Capturing the Real Influencing Factors of Traffic for Accurate Traffic Identification

Géza Szabó*, János Szüle†, Bruno Lins‡, Zoltán Turányi*, Gergely Pongrácz*, Djamel Sadok‡, Stenio Fernandes‡
*TrafficLab, Ericsson Research, Budapest, Hungary, [{geza.szabo, zoltan.turanyi, gergely.pongracz}@ericsson.com]
†Complex Networks Laboratory, Eötvös Lóránd University, [szule@complex.elte.hu]
‡Networking and Telecomm. Research Group, Universidade Federal de Pernambuco, [{bruno, djamel, stenio}@gprt.ufpe.br]

***Abstract*—In this paper we introduce a novel framework for traffic identification that employs machine learning techniques focusing on the estimation of multiple traffic influencing factors. The effect of these factors is handled with the training of several machine learning models. We utilize the outcome of the multiple models via a recombination algorithm to achieve high overall true positive and true negative and low overall false positive and false negative classification ratio. The proposed method can improve the performance of every kind of machine learning based traffic identification engine making them capable of efficient operation in changing network environment i.e., when the probing node is trained and tested in different sites.**

***Keywords*-traffic classification, machine learning, packet header**

I. Introduction

In-depth understanding of the Internet traffic is a challenging task for researchers and an essential requirement for Internet Service Providers (ISP). Usually Deep Packet Inspection (DPI) is used by ISPs to profile networked traffic. Using the results ISPs may apply different charging policies, traffic shaping, and offer differentiated QoS guarantees to selected users or applications (where legally possible). DPI usually extracts information from both the packet headers and the payload. In some cases this approach is not feasible due to, e.g., processing constraint or when the payload is encrypted.

Our goal is to classify traffic based solely on packet header information, such as packet size, arrival time, addresses, protocols and ports. The following requirements have to be fulfilled by our system:

- It should be robust: the characteristics of the network, such as speed or load should not impact accuracy
- It should be accurate: results should have high true positive (TP) and true negative (TN) ratio with minimal false positive (FP) and false negative (FN) ratio

In current state-of-the-art traffic classification engines, which rely only on packet header information, the effects of network environment changes and the characteristic features of specific protocols can not be separated and are hence considered together (e.g., [1], [2]). This results in reduced accuracy when the model trained in one network is used for testing in a different one. In this paper the effects of network environment changes and the characteristic features of specific protocols are treated separately. In this way we achieved that the method performs well under changing network conditions. We propose to use multiple models to estimate the non-independent factors impacting the traffic and also their correlation. The combination of the results of these models is done automatically via a recombination algorithm. We cover the full space of the combination of evidences coming from the multiple models via using a method derived from the Dempster-Shafer theory [3]. The idea can be followed in Figure 1.

The main contributions of the paper are as follows:

- the identification of traffic influencing factors
- the introduction of specific multiple machine learning models and an algorithm to combine their outcome to support high TP, TN and low FP, FN traffic classification ratio
- a step-by-step evaluation of the methodology introduced

This paper is organized as follows. Section II overviews the related work and introduces the terms used in the paper. In Section III factors influencing the traffic are introduced and the data used for evaluation purposes is described. In Section IV-B methods are proposed to minimize the FP and FN hit ratios. We also examine how the method succeeds in differentiate the network specific characteristics from the protocol specific characteristics by repeating the comparison of [1]. Finally, the paper is concluded in Section V.

II. Related work and taxonomy

In the following bullets we define the terms used in current state-of-the-art papers about machine learning (ML).

- *Feature*: An attribute of the studied objects (e.g., the average bitrate of a flow), the input to machine learning algorithms. The algorithms aim at segment the space defined by the features as dimensions.
- *Label*: The goal of ML is to learn to categorize objects based on features. The labels are the name of the categories, hence the label is the result of the testing phase.
- *Training*: The first phase of ML algorithms, when the set of input samples are evaluated (using their features) and models are created.
- *Testing*: The second phase when the models are utilized and tested on unknown traffic to find which model describes them the best. The input to this phase is the models and the features of an unknown object (e.g., flow).

