IPTV streaming source classification

M. Masciopinto #1, P. Comesaña #2

Signal Theory and Communications Department, University of Vigo
E. E. Telecomunicación, Campus-Lagoas Marcosende, Vigo 36310, Spain

{1mmasciopinto, 2pcomesan}@gts.uvigo.es

***Abstract*—In the last years video streaming over IP networks has changed the entertainment habits of society. Video on demand and multicast services have proliferated, bringing new challenges. Although the majority of the new service providers are compliant with copyright protection policies, some services are streamed without the proper rights, leading to a new kind of content piracy. Therefore, content right owners are interested in finding out the distribution channel followed by these unauthorized contents, in order to learn about illegal distributor sources. We present the first approach to this problem, in which we deal with the classification of IPTV streamed contents on satellital (DVB-S) or terrestrial (DVB-T) sources, both for live and delayed streaming. Our proposal is based on analyzing the time distribution of IP packet dispatches, extracting high-order statistics, and performing the source classification using a SVM. The reported results show the goodness of the proposed approach.**

I. Introduction

The spectacular growth of Internet Protocol (IP) networks in the last decade, as well as the generalization of their use, have changed our way of interacting with people and learning, our consumption habits, and even our entertainment preferences. Concerning the latter, probably the first example are the widespread online games, although in the last years the online consumption of multimedia contents has also experienced a quantum leap. Due to its bandwidth characteristics and its added value, special attention is deserved by the IP distribution of video contents. Typically, two ways of video distribution can be found:

- Video on Demand (VoD): the user asks for the content to be delivered.
- Multicast: the content is scheduled for delivering and the user applies for being included in the destination list.

Generally VoD streams contents (videos or TV shows) which are stored at the streaming server (a.k.a. delayed or time-shifted playing), as movies or TV shows, while multicast deals with the streaming of live events, typically sports.

These new video distribution models have advantages from the point of view of the user:

- ease of use,
- possibility of choosing the screened contents among a huge selection,
- pause, play, rewind and forward features,
- paying just for those contents which are really consumed (unlike most of the previously used cable/satellite systems),
- enjoy their favorite programs/movies even when the users are not at home,

and open a number of possibilities to the content distributor:

- almost unlimited potential market,
- cheap infrastructures,
- well-known distribution technology,

leading to a new video distribution business model.

Nevertheless, new technological and legal problems must be also faced, probably being the most important of them the streaming of contents by providers not owning the corresponding rights. This misuse of the contents causes huge losses to the legitimate content right owners, who indeed are interested in tracing back the distribution channel undergone by the content for arriving to the server of the unauthorized provider.

Since terrestrial and satellital channels are two of the most important ways of broadcasting video programs, this work proposes the first (to the best of the authors' knowledge) scheme designed for classifying the origin of one streamed program between terrestrial and satellital, both for time-shifted and live streaming. Due to the geographical location of our group, the terrestrial streams are samples of Spanish Digital Vision Broadcast-Terrestrial (DVB-T), while the satellital ones follow DVB-Satellite (DVB-S), which are the standards in use throughout Europe. Furthermore, we will assume the streamed video to have high quality (specifically, no video or audio transcoding will be performed) and the resulting IP stream to be Internet Protocol Television (IPTV) [1] standard compliant. Finally, the VideoLan Client (VLC) libraries [2] were used for streaming in the reported results.

The remaining of the paper is organized as follows: Sect. II summarizes the main characteristics of IPTV, while Sect. III introduces the time statistics and Support Vector Machines (SVM) the classification is based on. Classification results are reported for time-shifted streaming in Sect. IV, and for live streaming in Sect. V; conclusions are summarized in Sect. VI.

II. IPTV description

IPTV [1] is a packet-switched network oriented protocol, whose target is the transport of Moving Picture Expert Group (MPEG-2) programs over IP-based networks. In order to do

WIFS'2012, December, 2-5, 2012, Tenerife, Spain. 978-1-4244-9080-6/10/$26.00

that, IPTV relies on a series of packet-switched network protocols; specifically, IP, User Data Protocol (UDP), and Real Time Protocol (RTP) [3], [4] are used in the transmission of the MPEG-2 Transport Stream (TS) [5] payload packets. Note that one MPEG2-TS might contain several programs; in this work the case of multicasting those programs one at a time with IPTV will be considered.

