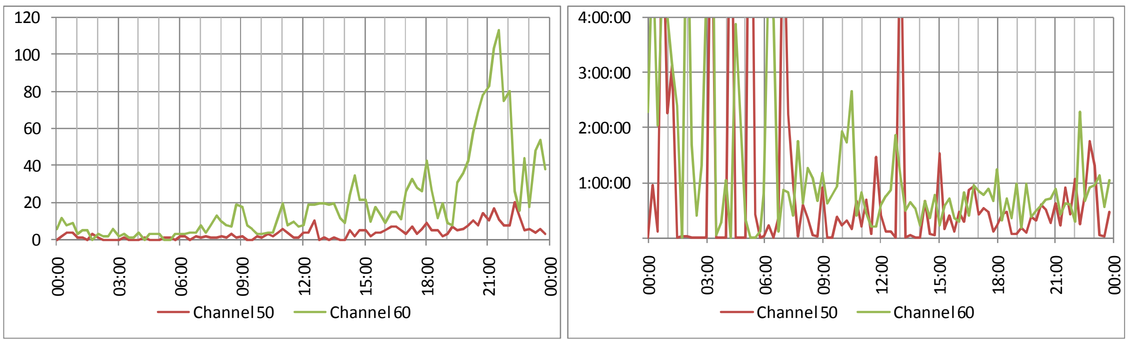

Figure 3-36 Amount of Channel Tune events (left) and average Channel Tune duration (right) of 2 specialized channels

The duration of the Channel Tune event allows computing the amount of STBs which were, in any given moment, tuned to each channel. While the distribution of the Channel Tune events identifies the moments when the STB tunes to a new channel, the duration of the Channel Tune allows identifying the period of time during which the STB remained member of the multicast group regarding that channel. Although during this period the STB is receiving the video stream by multicast, it is likely to request D-Servers to retransmit any lost packets.

Therefore, it is important to know how many STBs are members of the broadcasting groups as a means to estimate the likelihood of having additional requests to the retransmission system. Furthermore, multicast membership operations are more sensible with large multicast trees as any problem can potentially affect a larger number of users and take longer time to recover. Figure 3-37 is the result of pre-computing the end time of each live TV Channel Tune event and determining for each instant, in 5 minutes steps, the amount of Channel Tune events which have started before and ended after that instant. This is a different perspective from the one obtained from just looking at the moments when Channel Tune events start which keep increasing until around 13:15 (see comparison in Figure 3-38). Likewise, from 16:00 onwards, the global amount of simultaneous viewers increases until it peaks at roughly 20:00, while the amount of Channel Tune events increases until 21:30. Despite the increase in the count of events, we could realize that the platform was actually losing simultaneous users from 12:30 and from 20:00 onward. The connected users were changing channels more often (larger amount of channel tune events) and staying in the same channel during less time (decrease in the average channel tune duration).

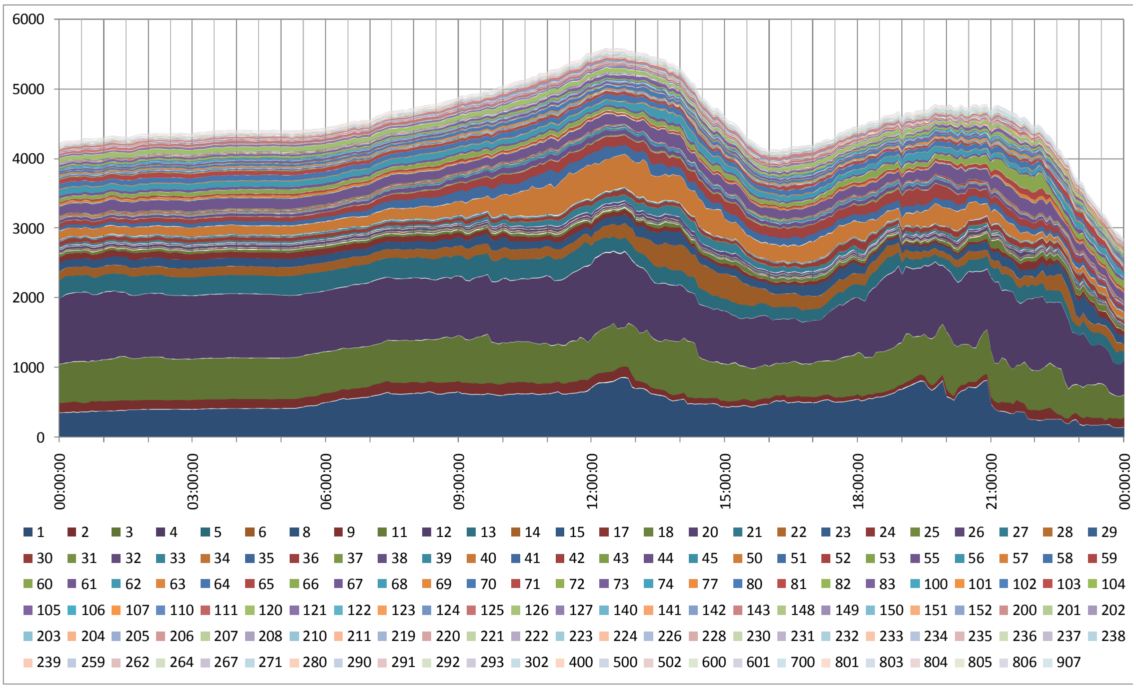

Figure 3-37 Amount of STBs with an active Channel Tune of a live TV Channel, grouped by the channel number, in 5 minutes steps

It is now clear that the distribution of the Channel Tune event is not a direct representative of the audience, and thus of the size of the multicast groups because the duration of the channel tunes varies and is in average relatively large. The actual audience distribution is phase shifted with the Channel Tune distribution (Figure 3-38) and the causes for the shift are different combinations of the amount of Channel Tune events and their duration.