978-1-4577-2053-6/12/$31.00 ©2012 IEEE

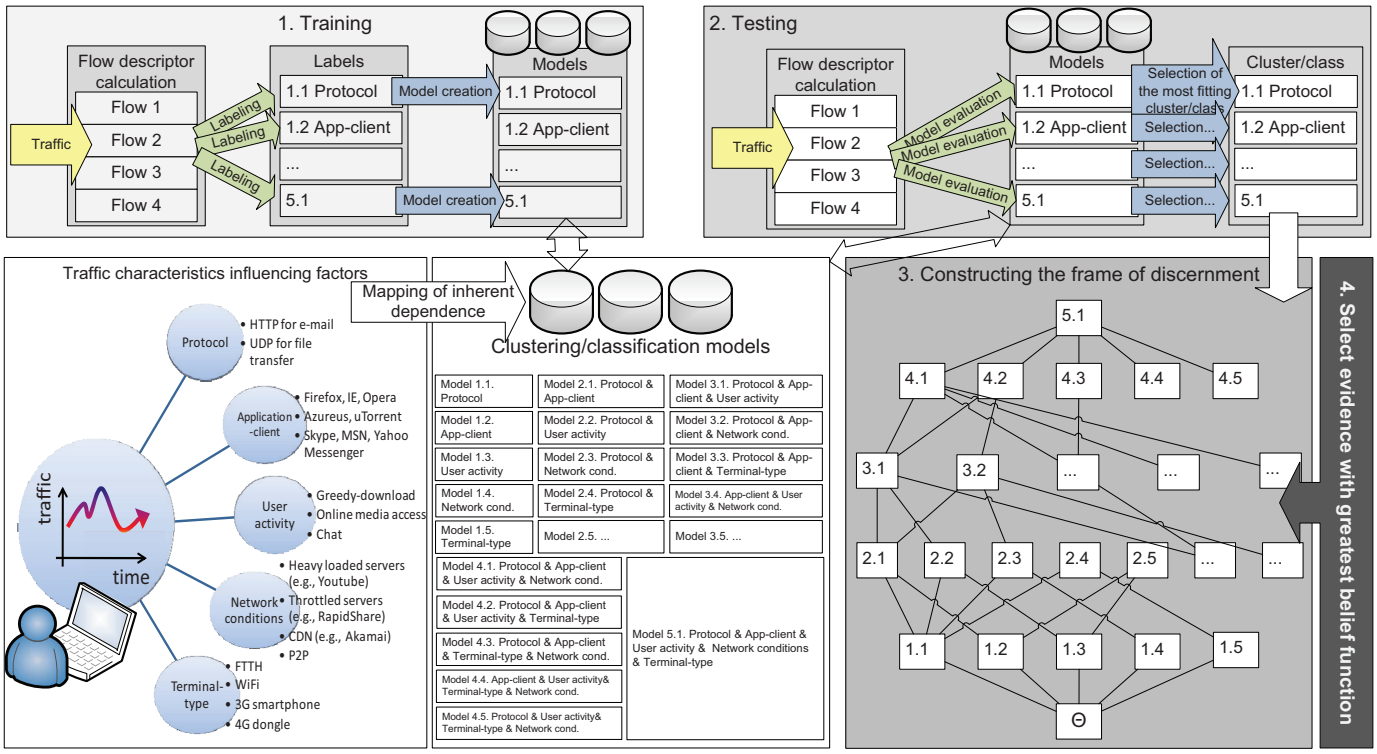

Fig. 1. The system architecture

	Features (measured)					Test result		
Flow ID	avg IAT	psize dev	sum byte	time len	Label	Classification	Clustering (hard)	Clustering (soft)
1	41	54	53	74	P2P	P2P	1	1(80%), 2(15%)
2	64	6	62	45	P2P	P2P	1	1(75%),3(10%)
3	48	80	27	83	E-mail	P2P	2	2(95%)
4	48	83	35	78	VoIP	VoIP	3	3(45%),2(9%)

