Future Network and MobileSummit 2012 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2012
ISBN: 978-1-905824-29-8

Network Management without Payload Inspection: Application Classification via Statistical Analysis of Bulk Flow Data

Barış KURT[1], A. Taylan CEMGIL[1], Muhittin MUNGAN[2], Neval POLAT, Alper ÖZDOĞAN[3], Ece SAYGUN[3]
[1]*Department of Computer Engineering, Boğaziçi University, P.K.2, 34342, Istanbul, Turkey*
Email: {taylan.cemgil,baris.kurt,neval.polat}@boun.edu.tr
[2]*Department of Physics, Boğaziçi University, North Campus, KB Building, 34342, Istanbul, Turkey*
*Email:*mmungan@boun.edu.tr
[3]*Ericsson Turkey, Uso Center NO:61, 34398 Maslak, Istanbul, Turkey, Email:* {alper.ozdogan,ece.saygun}@ericsson.com

Abstract: We describe a statistical approach to application classification from network traffic flows. The packet payloads are not investigated, instead we just derive easy to collect statistics such as packet size, download/upload direction, protocol and interarrival time, along with ip-number:port pairs. Each flow is modeled by a mixture of Markov models. We employ a nonparametric Bayesian approach to identify flow clusters. An important feature of our clustering method is that we don't have to specify the number of clusters in advance and the model is able to infer new flow types in an unsupervised manner. We illustrate our approach on a real dataset collected from live traffic.

Keywords: Bayesian nonparametric clustering, Dirichlet process mixtures, network application classification, flow based traffic classification, Markov models for network flows

1. Introduction

New generation wireless technologies enable operators to provide broadband coverage. Especially with the introduction of LTE and smart phones, network management for data traffic is becoming a harder problem everyday. Data traffic is increasing rapidly and network operators cannot respond fast enough to demands for the capacity increase. The expectation is that demands on the infrastructure will be comparable or even exceed the current utilization of conventional fixed broadband connections. The trend is already clear, data transmitted in networks for mobile users is increasing fast and the operators need to find ways to reduce their investment per traffic. It is therefore more important than ever before to observe the network utilization and take necessary actions in terms of maintaining QoS per application, hence optimize the network usage for improved customer satisfaction and still remain profitable.

1.1 Application Types and Traffic Identification/Classification

Network operators need to understand the application types and the type of traffic they generate so that they can prioritize types of traffic they choose, track the utilization of their network and determine the mobile data usage characteristics of their subscribers to improve their service. Hence, network traffic identification is an important and

challenging problem as it plays a crucial role in network management, in particular quality of service, security, and trend analysis. It is hence necessary that efficient data driven computational techniques are developed for classification of user and traffic types.

Current traffic identification methods can be grouped in roughly four categories:

- Payload-based (Requiring DPI),
- Flow-based (packet size, stream features),
- Host-based (analyze the behavior of the host)
- Network-wide (analyze the connectivity patterns of multiple hosts engaging in communication)

Methods from all categories have their potential limitations and drawbacks, especially for detection of novel and emerging applications that create their unique traffic patterns.

In order to identify selected type of network usage relevant to their business needs, network operators are currently using *payload based* identification techniques, such as deep packet inspection (DPI) tools. Such tools analyze packet content and bit streams to infer the type of packet being transmitted. If configured correctly,they can identify a flow with minimum ambiguity by signature matching and the resulting classification accuracy is typically quite high. Whilst possible in principle, the amount of investment required to inspect all the traffic in detail is huge. Moreover, identifying the signatures is not always easy, as new applications types constantly emerge, the content is often encrypted, and providers seek new ways to fool package inspections. Such approaches therefore require constant maintenance and in practice deep packet inspection mechanisms deployed by the operators are configured for detecting flows important for the operator's business leaving a high fraction unidentified. This observation suggests that some form of Bulk Data Analysis may prove to be very useful to classify the total flow data into coarse application types such that the network operators can identify the main trends in order to see "the big picture". This does not necessarily need to be done in real time, or in an online fashion. Rather, employing statistical techniques, it is often possible to identify application types from a tiny fraction of the data that is being transmitted.