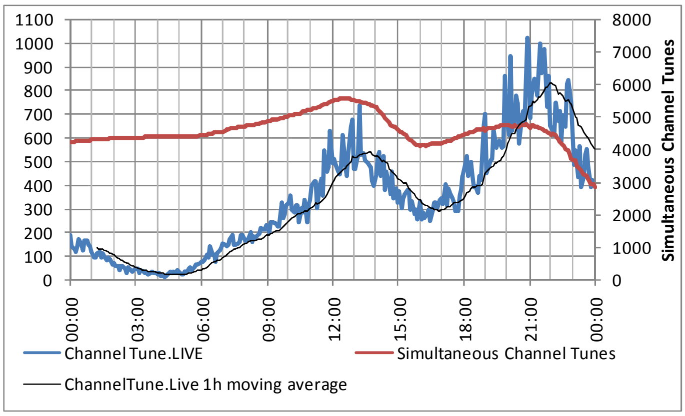

Figure 3-38 Comparison between Channel Tune event distribution and the amount of simultaneous active channel tunes

We must also notice the effects of the data limitations previously discussed regarding the Activity Logs (see section 3.3.2). We consider unlikely that during prime time (around 20:00) there are less ongoing Channel Tunes than during the afternoon and the difference between connected STBs at 0:00 and 24:00 is also very high (roughly 30% decrease). The most plausible reason for this discrepancy is in the large amount of STBs for which the latest available information is around 15:30 and during the evening, as discussed in section 3.3.2. If that was not the reason, than the amount of powered-on boxes would have to decrease during the afternoon, and the Box Power event could reveal that. As we can see in Figure 3-39, from 13:00 to 15:30 the variation of powered-on STBs is null, and then increases during the afternoon. Therefore, the total amount of channel tunes during prime time should be greater than in the beginning of the afternoon.

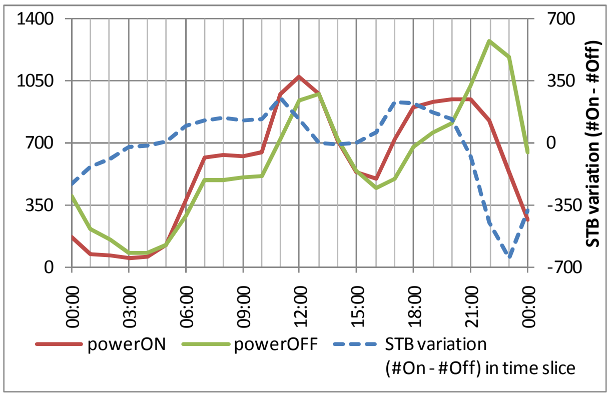

Figure 3-39 Distribution Power Box On/Off event, and their difference as the variation of powered-on STBs

3.3.6 Browse Panel

The Browse Panel event occurs when the user browses through the available channels using the GUI. When this is done, the STB receives a video unicast stream of the browsed channel from a Distribution Server. This video stream is presented in PIP and the Distribution Server will eventually assist the channel change (ICC) if the user selects that channel. These events are expected to occur in bursts, as the user “zaps” through the channels searching for something of his interest. Therefore we assume that for each user, these events can occur in groups. Unlike the Channel Tune event, we are unaware of any logging restrictions performed by the STB. By computing the time distance between each Browse Panel event that occurs within the same “power session” and grouping the events that are close within the given bound we can identify clusters of events which can thus recognize periods of increased workload.

To determine the adequate distance measure to define if two events are in the same cluster, we first computed the pdf of the distance between Browse Panel events of the same STB and the same “power session”. Similarly to the rational regarding the Chanel Tune event, the “power session” allows us to ignore the distances between events that have a power off in between them. The resulting pdf is plotted in Figure 3-40 and we see that 90% of events are separated by less than 1 minute. The remaining 10% are scattered through higher distance values. Therefore, the time limit for cluster separation is set to 1 minute.

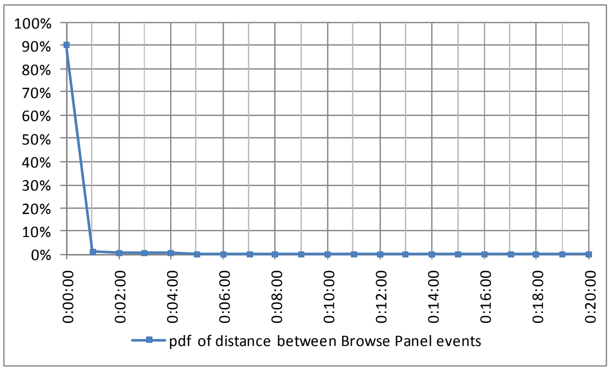

Figure 3-40 Probability distribution function of the distance between Browse Panel events of the same STB and same “power session”

The results of the clustering are shown in Figure 3-41.