Fig. 2. Example input for ML algorithms derived from network traffic

- *Accuracy*: In the test phase, what fraction of the tested objects get the proper label. Labeled test data is needed to measure the accuracy.
- *Classification* is a type of ML algorithm. Label information is used during training (along with the features) that is why it is called *supervised learning*.
- *Clustering* is another type of ML algorithm, also called *unsupervised learning*. This method automatically assigns points into clusters based solely on the features. The label information is not needed during clustering thus it makes possible to deal with new unknown applications. After clustering the label to cluster mapping function must still be defined. One approach is e.g., the most labels in the specific cluster.

Figure 2 shows an example input for ML algorithms derived from network traffic.

There is a large number of publications in the ML-based traffic identification area. Most papers usually focus on utilizing either clustering [4], [5] or classification [2], [6] methods or compare their accurracy and calculation performance [7].

Authors of [1] demonstrated that ML-based methods can offer performance similar to the ones of DPI tools when the classifier is trained for a specific site. On the other hand they also demonstrated that even if a statistical classifier is very accurate on one site, the resulting model cannot be applied directly to other locations. They showed that this problem stems from the statistical classifier learning site specific information. This is a serious problem for the practical use of ML-based techniques, because this means that a traffic identification box has to be trained locally on the new measurement site providing it with proper labeled training data e.g., with a DPI node. A pretrained probing box is preferred. In this paper we propose a mechanism to make the ML-based classifier capable of differentiating the protocol and network specific characteristics.

III. Factors Influencing Traffic Characteristics

In our work we attempt to identify effect of network characteristics and protocol specifics separately to be able to filter network effects and to arrive to a more robust identification system. In fact, we identified the following factors that influence traffic characteristics:

- Which protocol does the application used for it ('Protocol' in Table I and in Figure 1)
- Which application generated it ('Application-client' in Table I and in Figure 1)
- What is the actual functional intention of the user during the usage ('User activity' in Table I and in Figure 1)
- What was the network distance, server load, etc. ('Network conditions' in Table I and in Figure 1)

TABLE I
COMPOSITION OF MERGED TEST DATA

Protocol	flow%	User activity	flow%	Application-client	flow%	Network cond.	flow%	Terminal-type (–access capability)	flow%
BitTorrent	61.11	audio-playback	0.01	Counter-Strike	10.31	AOL	0.02	ANDROID-na	0.11
DNS	4.50	email	0.27	Internet-Explorer	48.86	Apple	0.03	BLACKBERRY-na	0.22
DirectConnect	0.06	file-download	0.01	iPhone-MPlayer	0.11	Facebook	0.28	HANDHELD-na	7.68
FTP	0.01	file-sharing	75.43	iTunes	0.11	Google	1.59	IPHONE-na	0.25
Gnutella	6.87	gaming	0.80	Limewire	4.99	iTunes	0.01	M2M-GPRS	0.04
HTTP	19.82	instant-message	2.11	Ms-Windows	0.98	Megavideo	0.01	PC-FTTH	82.25
ICMP	4.05	maps	0.01	Mozilla-Firefox	31.81	MSN	0.05	PC-HSDPA	1.43
IGMP	0.01	MMS	0.24	Opera	1.74	Microsoft	0.06	PC-HSPA	0.02
IMAP	0.03	remote-access	0.01	Skype	0.11	P2P	93.66	PC-na	7.84
POP3	0.18	social-network	0.25	Symantec	0.76	PPStream	0.01	ROUTER-HSDPA	0.15
PPStream	0.16	software-update	0.09	Twitter	0.11	Qbrick	0.03		
RTP	0.02	system	11.07	uTorrent	0.11	Symantec	0.04		
RTSP	0.02	video-playback	0.36			Tencent-QQ	4.09		
SIP	0.94	VoIP	0.01			Twitter	0.03		
SMTP	0.01	web-browsing	9.34			Yahoo	0.01		
SSH	0.23					YouTube	0.09		
Source-engine	0.53								
UPnP	0.05								
Windows	1.40								
XMPP	0.01								
Label coverage / total flow#	97.23		88.04		4.26		70.70		100