The typical structure of IPTV packets can be seen in Fig. 1, where the header structure, containing a header for each of the three mentioned protocols, is reflected; given that our proposal exploits the distribution of the IPTV packet dispatch time, special attention will be paid to the RTP part, as it contains a time stamp necessary for playing multimedia contents, as those of interest in this work.

Although in general any integer number of 188-byte MPEG2-TS packets can be found within an IPTV packet, the maximum size of Ethernet-based packets is upperbounded by 1492 bytes, so the maximum number of MPEG2-TS packets per IPTV packet is 7; since the use of IPTV packets with a smaller number of MPEG2-TS packets would be an underuse of the network resources, almost all the IPTV transmissions use IP packets with 7 MPEG2-TS packets. This structure is illustrated in Fig. 1.

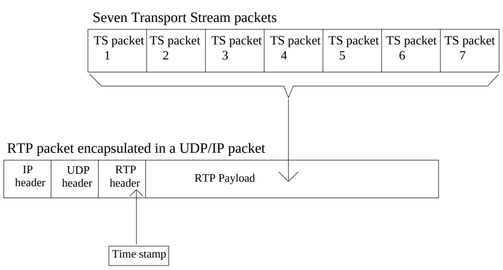

Fig. 1. Typical structure of IPTV packets.

A. Bitrate

Both DVB-T and DVB-S transmission is performed at a constant rate by using statistical time division multiplexing, a medium-sharing strategy where the different programs are aggregated; due to the variability of each of the constituent programs, which are frequently assumed to be mutually independent, the average total bitrate that can be transmitted for a given overflow probability is significantly larger than the average total bitrate that could be achieved if each program were assigned a bitrate in a static way. In order words, statistical multiplexing takes advantage of the non-coincidence of instantaneous bitrate peaks, leading to an efficient channel use. Since the bitrate for standard DVB-T and DVB-S channels is significantly different, about 20 and 42 Mbps, respectively, the number of multiplexed programs will be also different for both schemes, and consequently the effect of the statistical multiplexing is expected to be also characteristic for the two compared video sources. Indeed, this is the basic idea we try to exploit in this work: identifying the source (DVB-T or DVB-S) of a video stream by analyzing the time footprints left by the different statistical multiplexing.

Nevertheless, besides the time footprints left by the statistical multiplexing of DVB, one has to be also aware of the time modifications introduced by the IPTV server when preparing the IPTV stream. Specifically, IPTV bitrate has characteristic features [1], as the fact that streams containing multiple Program Clock References (PCRs) time bases are constant bitrate streams, while streams containing a single PCR time basis may be constant or variable bitrate. The latter is probably the most extended case, as it allows to adapt the instantaneous bitrate to the characteristics of the video to be transmitted; but even for variable bitrate transmission, the bitrate is piecewise constant, and consequently the IPTV server shall produce a piecewise uniformization of the output program rate, where constant rate bursts will be observed. As long as the the bitrate changes provide information about the input program rate variability, they will also contain clues about the kind of statistical multiplexing that was used for aggregating that video source. An example of instantaneous bitrate of a streamed program delivered over IP using IPTV is shown in Fig. 2. In that figure one can also observe the presence of low bitrate peaks in some constant rate bursts; they are probably due to some VLC artifact related to the case where the buffer at the input of the rate uniformizer is getting empty. A detail of one of those low bitrate peaks is shown in Fig. 3.

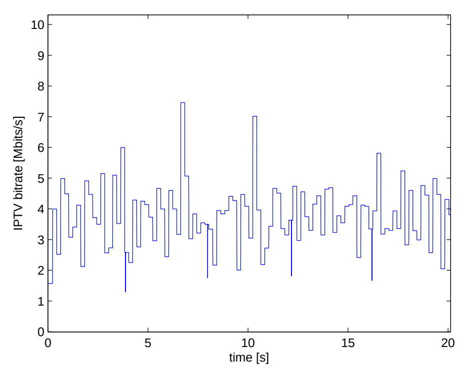

Fig. 2. IPTV piecewise constant bitrate example of a streamed DVB-T program.