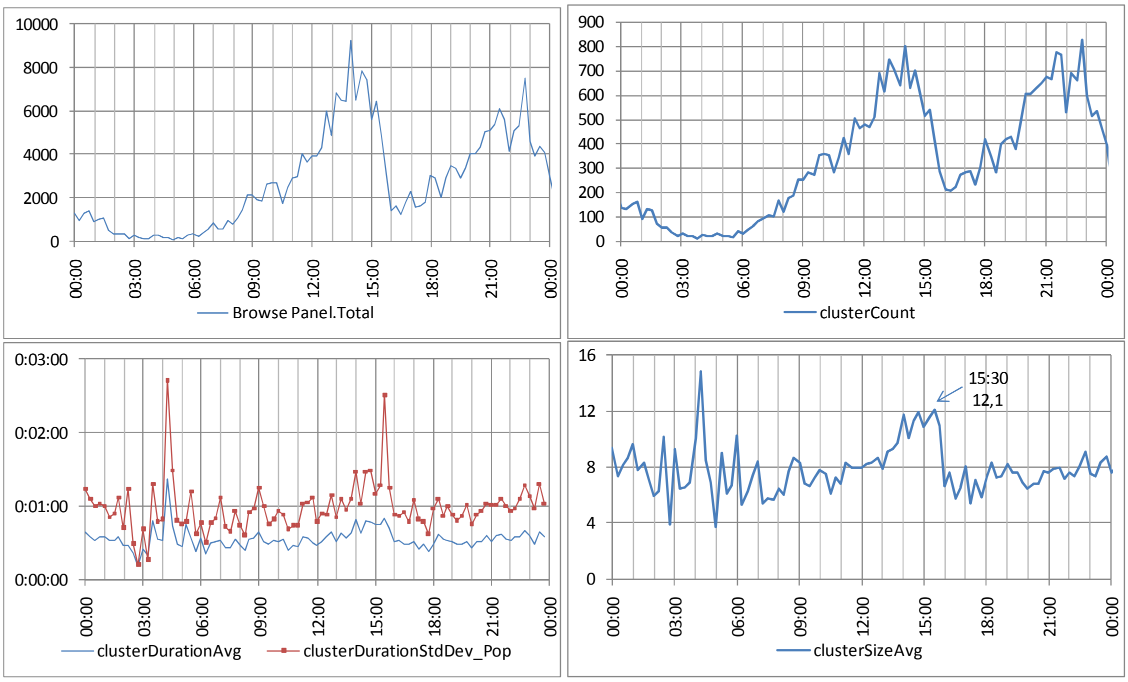

Figure 3-41 Browse Panel event distribution (top left) and Browse Panel clusters, in 15 minutes aggregates

The Browse Panel clusters follow the same global pattern with two daily peaks as the plain Browse Panel event distribution. These periods also correspond with the peak periods identified in the Channel Tune events and are occurring at the same time. The average cluster size appears to be globally quite stable throughout the day, being on average about 7 channel browses each time. Nevertheless, in the peak period between 13:00 and 15:30 (and only in this period), the amount of events in each cluster increases by approximately 50%. It then drops back to an average cluster size of 7, remaining like that throughout the prime time period.

We speculate that the most critical period in terms of workload contribution occurs when both the average size of the clusters and the amount of clusters is at peak. Graphically analyzing Figure 3-41, this period occurs from 14:00 to 14:30, right after the overall amount of Browse Panel events peaks. In order to test this hypothesis we need to look at the available performance information.

In what respects with the duration of the channel browse process, that is, the amount of time that a user spends scrolling from channel to channel until he settles for a particular channel, there is also a period from 14:00 and 15:30 where the average duration of the browse period is above the average. This result sustains that during the same period there is an increase on the amount of events in each cluster, which is therefore the cause of this duration increase.

3.3.7 Trick State

The Trick State event occurs whenever a user interacts with the STB and triggers a change to a new trick state, for example, Pause, Fast Forward, Rewind, or Play. For a Live channel or previously recorded DVR content, the trick streams (e.g. Rewind, Fast Forward) are locally produced by the STB based on the received video stream, and a request to a trick stream does not generate additional traffic on the service provider's network. On the other hand, trick actions regarding VOD content result in requests to the VOD servers. Figure 3-42 illustrates the distribution of Trick State events regarding Live, DVR and VOD channels. The overwhelming majority of Trick State actions are related to the DVR channels followed by Live channels. Trick State actions regarding VOD channels are less frequent because VOD purchases and viewing also occurs much less frequently than tuning a live channel or watching a previously recorded program.

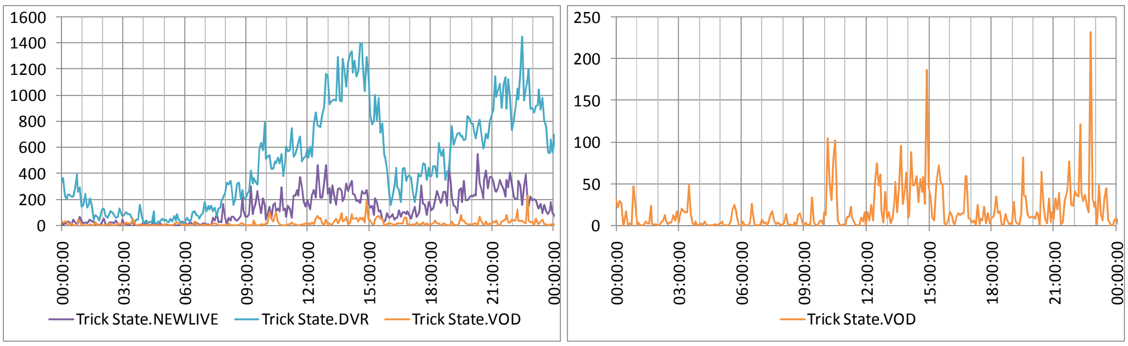

Figure 3-42 Distribution of Trick State events regarding Live, DVR and VOD channels (left), and detail of Trick State events regarding VOD channels, in 5 minutes aggregates

The Trick State actions are Play, Pause, Stop, Skip, Fast-Forward, Rewind, Replay and Stop. Those that should introduce more workload on the servers are those that require the server to start or resume a video stream, which correspond to the Play, Fast Forward, Rewind, Replay and Skip actions. We reanalyzed the VOD Trick State distribution grouping the events in two sets: (i) with the trick state actions considered to increase the video server's workload, and (ii) with the remaining actions. The result is presented in Figure 3-43 from where we conclude that the majority of actions are the ones that incur in a higher workload. Hence, we conclude that Trick State events should not be ignored when analyzing VOD usage because most of the user's actions can increase the server's workload.