- What was the terminal and its access type ('Terminal-type' in Table I and in Figure 1)

These factors are usually somewhat correlated and in many-to-many relations. Note that the results in this paper and the performance of the methods are evaluated in these dimensions independently one-by-one.

Later in the paper the following data is used for evaluation purposes. We constructed the training and testing data in the same way as it was done in [7]. The *training data* of the system we used a one day long measurement from an European FTTH network, a 2G and a 3G network measurement from Asia and a measurement from a North-American 3G network each of them measured in 2011. We aimed at choosing measurements from networks with very different access technologies and geolocations to make the traffic characteristics varied. Flows are created from the network packet data, where a flow is defined as the packets traveling in both directions of a 5-tuple identifier i.e., protocol, srcIP, srcPort, dstIP, dstPort with a 1 min timeout. Flows are labeled with a DPI tool developed in Ericsson. The labeling mechanism is a best-effort DPI, thus it provides results in a specific factor when information was available and not all the five factors are necessarily filled for each flow as can be seen in Figure 4. The ratio of filled labels and total flow ratio for each factor is shown in Table I. The flows are randomly chosen into the training and test data set with 1/100 probability from those flows where at least one traffic influencing factor from the protocol, application-client, user activity and network condition factors[1] is recognized by the DPI tool and contained at least 3 packets. Merging all training and test data together both data sets contain 50m flows each. Table I shows the composition of the merged training data regarding the different factors. The ratios shown are calculated per factor for those flows where there was information available for that specific factor.

[1]The terminal-type is always recognized exactly from signaling traffic

IV. EFFICIENT USE OF THE MULTIPLE FACTORS

Before examining how the introduced factors make it possible to remove the network effects on protocol characteristics we have to deal with the false positive and false negative hits of the five factors. First we examine how the basic ML algorithms can perform with the above (see Section III) introduced factors on the data mixed from several networks (Section IV-A). Later we introduce methods to minimize the FP and FN ratio (Section IV-B,IV-C,IV-D). The effect of these introduced steps is evaluated with the mixed data set thus we can avoid that the evaluation would be network dependent. Finally we show how the introduced method performs in network dependent conditions and how effectively it can eliminate these effects from the ML-models (Section IV-E).

A. Individual model performance

We created five different ML-models in five different runs for each factor. During the labeling only the available label of a specific factor is used for each flow. We made experiments with the implemented clustering and classification algorithms in [8] with several parameter settings. In case of clustering methods the mapping of a specific cluster to an application is a majority decision, e.g., if in the training phase `Cluster_5` contained 100 Gnutella flows and 10 POP3 flows than during the testing phase if a flow happen to fall into `Cluster_5` it is considered P2P. The features we used are the total set of features, which were mentioned in the related work in Section II (e.g., [2], [9]) and the feature reduction in [10] were applied on them.

In Figure 3 column 'Clustering with majority decision' and 'Classification' we show one clustering and one classification algorithm, which gave the maximum accuracy. These algorithms were the Expectation Maximization (EM) and the Support Vector Machine respectively. The accuracy measures the ratio of correctly classified flow number in terms of the specific factor (TP value). Only those hits are considered

in the evaluation which had pair in the original label. The requirement of TP hit is the label and hint equality for a given factor. FP hit occurs when the hint is not equal to the label. Those flows which have no label information for that specific factor are ignored.

We found that algorithms mainly differ in learning speed and in the number of parameters which has to be set (same conclusion in [7]). Our introduced method is independent from the selection of ML-algorithm thus we select the EM algorithm with majority decision for the later experiments. It can be seen that the protocol factor can be learnt with 84% accuracy. The other dimensions are more difficult to learn by the ML-algorithms and the accuracies vary from 60-72%.