III. Empirical statistics definition and SVM

In order to get information about the IPTV instantaneous rate, that, as it was mentioned before, is the footprint we will focus on for classifying the source of the studied stream, we will consider the RTP timestamps present in each IP packet [3] (Fig. 1), and compute the difference between consecutive timestamps. Denoting by $q_k(i)$ the ith RTP timestamp of the kth considered program, where $k = 1, \ldots, N$, N is the number of streamed programs, $i = 0, \ldots, n_k - 1$, and n_k is the number of IP packets corresponding to the kth streamed

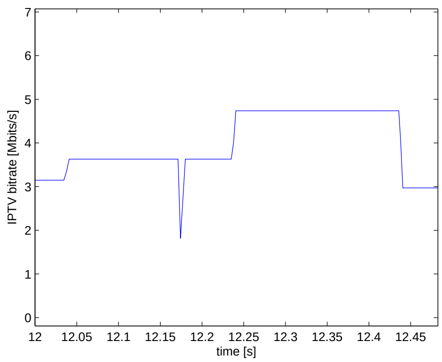

Fig. 3. Detail of IPTV piecewise constant bitrate example of a streamed DVB-T program with a small bitrate peak.

program, our analysis will be based on

$$R_k(i) = q_k(i+1) - q_k(i).$$

Let $\mathbf{R}_k$ be the vector collecting all the samples of $R_k(i)$. Then, the following characteristics are defined trying to summarize the rate variability of the DVB program that was input to the IPTV server:

- $T_k(j)$: length (in number of IP packets) of the jth constant bitrate burst for the kth considered program,
- $P_k(j)$: IP packet period of the jth constant bitrate burst for the kth considered program,
- $D_k(j) = |P_k(j+1) - P_k(j)|$: absolute value of the difference between periods of consecutive bursts for the kth considered program,

where $j = 0, \ldots, m_k - 1$, and m_k is the number of rate changes for the kth program. Since low bitrate peaks do not appear in the middle of constant rate bursts, their effect will not be considered for the calculation of the characteristics we have just defined. The vectors collecting all the samples of $T_k(j)$, $P_k(j)$, and $D_k(j)$ are denoted, respectively, by $\mathbf{T}_k$, $\mathbf{P}_k$, and $\mathbf{D}_k$.

Notice in any case that these empirical statistics are probably not very useful from a practical point of view, as the full length of the considered program capture must be processed for obtaining the values introduced above. Alternatively, we will split $\mathbf{T}_k$, $\mathbf{P}_k$, and $\mathbf{D}_k$ in L-length blocks, i.e.,

$$\begin{aligned} T_k^l(j) &= T_k(l \cdot L + j), \\ P_k^l(j) &= P_k(l \cdot L + j), \\ D_k^l(j) &= D_k(l \cdot L + j), \end{aligned}$$

where $l = 0, \ldots, r_k - 1$, $j = 0, \ldots, L-1$, and $r_k = \lfloor \frac{m_k}{L} \rfloor$; similarly, $\mathbf{R}_k^l$ contains the values of $\mathbf{R}_k$ for the IP packets in those constant rate bursts. Note that r_k depends on several factors, such as the observation period (i.e., how much time the streamed program is observed), or the frequency of IPTV bitrate changes. Thus, the value of r_k is difficult to predict.

This splitting is useful from a practical point of view, as the full length of the program is not required in order to classify it. Instead, a fixed number of rate changes are considered, significantly reducing the observation time necessary for performing the classification. This being said, strictly speaking the observation time required for producing these L-length blocks is not upperbounded; think, for example, of the degenerated case where the rate of the observed streamed program is constant. In any case, in typical situations this block division strategy will enable the classification in real scenarios. Sects. IV and V provide results on the average and maximum observation time required for obtaining one L-length block for the IPTV streams considered in each scenario.