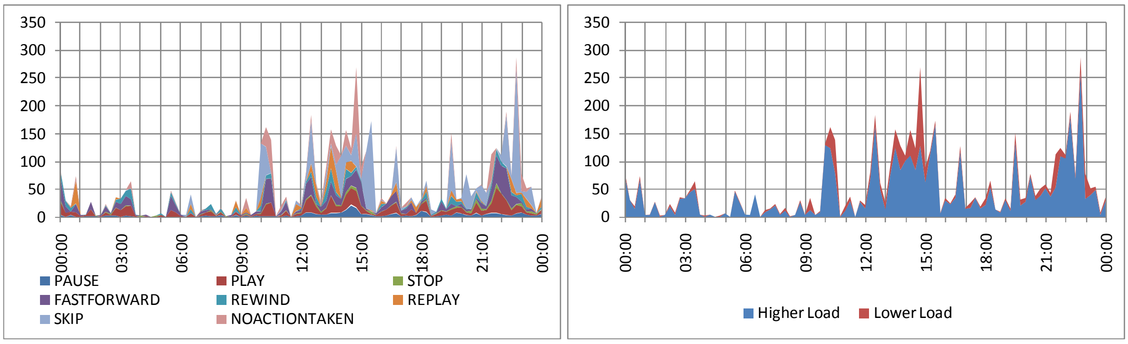

Figure 3-43 Cumulative representation of Trick State actions for VOD channels (left), and cumulative representation of Trick State actions for VOD channels grouped by those requiring higher and lower workload from video servers (right), in 15 minutes aggregates

3.3.8 Stream Management Stream Request

The Stream Management Stream Request (SMSR) is the event that occurs most frequently in the Activity Logs. Although it is not directly triggered by an explicit user action (e.g., press of a button on the remote control), it is related with the state of the STB, for instance, when tuned to a Live channel or VOD. The daily distribution follows a similar daily pattern as the Channel Tune or the Browse Panel. Figure 3-44 shows the daily distribution for this event and also the average amount of SMSR events per STB in 15 minutes aggregates. The SMSR events referring to live stream requests are completely dominant over those referring to DVR or VOD, being nearly three orders of magnitude larger. We note that around 15:00 and by the end of the day, the average SMSR events per STB reaches a significant peak.

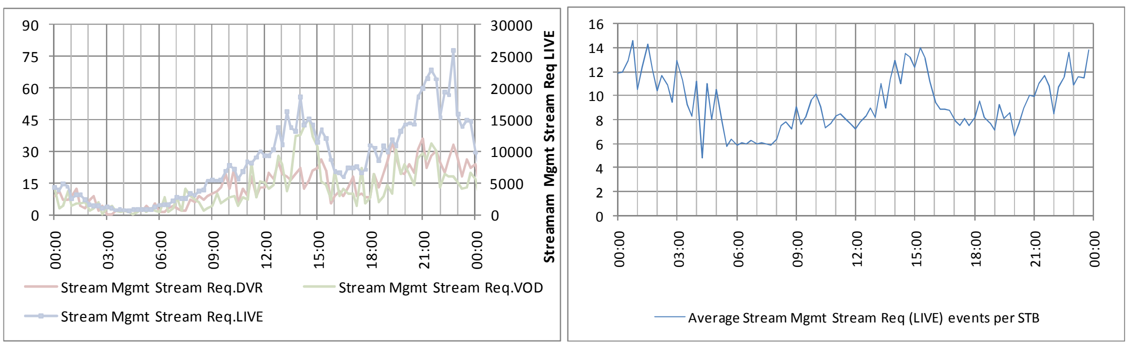

Figure 3-44 Daily distribution of the Stream Management Stream Request event, grouped by stream type, in 5 minutes aggregate – the secondary vertical axe refers to the Live stream type which occurs in much larger order of magnitude (left). And average amount of Stream Management Stream Request events per client, in 15 minutes aggregate (right)

The SMSR event is also characterized by an attribute called stream priority. The known values for the stream priority are 100000, 200000, 300000, and 400000. From the sample we derived the following rules:

- Live stream requests can have priority 100000, 200000 or 400000.
- DVR stream requests have priority 300000.
- VOD (HD or SD)stream requests have priority 200000.

The daily distribution of the SMSR events grouped by media type, priority and stream definition (Figure 3-45) allow understanding the meaning of the priorities the system. The priority 100000 traffic is the least frequent one and only occurs for some Live streams. This is the highest priority available. The following priority, 200000, is only one used for VOD content which therefore has higher priority than most of the traffic. The majority of the Live traffic has priority 300000 and the second most frequent has the lowest priority, 400000. DVR playback is categorized with the “standard” priority of 300000, but DVR traffic only refers to local playback by the STB or to playback requests from another STB in the home network.