B. How to decrease the FP, FN ratio?

As testing algorithms always return hints, we end up with hints for all the factors even if there were no labels for that specific factor in the training data. This is not a preferred output of the algorithms as there can be invalid hint combinations e.g., BitTorrent for the protocol and chat for functionality. The preferred outcome is that the algorithm can learn the case when it should not return with hint.

From now on we define TP hit in case of label and hint equality. FP hit occurs when the hint is not equal to the label considering also those cases when there was no label at all. TN hit occurs when there were no hint and no label as well. FN hit takes place when there is no hint even though there was a label. In short, we want high TP and TN values (guessing it right if there is a label and what it is) and low FP and FN values.

Switching to these definitions, comparing to the last 'Clustering with majority decisions' case in Figure 3 the results changed in the 'FP redefinition' column. As the algorithms still provide hints for every flow there is no TN and FN at all, but the FP increases a lot for all the factors, since we include a lot of flows with no labels (all of them FP). Note that basically the FP ratio changes but as the number of flows where the ratio is calculated also increases the TP drops for this reason as well.

One trivial way to solve this issue is to introduce a threshold for the confidence intervals from which we accept the hint of the given model. This method would give some sort of solution to make the system capable of returning TN and FN hints. On the other hand the system would be still able to return non-conform hint constellations e.g., protocol hint is SMTP but functionality hint is file-sharing.

To solve the above issue it is possible to expand the system by training a classification algorithm with the possible hint constellations as input and label information as output. A C4.5 classifier [8] was applied for this purpose. A problem with this approach is that not all possible constellations may occur in the input data for which the system should provide response. In case of a new input hint constellation the system may not know how to respond. Further, in classification trees there is no semantic knowledge about the relation of the dimensions, the preferences of the user who interpret the results. In Figure 3