Finally, it is worth mentioning that the rate variability depends on the type of considered content; e.g., the news' bitrate will be smoother than an action movie's one. Therefore, the fact of simultaneously considering different content types will make harder the correct classification, since no just the DVB-S and DVB-T statistical multiplexing will leave their footprint on bitrate variability, but also the content type itself. In the results provided in this work all the programs of the considered TSs were streamed (one at a time), so bias depending on the content type is not expected.

A. Empirical statistics

Once the vectors $\mathbf{R}_k$, $\mathbf{T}_k$, $\mathbf{P}_k$ and $\mathbf{D}_k$ have been computed (or equivalently $\mathbf{R}_k^l$, $\mathbf{T}_k^l$, $\mathbf{P}_k^l$ and $\mathbf{D}_k^l$), one must decide how the information in those vectors will be exploited to construct a classifier. Tab. I shows the statistics used in this work for classifying the streamed programs when the full-length version of vectors are used; similar definitions can be applied if their block-split versions are considered. The first column in Tab. I contains the identification number (ID) of each statistic to be used later, while the second column shows its definition.

ID	Definition
1	$\bar{T}_k = \frac{1}{m_k}\sum_{j=0}^{m_k-1} T_k(j)$
2	$s(\mathbf{T}_k) = \left(\frac{1}{m_k}\sum_{j=0}^{m_k-1}\left[T_k(j) - \bar{T}_k\right]^2\right)^{1/2}$
3	$\bar{P}_k = \frac{1}{m_k}\sum_{j=0}^{m_k-1} P_k(j)$
4	$s(\mathbf{P}_k) = \left(\frac{1}{m_k}\sum_{j=0}^{m_k-1}\left[P_k(j) - \bar{P}_k\right]^2\right)^{1/2}$
5	$\bar{D}_k = \frac{1}{m_k}\sum_{j=0}^{m_k-1} D_k(j)$
6	$s(\mathbf{D}_k) = \left(\frac{1}{m_k}\sum_{j=0}^{m_k-1}\left[D_k(j) - \bar{D}_k\right]^2\right)^{1/2}$
7	$s(\mathbf{R}_k) = \left(\frac{1}{n_k}\sum_{i=0}^{n_k-1}\left[R_k(i) - \bar{R}_k\right]^2\right)^{1/2}$
8	$sk = \frac{\frac{1}{n_k}\sum_{i=0}^{n_k-1}\left(R_k(i)-\bar{R}_k\right)^3}{\left(\sqrt{\frac{1}{n_k}\sum_{i=0}^{n_k-1}\left(R_k(i)-\bar{R}_k\right)^2}\right)^3}$
9	$kurt = \frac{\frac{1}{n_k}\sum_{i=0}^{n_k-1}\left(R_k(i)-\bar{R}_k\right)^4}{\left(\frac{1}{n_k}\sum_{i=0}^{n_k-1}\left(R_k(i)-\bar{R}_k\right)^2\right)^2}$
10	$m_5 = \frac{\frac{1}{n_k}\sum_{i=0}^{n_k-1}\left[R_k(i)-\bar{R}_k\right]^5}{s(\mathbf{R}_k)^5}$
11	$m_6 = \frac{\frac{1}{n_k}\sum_{i=0}^{n_k-1}\left[R_k(i)-\bar{R}_k\right]^6}{s(\mathbf{R}_k)^6}$

TABLE I
EMPIRICAL STATISTICS USED FOR CLASSIFICATION IN THE CURRENT WORK.

The used statistics are the first and second order empirical central moments of $\mathbf{T}_k$, $\mathbf{P}_k$, and $\mathbf{D}_k$, as well as the higher order (from 2 to 6) empirical central moments of $\mathbf{R}_k$ (including Fisher's measure of skewness, kurtosis, and the 5th and 6th normalized empirical central moments).

The mean of $\mathbf{R}_k$, i.e. $\bar{R}_k$ was not directly considered in Tab. I since it does not quantify the variability of the rate of the streamed programs, and indeed it is mainly related to the streamed video quality [6].

B. Threshold-based classifier

As a first step for defining a classification strategy, we use the statistics defined in Tab. I (specifically, their full-length version), and the source decision is based on the comparison of one of those statistics with a threshold. Although the resulting classifier is very simple, the achieved performance is quite poor, as it can be checked in Figs. 4 and 5, where the performance of the threshold-based classifier for statistics 1 and 7 is shown; the corresponding probabilities of classification error in the cross point are 36% and 39%, respectively. The results obtained by using other single statistics are similar.