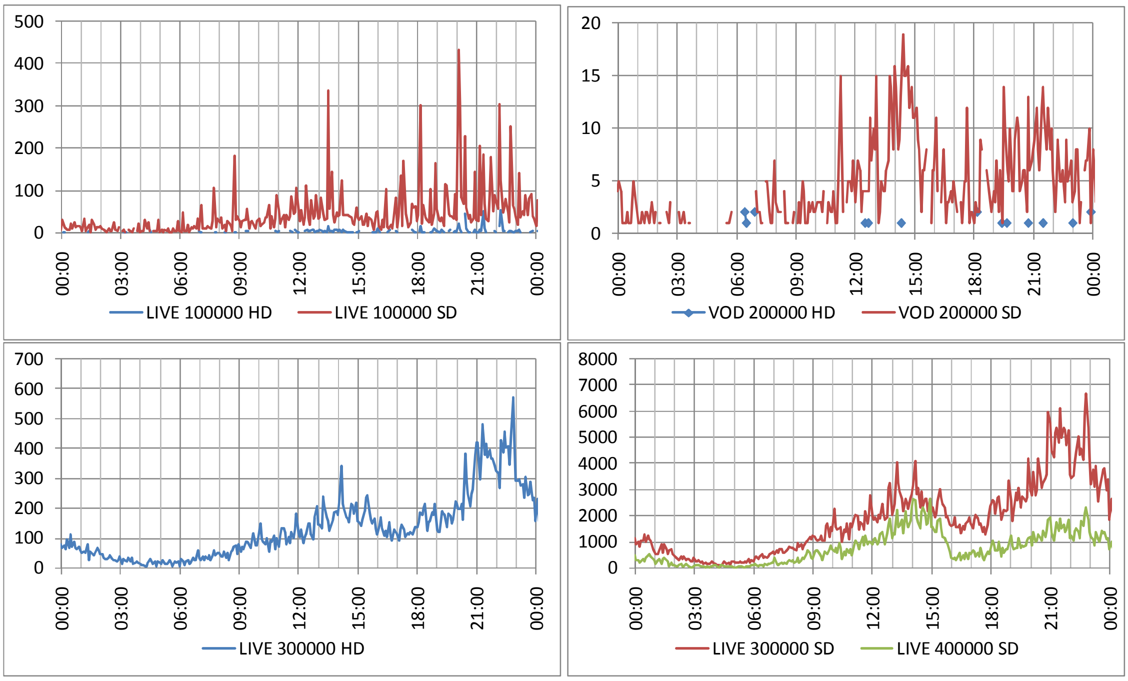

Figure 3-45 Distribution of SMSR events grouped by media type (Live, DVR, VOD), priority and stream definition (SD, HD)

Finally, we try to understand the relationship between the SMSR event and other events that are known to trigger stream requests, such as the Channel Tune and the Browse Panel. Figure 3-46 shows the daily evolution of the ratio between SMSR events and Channel Tune plus Browse Panel events. We observe that the ratio remains constant during several periods, and it is remarkably stable from 16:00 until 00:00 which is a long period of increasing demand and increasing amount of interactions (see Figure 3-44, Figure 3-41, and Figure 3-31).

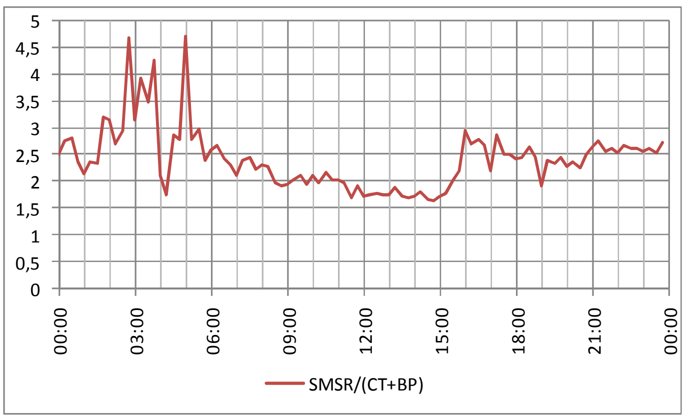

Figure 3-46 Relation between SMSR (regarding Live streams) event and the sum of the Channel Tune events (regarding Live channels) and Browse Panel events, in 15 minutes aggregates

In what concerns VOD, there is no event with equivalent functionality as the Browse Panel. The only events that should trigger a VOD stream request is tuning on a VOD channel. Therefore we presume that the relation between the occurrences of the SMST and Channel Tune events that are categorized as VOD, should also be linear. Figure 3-47 shows the occurrence of these events and plots the ratio between the two. We confirm that the relationship between these two events is mostly constant, with exception for some peaks that occur in the early morning.

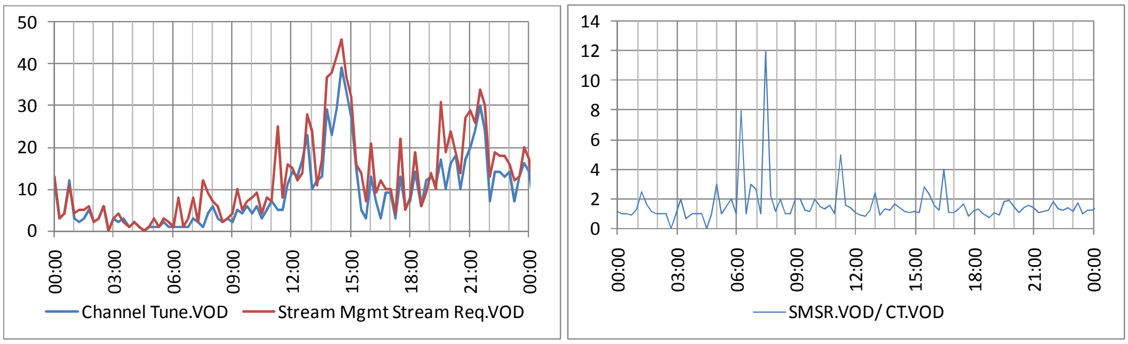

Figure 3-47 Daily distribution of Channel Tune and SMSR events regarding VOD channels (left) and the ratio between the these two events (right), in 15 minutes aggregates

3.4 Summary and discussion

We now present a summary of the main results obtained in this project and combine some of them reaching further conclusions.

User activity and D-Server's efficiency

When analyzing the efficiency of the D-Servers we have realized that there are periods where the occurrence of efficiency problems is more frequent. This can be clearly seen in Figure 3-48 where the period from 11:00 to 16:30 is denser.

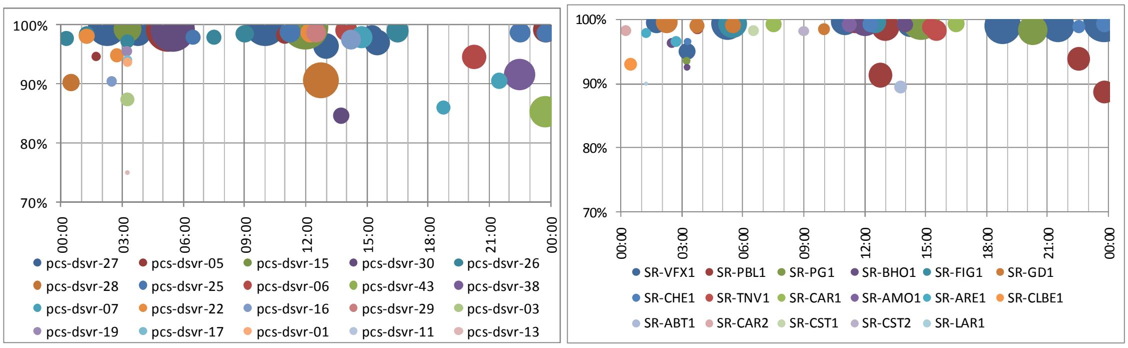

Figure 3-48 D-Server's efficiency per server (left), and per geographical area (right)

The question is: what is the cause of this increased density? The cause cannot solely be based on the amount of events/users because later in the evening there are much more events while the density of these occurrences is inferior.