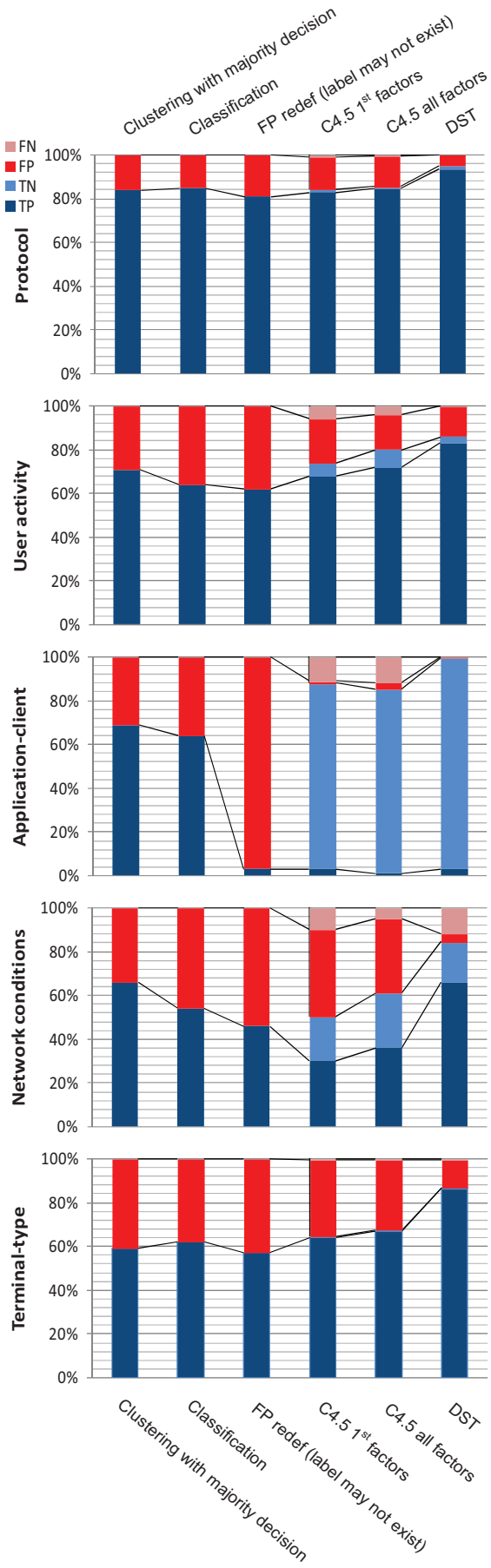

Fig. 3. The summary of the accuracy in case of the application of the proposed improvements

	Features		Labels given by DPI engine					Constructed labels		
Flow ID	avg IAT	psize dev	Protocol	App.-client	User activity	Network conditions	Terminal type	Model 3.1	Model 3.2	Model 4.4
1	34	30	BitTorrent	uTorrent	File-sharing	P2P	DSL	BitTorrent.uTorrent.File-sharing	BitTorrent.uTorrent.DSL	BitTorrent.File-sharing.P2P.DSL
2	83	58	BitTorrent	Azureus	File-sharing	P2P	FTTH	BitTorrent.Azureus.File-sharing	BitTorrent.Azureus.FTTH	BitTorrent.File-sharing.P2P.FTTH
3	49	29	POP3	MSOutlook	E-mail	?	WiFi	POP3.MSOutlook.E-mail	POP3.MSOutlook.WiFi	POP3.E-mail.?.WiFi
4	1	59	Skype	Skype	VoIP	P2P	Smartphone	Skype.Skype.VoIP	Skype.Skype.Smartphone	Skype.VoIP.P2P.Smartphone
5	71	72	DNS	?	?	?	USB dongle	DNS.?.?	DNS.?.USB dongle	DNS.?.?.USB dongle
6	90	26	?	Skype	File-sharing	P2P	DSL	?.Skype.File-sharing	?.Skype.DSL	?.File-sharing.P2P.DSL
7	2	27	HTTP	iExplorer	?	Youtube-CDN	DSL	HTTP.iExplorer.?	HTTP.iExplorer.DSL	HTTP.?.Youtube-CDN.DSL

Fig. 4. Label construction for the multiple model approach

the 'C4.5 1st factors' column shows this case. This is the first time when TN and FP ratio can be calculated. The high FN ratio shows that the C4.5 classifier has difficulties learning those cases when it is not necessary to provide results.

C. Multiple ML-models

We propose to use multiple models to automatically provide the mapping among the non-independent factors of the traffic. The possible trade-off is that the more we may know about the original five factors, the less accurate the models are. This effect can be due to the decrease of sample size and the more likely overfitting.

The multiple dimensional label information is captured in the following way (see Figure 1, 4): The input for e.g., 3.1 model is the feature set and the constructed label set for the 3.1 model. A model is trained with labels constructed from those rows which have all the required fields filled in. Labels with missing fields are omitted from the training set. Note that increasing the number of required fields, the size of training data may decrease (white background fields show the unusable labels).

D. Selection from the models

The selection of the best model is done automatically via a recombination algorithm. We can define several strategies to determine the final result from the multiple models.

1) C4.5: In this case all possible hint combinations are fed to the classification method as separate feature in the training data (as an example see the models in Figure 4). The expected response is the original label for the flows.

In Figure 3 'C4.5 all factors' row shows that the introduction of multiple models results in TP gain which is due to the reduction of the FN ratio. It means that C4.5 classifier can more efficiently learn at which cases it should provide results if the results of more dimensions are also available.

2) Demspter-Shafer Theory: Since we aim to cover the full space of the combination of evidences with the multiple clustering models and their combinations, we found that the Dempster-Shafer theory provides an excellent way to combine the results. Authors in [11] have already used a similar approach successfully on the combination of several traffic classification methods.