Since the classification properties of these simple classifiers are not good enough, in the next section a more complicated classifier, based on the use of SVM [7] for fusing the information provided by the statistics mentioned above, will be introduced; by using this new classifier, one would expect the classification results to improve, although at the cost of a larger computational cost.

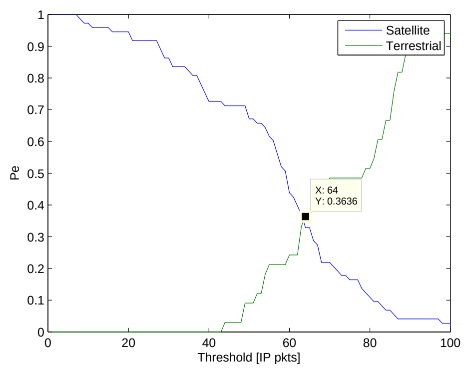

Fig. 4. Classification error probability when the threshold-based classifier using statistic 1 is used.

C. SVM classifier set-up

In the current work, whenever SVM classifiers are used, they are applied to the block-split version of the statistics in Tab. I. Furthermore, the reported performance is obtained by using cross-validation, i.e., those samples which are not in the training set are used for testing the performance of the SVM system.

At this point, the choice of the block-size L introduced in Sect. III must be discussed. The value of L was chosen trying to increase the number of samples, but without penalizing their quality from a statistical point of view, i.e., without increasing the classification error probability. Based on exhaustive tests, a value of $L = 300$ has been observed to achieve a good trade-off between both objectives. Thus, that was the value used in all the subsequent results. Moreover, the Radial Basis Function kernel is used, and its characteristic parameters C and σ are chosen to optimize the results for both scenarios studied in the next sections [8].

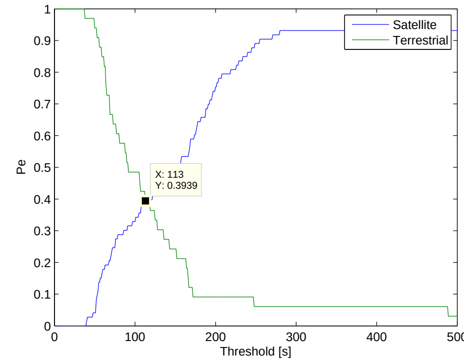

Fig. 5. Classification error probability when the threshold-based classifier using statistic 7 is used.

Finally, 500 realizations are considered for the training and test sets partition, as the SVM results could be very sensitive to the chosen sets. In any case, the variability of the system performance with respect to its average value will be also reported in the next sections, as the standard deviation values will be provided.

IV. Time-Shifted Streaming Classification Results

This section reports the results obtained by streaming the video programs within each of the TSs shown in Tabs. II and III; for each of those TSs, all the available video programs but the encrypted ones are streamed, as an appropriate decoder will be required for streaming them. The resulting number of considered streamed satellital programs is $N_S = 102$, while its terrestrial counterpart is $N_T = 33$.

Satellites	Orbital Pos.	Frequency (GHz)	Polarity
Hotbird	13°E	11.411	H
Hotbird	13°E	11.542	V
Hotbird	13°E	10.873	V
Hotbird	13°E	11.804	V
Hotbird	13°E	10.992	V
Astra	28.2°E	10.906	V
Astra	19.2°E	11.954	H
Astra	19.2°E	12.110	H
Hotbird	13°E	11.138	H
Eurobird	9°E	11.919	V
Astra	19.2°E	11.837	H
Hotbird	13°E	12.149	V
Astra	28.2°E	10.773	H
Hotbird	13°E	11.785	H

TABLE II
DVB-S Transport Streams used for the experiments reported in Sect. IV.

Digital Channel	Frequency (MHz)
39	618
63	810
45	666
58	770
48	690
68	850
54	738
66	834

TABLE III
VIGO AREA (SPAIN) DVB-T TRANSPORT STREAMS USED FOR THE EXPERIMENTS REPORTED IN SECTS. IV AND V.