We found that it is possible to identify statistical outliers of the packet requests made to the D-Servers. Furthermore, we identified that these outliers can occur almost simultaneously for a subset of servers, suggesting that some momentary common characteristic of the servers or of the requests is the cause (Figure 3-49). The period with higher density of efficiency problems maps with the occurrence of peaks in the amount of requested packets.

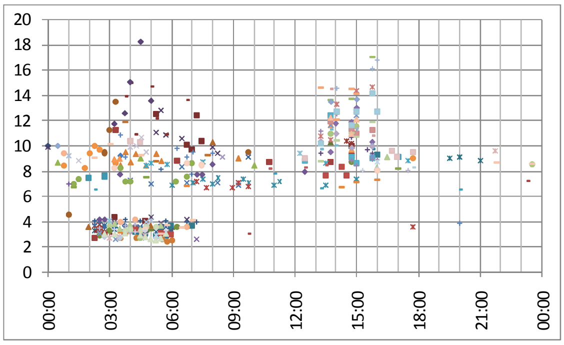

Figure 3-49 Average amount of packet requests in 15 minutes aggregates that are considered outliers in each server's distribution

In terms of user activity, we found that the average Browse Panel cluster size peaks at 14:00 and remains high until 15:30. This period with large Browse Panel clusters contributes to the workload of the D-Servers. Since the cluster size returns to a lower value and the D-Servers show more sparse efficiency problems for the reminder of the day, it is likely that these two features are correlated.

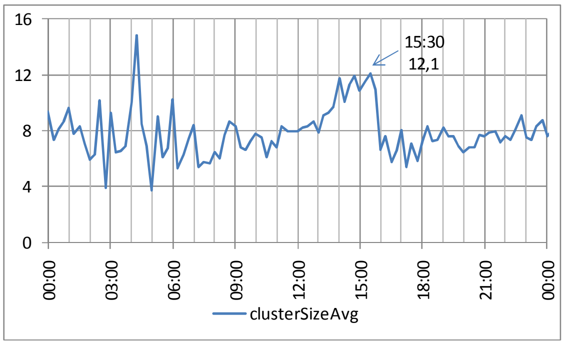

Figure 3-50 Browse Panel average cluster size

Additionally, we found that each STB triggers more SMSR events in the same period as the efficiency problems were more frequent. This includes the end of the day, when the amount of SMSR events as well as the global audience size was proven to be decreasing while the number of SMSR per STB increases (Figure 3-51). Therefore, the amount of SMSR events per STB influences the D-Server's efficiency.

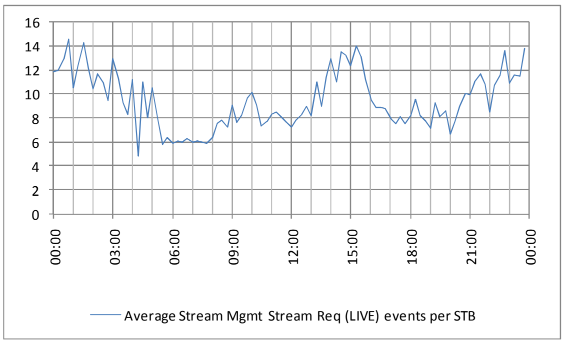

Figure 3-51 Average amount of Stream Management Stream Request events per client, in 15 minutes aggregate

D-Server load balancing mechanism

We realized that from 16:00 onward, the overall amount of requested packets remained fairly constant, even if the amount of distinct clients connected with the D-Servers increased. This result suggests that a load balancing mechanism is in place and the servers analyzed in this study are somehow "saturated" in the sense that requests are being "redirected" to other servers. If this is the case, then the period between 13:00 and 16:00 can in fact represent the moment when the amount of requested packets reaches a threshold that triggers the load balancing mechanism. This activation can then cause some instability reflected in the multiple peaks in requested packets, and ultimately several efficiency problems.

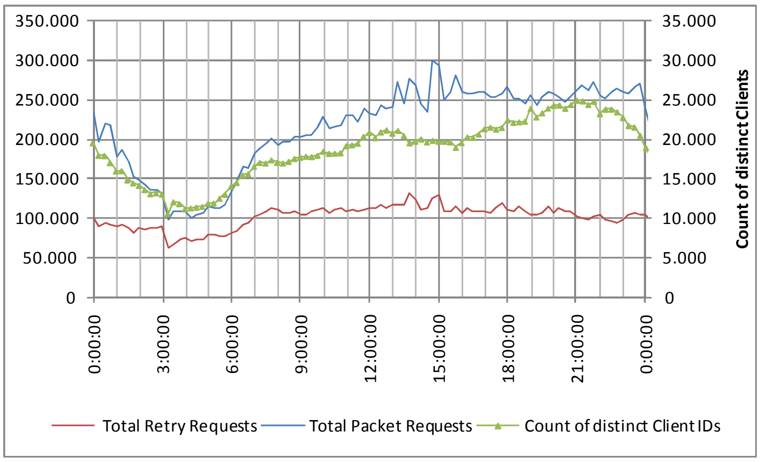

Figure 3-52 Overall amount of requests, packet requests, and count of distinct client IDs (secondary vertical axe) in 15 minutes aggregates

Browse Panel distance analysis

The majority of Browse Panel events of the same user and of the same "power session" are separated by less than 1 minute. Therefore, we grouped them in clusters of consecutive Browse Panel events, separated by less than 1 minute and added this information to the analysis of the system's workload.