The Dempster-Shafer theory (DST) is a mathematical theory of evidence, an extension developed by Glenn Shafer [3], under an initial theory introduced by Arthur Dempster in [12]. The DST is viewed as a mechanism for reasoning under (knowledge) uncertainty. The part of the DST which is of direct relevance to our work is the *Dempster's rule of combination*.

The central idea of using DST for this work is to use the combination of evidence generated by sensors, in order to be able to determine a common knowledge. For example, given a sensor P_1 and P_2, these systems can identify a particular flow is classified as Torrent and P2P as the results of probing the flow features against model 1.2 and 1.5., respectively (see Figure 1). These classifications are to be taken as further evidence of our set of evidence (a.k.a Frame of discernment Θ). Considering all evidences for a given flow, it is possible to identify the mass of belief of all the elements of frame of discernment and then to determine the element most likely range of belief. Unlike the Bayesian Network theory, the belief is not assigned to a particular subset of the frame of discernment A, thus cannot be attributed to complement this subset A. The trusting in the unknown is always divided by the frame, including the own subset A. Thus it is possible to determine, for any subset of Θ, with belief not null, a range of belief that determines the belief assigned to certain evidence and every belief that can still be assigned to it. Then, after examining the evidence, it is possible to achieve a more appropriate inference for this knowledge set.

The belief value in our system is provided by the clustering method calculated from the distance between the tested element and selected cluster centroid with the $bel = (1 + e^{-dist})^{-1}$ normalization formulae. The normalization is an important step as the distance metrics in the defined multiple models have completely different meaning due to the differing feature set (as feature selection is done per model basis). Normalization makes it possible to put the distances in the same order of magnitude.

Choice of the best model is made by evidence generated by each sensor. For each flow in the input data, DST has to analyze what are the factors of this flows (shown in Table I). With the factors of a given flow at hand, it can generate a frame of discernment with all possible combinations of those factors. From the frame of discernment for a given flow it is necessary to evaluate each rating regarding such flow. Each sensor presents the inferred model, and the mass belief to the analysis. Once all the all evidences are considered, it is possible to extract from the frame of discernment which combination achieved the highest belief level. After a more appropriate model is selected it is possible to find the sensor responsible for such model. At the end of the process, we have the appropriate model and the level of classification of

		Testing data															
	Protocol	Europe FTTH				Asia 2G				Asia 3G				N America 3G			
	Model type	Single		Multiple		Single		Multiple		Single		Multiple		Single		Multiple	
	[%]	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
Training data	Eu FTTH	85	15	87	12	56	44	58	41	8	92	48	51	40	60	43	56
	Asia 2G	24	76	81	19	82	18	83	15	15	85	71	28	68	32	89	9
	Asia 3G	15	84	81	18	65	35	76	20	90	10	95	4	54	46	86	12
	NAm 3G	16	84	79	21	75	25	84	14	15	85	74	25	89	11	89	10
	Mixed data	85	15	95	5	73	27	71	25	23	77	85	14	57	43	76	22

Fig. 5. Comparison of the protocol identification accuracy of the single and multiple model case in function of the source of the training and testing data

this model.

DST shows significant improvement on C4.5 by increasing the TP hit and practically eliminate the FP and FN ratio in all dimensions (see Figure 3 in 'DST' column). Note that the high performance of DST is due to the high quality data which is provided from the multiple clustering models.

E. Evaluation of the effect of the application of multiple models

In theory our proposed method filters out the protocol independent noise from the features of the training data, thus the accuracy of the multiple model case is at least as high as the single model case by definition. To evaluate the effects of the introduction of multiple models in practice we divided the measurements of different networks into training and testing set. Once the training data of one network was used to train the single protocol model and tested on the testing data of other networks (like in [1]). In a second run all the multiple models were trained and the DST was applied to give the finest hint combination for a specific flow and only the protocol field of this combination was used for the comparison to the single model case. The results are collected in Figure 5. It shows that the multiple model case outperforms the single model considering the TP ratio in every case.[2] When the effect of the network is high on the protocol e.g., in the case of training on a 2G and testing in a FTTH measurement then the difference is even higher in the single and multiple model cases. The 'mixed data' row shows the importance of the high quality training data. The mixed data (see Section III) contains high number of samples from several networks and the TP ratio of even the simple model case is higher than those cases when the network was trained from one simple network.