Following the methodology described in the previous section, and using a pseudo-random permutation at block level for defining the training and test sets, the obtained results are as follows (P_e^S stands for the probability of wrongly classifying an IPTV stream generated from DVB-S as being DVB-T-generated, while P_e^T is the probability of wrongly classifying an IPTV stream generated from DVB-T as being DVB-S-generated; P_e is the average classification error probability):

- SVM Classification System:
  Cross-validation over blocks.
  Training: 410 blocks.
  Test: 276 blocks.
  Kernel function: RBF.
  Parameters: $\sigma = 0.7$, $C = 100$.
  - Results averaged over 500 realizations:
  P_e^S: **17.4%**, ($\sigma_{P_e^S} = 3.54\%$).
  P_e^T: **16.6%**, ($\sigma_{P_e^T} = 3.72\%$).
  P_e: 17.0%, ($\sigma_{P_e} = 2.21\%$).

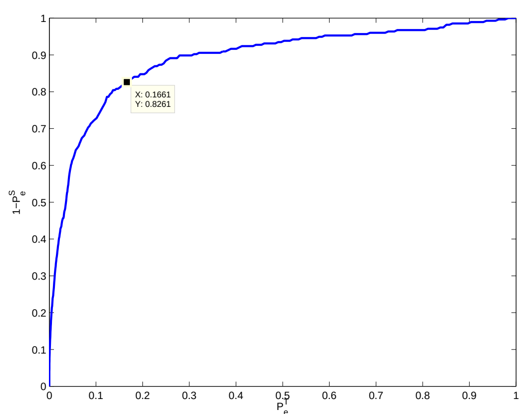

Fig. 6. ROC curve for time-shifted streaming classification.

The Receiver Operating Characteristic (ROC) curve shown in Fig. 6 might be helpful to achieve a more in-depth evaluation of the classifier system performance. Note that the marker on the curve indicates the operating point reported in the results.

For the programs used in this section, the average observation time necessary to obtain one L-block is 50.35 s, while the maximum of that magnitude is 95.74 s.

V. LIVE STREAMING CLASSIFICATION RESULTS

Concerning the live streaming results, we have used the DVB-S TSs listed in Tab. IV, while those used as DVB-T representatives are contained in Tab. III, i.e., those also considered for the time-shifted scenario (although at different time intervals). Therefore, the number of streamed programs taken into account the reported live streaming tests is $N_S = 73$ and $N_T = 33$.

Contrarily to the procedure used in the last section, and given the live streaming peculiarities (basically, from an application point of view in this scenario it does not make sense for a given video to be simultaneously placed in the training and test sets), in this case we have decided that the permutation used for defining the training and test sets will be applied at program level, i.e., the analyzed programs are partitioned into training and test sets. In other words, the training set will include all the block-based empirical statistics related to a program, or they will be included in the test set, but block-based empirical statistics corresponding to the same program cannot be found in both sets for the same experiment. By considering this program partition we try to illustrate practical live streaming scenarios, where the SVM is previously trained based on captures corresponding to a set of programs, and then it is used in real-time with a different set of programs, or with the same TS programs but at very distant time intervals (e.g., days or months later).

Satellites	Orbital Pos.	Frequency (GHz)	Polarity
Hotbird	13°E	12.111	V
Hotbird	13°E	11.542	V
Hotbird	13°E	10.873	V
Hotbird	13°E	11.804	V
Hotbird	13°E	10.992	V
Hotbird	13°E	10.723	V
Hotbird	13°E	11.138	H
Hotbird	13°E	12.149	V
Hotbird	13°E	11.785	H

TABLE IV
DVB-S TRANSPORT STREAMS USED FOR THE EXPERIMENTS REPORTED IN SECT. V.

In this scenario, the obtained results are:

- SVM Classification System:
  Cross-validation over programs.
  Training: 64 programs ($\approx$ 320 blocks).
  Test: 42 programs ($\approx$ 210 blocks).
  Kernel function: RBF.
  Parameters: $\sigma = 3$, $C = 100$.
  - Results averaged over 500 realizations:
  P_e^S : **12.6%**, ($\sigma_{P_e^S} = 5.9\%$).
  P_e^T : **11.8%**, ($\sigma_{P_e^S} = 5.9\%$).
  P_e : 12.2%, ($\sigma_{P_e} = 3.8\%$).