We speculate that the most critical period in terms of workload contribution occurs when both the average size of the clusters and the amount of clusters is at peak. This period is from 14:00 to 14:30, although we cannot verify it with the available information. However, in terms of efficiency, there is no evidence that the period between 14:00 and 14:30 is worse than the remaining period where efficiency problems were detected.

Channel Tune distance analysis

From the analysis of the distance between the Channel Tune events, we conclude that it is not too relevant because the STB filters channel tunes with short duration. We verified that this analysis did not bring additional information that could not be obtained from the plain distribution of the events throughout the day.

Log data reduction

The Program Watch event is highly correlated with the Channel Tune because it provides a time mark whenever a program changes during the period that the user is tuned to a given channel. Therefore, for each Channel Tune event there is at least one Program Watched, and depending on the amount of time that the user stays tuned with the channel, there can be multiple Programs. If there is no interest in querying the database to find the audience of a specific program, the Program watched event does not bring additional interesting information that cannot be obtained from the Channel Tune event. Since it represents nearly 12% of all logged events, this is a relevant conclusion if storage is to be optimized by reducing the amount of data.

Further data reduction can be achieved disregarding the Trick State events that are not related to VOD channels, which have very few occurrences when compared with the Trick State events related to DVR and Live. This would additionally reduce 10% the amount of stored information.

Chapter 4
Conclusion

In this report we present a summary of the information that can be obtained from the D-Server's statistical reports, from the VOD servers IIS logs and the STBs user activity logs of a large production IPTV platform. In what concerns the servers, the main focus was to characterize their workload and identify metrics that could indicate performance problems. In what concerns the STBs, the main focus was to identify the subset of events that cause higher workload on the platform. The final objective was to find possible correlations between them, increasing our knowledge of the factors that influence the system's performance, and acquiring new tools for interpreting it.

We have defined the D-Server's efficiency as the ratio between served packets and requested packets. In the surveyed period there were efficiency problems in 20 of the 79 analyzed servers (25%) and in 20 of the 294 channels.

We found that the period where the D-Server's efficiency is more often inferior to 100% is from 13:00 to 16:00, which does not coincide with the prime time period, as would be expected. In fact, we found that in the period from 13:00 to 16:00 there are multiple peaks in the average amount of requested packets and these peaks are outliers of the daily distribution of the average amount of solicited packets. We verified that these outlier data points can be successfully identified using the criteria of their distance to the daily average being greater than twice the standard deviation of the sample. This rule can be used to identify servers with abnormal workload and raise alerts with high probability of efficiency problems occurring in the system. These problems may be related, for example, with the management of multicast groups, or with the access network.

Additionally, we also found that the average amount of packets per request, which is usually bounded between 1 and 4, presents an outstanding peak at 3:15 which completely disrupts the daily distribution of this metric. Simultaneously, we observed that there are multiple servers reporting efficiency levels below 100% with some geographic consistency in 3 identified areas (VFX1, BHO1, and PG1). The subset of servers is not the same in both observations but the time consistency of both occurrences suggests that they are related. This efficiency problem may have caused an increase of the amount of packets solicited in each request. In practice, these finding suggest that the amount of solicited packets per request is a metric that should also be monitored as a means of identifying efficiency problems occurring in the system.

Despite the increasing amount of distinct STBs that connect to the D-Servers during the afternoon, the total amount of requested packets has a "saturation" value that remains roughly constant. Nevertheless, the amount of requested packets flickers before "saturating", suggesting that a transition period exists. This period corresponds with the moments when the efficiency was more frequently different from 100%.

In our analysis we also found that there are gaps in the 5 minute pace of the D-Server production of statistical information. Although we could not correlate this information with the previously analyzed information, this behavior should be further analyzed.

Regarding the VOD servers, we found that there are 2 distinct groups of servers in terms of their load profile. With this information it is possible to determine if the VOD asset allocation is correctly distributed through the VOD Server architecture. Furthermore, the workload distribution should be reviewed as the less popular servers form a much larger group than the most popular, and have significantly less workload.

We also characterized the user activities that most influence the platform's workload, namely (i) the Channel Tune event that involves ICC and increases the multicast tree size, (ii) the Browse Panel that requests a PIP video stream to the D-Servers, and (iii) the Trick State that interacts with the video stream. We also looked in detail to the Stream Management Stream Request event, which is a management event but is related with the state of the STB, for example, if it is tuned to a Live channel or VOD.

We found that 50% of the Channel Tune events are shorter than 8 minutes and 90% have duration shorter than 1,5 hours. We analyzed the distance of the Chanel tune events and concluded that such analysis is not advantageous because the STB does not log the Channel Tunes with less than 1 minute. On the other hand, we found that the Browse Panel event usually occurs in clusters separated by less than 1 minute containing an average of 7 events. Nevertheless, between 13:00 and 15:30, the amount of events in each cluster increases by approximately 50%. This characteristic is expected to increase the workload on the D-Servers and, in fact, this period matches with the period where the efficiency problems on the D-Servers were more frequent. This increase of Browse Panel events also increases the amount of SMSR events. The amount of SMSR events per STB also peaks during this period where efficiency problems were noticed. Furthermore, by the end of the day, when the global audience size decreases, the increasing SMSR events per STB matches with the additional occurrences of efficiency problems on the D-Servers.

Finally, we note that the amount of retained information regarding the Activity Logs can be reduced by 20% disregarding the Program Watched event as well as the Trick State events related with Live and DVR channels. This reduction does not influence the ability to analyze the STB information that most influences the system's performance.