F. Further utilization of DST

The proposed system would also fit to extend the information coming from DPI nodes. Fro example the analysis of packet level information with DPI can only provide that the user used HTTP protocol, but our proposed system can extend this information with user activity factor which shows that the user browsed some kind of social-networking site. The current working mechanism of the system is quite the same as described, only at the construction of the frame of discernment those model results which contained nonconforming results with the output of the DPI e.g., model 3.2 provided the 'BitTorrent.uTorrent.File-sharing' result are excluded. The DST can handle the missing beliefs and also can set the belief of the models derived by the DPI engine to high values, finally providing results e.g., 'HTTP.social-networking'.

[2]Note that in the multiple model case the TN and FN values are not presented thus the sum of TP and FP is not 100

V. Conclusion

In this paper we introduced several steps to improve the current state-of-the-art in traffic identification engines relying entirely on packet header data and make them capable of efficient operation in changing network environment i.e., when the probing node is trained and tested in different sites. We identified the effects of network environment changes and we proposed multiple ML-models to automatically provide the mapping among the non-independent factors impacting the traffic characteristics. The selection of the finest model is done automatically via a recombination algorithm. We found that the Dempster-Shafer theory provides an excellent way to combine the results. The final gain in accuracy over a simple clustering based traffic identification is 20-30% resulting in an overall 95-98% hit ratio. The FP ratio is practically eliminated (1%) and the FN ratio is as low as 0.25-8%. We achieved that the method performs well under changing network conditions.

References

[1] M. Pietrzyk, J.-L. Costeux, G. Urvoy-Keller, and T. En-Najjary, "Challenging statistical classification for operational usage: the adsl case," in *IMC '09: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. New York, NY, USA: ACM, 2009, pp. 122–135.

[2] A. W. Moore and D. Zuev, "Internet Traffic Classification Using Bayesian Analysis Techniques," in *Proc. SIGMETRICS*, Banff, Alberta, Canada, June 2005.

[3] "Shafer, G., A Mathematical Theory of Evidence," Princeton University Press, Princeton and London, 1976.

[4] A. McGregor, M. Hall, P. Lorier, and A. Brunskill, "Flow Clustering Using Machine Learning Techniques," in *Proc. PAM*, Antibes Juan-les-Pins, France, April 2004.

[5] J. Erman, M. Arlitt, and A. Mahanti, "Traffic Classification Using Clustering Algorithms," in *Proc. MineNet '06*, New York, NY, USA, 2006.

[6] F. Palmieri and U. Fiore, "A nonlinear, recurrence-based approach to traffic classification," *Comput. Netw.*, vol. 53, pp. 761–773, April 2009.

[7] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, pp. 5–16, 2006.

[8] "Weka 3: Data Mining Software in Java," http://www.cs.waikato.ac.nz/ml/weka/.

[9] A. Moore, M. Crogan, A. W. Moore, Q. Mary, D. Zuev, D. Zuev, and M. L. Crogan, "Discriminators for use in flow-based classification," Tech. Rep., 2005.

[10] J. H. Plasberg and W. B. Kleijn, "Feature selection under a complexity constraint," *Trans. Multi.*, vol. 11, no. 3, pp. 565–571, 2009.

[11] A. Callado, J. Kelner, D. Sadok, C. A. Kamienski, and S. Fernandes, "Better network traffic identification through the independent combination of techniques," *Journal of Network and Computer Applications*, vol. 33, no. 4, pp. 433–446, 2010.

[12] "A.P. Dempster, Upper and lower probabilities induced by a multivalued mapping," Ann. Math. Statist., 1967, pp. 325-339.