The ROC curve corresponding to these results is shown in Fig. 7.

For the programs used in this section, the average observation time necessary to obtain one L-block is 58.91 s, while

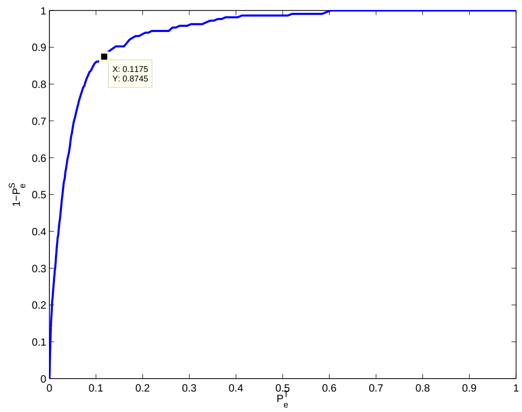

Fig. 7. ROC curve for live streaming classification.

the maximum of that magnitude is 64.51 s.

Note that the permutation procedure chosen in this case for generating the training and test sets reduces the dependency between the block-based empirical statistics in both sets, and consequently one might expect to obtain worse results that those reported in the previous section. On the contrary, the results reported here are better than those for the time-shifted scenario. We conjecture that this is due to the fact that instantaneous bitrate variations are probably more influenced by the input TS variability in the current framework, as the input TS hard disk recording and subsequent stream of just one program used at its time-shifted counterpart may remove some of the time artifacts introduced by statistical multiplexing. The empirical confirmation of this conjecture is a future research line.

It is also interesting to note that the classification error probability standard deviation is larger in this scenario in comparison with the time-shifted one. This could be explained by the program (instead of block) permutation performed in this section, which makes the variability between different realizations to be larger, similarly to what happens in statistical surveys when cluster sampling is used instead of simple random sampling in populations where the intra-cluster variance is small in comparison with inter-cluster one.

VI. Conclusions

An IPTV source classification DVB-S/DVB-T algorithm is presented for the first time in the literature. It exploits the IPTV rate changes characteristics, such as the length of the constant rate bursts, their period, and the absolute value of the difference between periods of consecutive constant rate bursts. The empirical statistics of those characteristics are fed to a SVM, that classifies the input IPTV accordingly. The proposed methodology is applied to two scenarios of interest, namely the time-shifted streaming of video contents, and its live streaming counterpart. The reported results show the feasibility of the proposed methodology for classifying the IPTV source in both scenarios, with average probability of classification errors of 17.0% and 12.2%, respectively.

Acknowledgments

Research supported by the European Union under project REWIND (Grant Agreement Number 268478), the European Regional Development Fund (ERDF) and the Spanish Government under projects DYNACS (TEC2010-21245-C02-02/TCM) and COMONSENS (CONSOLIDER-INGENIO 2010 CSD2008-00010), and the Galician Regional Government under projects ”Consolidation of Research Units” 2009/62, 2010/85 and SCALLOPS (10PXIB322231PR).

References

[1] *Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks*, ETSI Std. TS 102 034 v1.4.1, 2009.

[2] The VideoLAN website. [Online]. Available: http://www.videolan.org/

[3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A transport protocol for real-time applications,” RFC 1889, January 1996.

[4] D. Hoffman, G. Fernando, V. Goyal, and M. Civanlar, “RTP payload format for MPEG1/MPEG2 video,” RFC 2250, January 1998.

[5] *Information technology – Generic coding of moving pictures and associated audio information: Systems*, ISO/IEC Std. 13 818-1, 2000.

[6] *Information technology – Generic coding of moving pictures and associated audio information: Video*, ISO/IEC Std. 13 818-2, 2000.

[7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes*, 3rd ed. Cambridge University Press, 2007, ch. 16.5.

[8] S. Abe, *Support Vector Machines for Pattern Classification*, 2nd ed. Springer, 2010.