In order to complement these results we suggest:

- To use a data sample covering multiple days ensuring that the STB upload mechanism does not influence the quality of the analyzed data. Include all D-Servers in the analysis because the STBs can connect to any of them. If possible, include all or multiple STB Service Groups.
- To develop and integrate mechanisms to automatically compute and make available the most relevant performance and activity metrics. For example: average Browse Panel cluster size, Live Stream Management Stream Requests per STB, automatic detection of abnormal values of the amount of requested packets per retransmission request.
- To perform laboratory load testing, injecting requests simulating user actions and measuring the system's response under different workload conditions.

- To collect costumer reports of technical failures and correlate them with the discussed metrics to measure their relation with the probability of a user noticing the failure and file a complaint.

Throughout this study we also identified some practical issues which were left open but should be addressed in the future:

- To confirm the reason why the records on the D-Server reports indicate more served than requested packets.
- To analyze the load balancing mechanisms between D-Servers, verify its limitations and evaluate its performance.

Appendix

A.1 Activity Log event codes

EventID	EventDescription
100	Channel Tune
101	Box Power
102	VDP Purchase
103	RDP Purchase
104	Trick State
105	Browse Panel
106	Application
107	Menu Selection
108	RDP Application Launch
109	RDP Application Disconnect
110	RDP Application Navigate Away
111	RDP Application MCE Tune
113	PPV Purchase
114	Program Watched
115	DVR Start Recording
116	DVR Abort Recording
117	DVR Playback Recording
118	DVR Schedule Recording
119	DVR Delete Recording
120	DVR Cancel Recording
121	Credit Limit Exceeded
122	EAS Received
123	EAS Level 1 Displayed
124	EAS Level 1 Terminated
125	EAS Level 2 Displayed
126	EAS Level 2 User Canceled
127	EAS Level 2 Expired
128	EAS Event Preempted
180	Blackout Start
181	Blackout End
182	Blackout Change
17781	Stream Management Detune
17782	Stream Management Profile Update
17783	Stream Management Reboot
17784	Stream Management Contention
17785	Stream Management Stream Request
17786	Stream Management TV Interrupt
20203	Message Failure
20204	Message Success
36998	Blackout End
36999	Blackout Service Change

A.2 W3C extended log file fields

Field	Appears As	Description
Date	date	The date on which the activity occurred.
Time	time	The time, in coordinated universal time (UTC), at which the activity occurred.
Client IP Address	c-ip	The IP address of the client that made the request.
User Name	cs-username	The name of the authenticated user who accessed your server. Anonymous users are indicated by a hyphen.
Service Name and Instance Number	s-sitename	The Internet service name and instance number that was running on the client.
Server Name	s-computername	The name of the server on which the log file entry was generated.
Server IP Address	s-ip	The IP address of the server on which the log file entry was generated.
Server Port	s-port	The server port number that is configured for the service.
Method	cs-method	The requested action, for example, a GET method.
URI Stem	cs-uri-stem	The target of the action, for example, Default.htm.
URI Query	cs-uri-query	The query, if any, that the client was trying to perform. A Universal Resource Identifier (URI) query is necessary only for dynamic pages.
HTTP Status	sc-status	The HTTP status code.
Win32 Status	sc-win32-status	The Windows status code.
Bytes Sent	sc-bytes	The number of bytes that the server sent.
Bytes Received	cs-bytes	The number of bytes that the server received.
Time Taken	time-taken	The length of time that the action took, in milliseconds.
Protocol Version	cs-version	The protocol version —HTTP or FTP —that the client used.
Host	cs-host	The host header name, if any.
User Agent	cs(User-Agent)	The browser type that the client used.
Cookie	cs(Cookie)	The content of the cookie sent or received, if any.
Referrer	cs(Referrer)	The site that the user last visited. This site provided a link to the current site.
Protocol Substatus	sc-substatus	The substatus error code.

References

[1] Portugal Telecom. (2010, August) Results Presentation. [Online]. http://telecom.pt/InternetResource/PTSite/UK/Canais/Investidores/infofinanceira/reltrim/restrim.htm

[2] Stefaan Vanhastel and Raul Hernandez, "Enabling IPTV: What's Needed in the Access Network," *IEEE Communication Magazine*, vol. 46, no. 8, pp. 90-95, 2008.

[3] Ramesh Jain, "I want my IPTV," *IEEE Multimedia*, vol. 12, no. 3, pp. 95-96, July 2005.

[4] Heavy Reading. (2006, July) Assuring Quality of Experience for IPTV. http://www.heavyreading.com/.

[5] Olli Alanen, Mikko Pääkkönen, Timo Hämälinen, and Jyrki Joutsensalo, "Multicast Admission Control in DiffServ Networks," *Proc. ICON 2004*, vol. 2, pp. 804-808, November 2004.

[6] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," *RFC 3350*, July 2003.

[7] H. Uzunalioglu, "Channel Change Delay in IPTV Systems," *CCNC*, pp. 1-6, January 2009.

[8] M. Kalman, S.T. Eckehard, and B. Girod, "Adaptive playout for real-time media streaming," *IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 45-48, August 2002.

[9] Y. Km et al., "Reducing IPTV channel zapping time based on viewer's surfing behavior and preference," *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, pp. 1-6, April 2008.

[10] D. Banodkar, K.K. Ramakrishnan, S. Kalyanaraman, A. Gerber, and O. Spatscheck, "Multicast instant channel change in IPTV systems," *3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE)*, pp. 370 - 379, June 2008.

[11] A. C. Begen, N. Glazebrook, and W. V. Steeg, "Reducing Channel-Change Times with the Real-Time Transport Protocol," *IEEE Internet Computing*, vol. 13, no. 3, pp. 40-47, June 2009.

[12] P. T. Barrett, J. A. Baldwin, and D. L. Green., "Patent 20050081244: Fast channel change," April 2005.

[13] Suman Mandal and Michael MBuru, "Intelligent Pre-fetching to Reduce Channel Switching Delay in IPTV Systems," Texas A & M University, Department technical report.