The *flow-based techniques* depend on bulk data processing where easy to measure stream properties such as packet size, packet inter-arrival time, packet direction (upload/download) or protocol tag are available to profile and classify a given stream. Such features are easily obtained from packet headers so that the actual payload can be discarded entirely.

The *host based* and *network-wide methods*, rather than looking directly at flow based features, investigate the source and destination pairs and analyze the underlying structure that can often be represented as a graph [1, 2]. Such a graph based analysis captures intrinsic behavior of a user (or a destination such as a web service) by also looking at 'who interacts with whom'. Consequently, the resulting methods are potentially more robust against simple alterations of the flow based statistics and are harder to fool. However, these techniques require more extensive data collection and are viable only at the backbone, where most of the streams are observed.

2. Objective

In this paper, our goal is to develop methodologies that are robust to manipulations in flow characteristics and do not require inspection of the payload, *i.e.* without DPI in order to detect a broad range application types that generate a certain group of packets, such as video, peer-to-peer file sharing, gaming. Traditionally, an application could be inferred simply via investigation of the IP-port number, however this approach is no longer effective as there are now many different kinds of network applications, some of which deliberately change their behavior in order not to be detected, most notably file sharing or data streaming applications based on peer-to-peer protocols. Another difficulty is that due to privacy requirements and computational burden, it is desirable that classification algorithms are allowed to use only partial information present in the network data and avoid deep packet inspection (DPI).

While there are several different approaches to traffic classification [3, 4, 1], in this work, we are interested in *flow-based clustering* [5, 6, 7]. We are going to use flow statistics between two communicating pairs, ignoring everything else such as the payload of the network packages, DNS information, connectivity patterns of hosts, etc. We collected our real data by monitoring our own network activity. The individual flows labels in the real data may not be known, but we know which flows belong to which application.

3. Methodology

In this section, we will investigate a selected subset of most relevant approaches for flow analysis. More exhaustive surveys can be found in [8] and [9]. Flow-based analysis depends on the features extracted from connection streams (flows) between two communicating parties. A flow is characterized by the IP addresses and port numbers of source and destination nodes together with the protocol they use for the transmission. Unlike payload based methods, flows features can be extracted without the need of the inspection of the pay-load, therefore it is not affected by the privacy and encryption issues. Some of the flow features that are commonly used are flow duration, port numbers (for UDP and TCP), number of packets, maximum, minimum and average packet sizes, packet inter-arrival times (mean and variance) or average flow rate. However a given flow must be first extracted from a captured stream of packages before its features can be extracted. It is easy to identify a TCP flow, since TCP is a connection based protocol which starts and ends with predefined signals. However identifying UDP flows requires more attention but can be done by matching two host-ip:port-number pairs.

3.1 Statistical Approaches to flow-based analysis

Recent attempts for the flow-based analysis include some unsupervised, supervised and semi-supervised machine learning algorithms. Moore and Zuev [6] uses a supervised Naive Bayes algorithm together with Kernel Density estimators. Additionally they employ Fast Correlation-Based Filter for feature selection, which. identifies the important or redundant flow features among all. Their data come from the monitoring of the internet traffic of a research center for 24 hours over a period of one week. Only header information is stored and protocols other than TCP is disregarded. For the supervised learning, they manually classify *TCP flows* into categories such as *bulk, database, interactive, mail, services, www, P2P, attack, games, multimedia*.

Erman et. al. [10] compare the supervised Naive Bayes algorithm with an unsupervised Expectation Maximization based clustering algorithm (AutoClass). They report that the unsupervised algorithm outperforms the supervised one. Although they use the same features used in Moore's work [6], these features are extracted from a different data set. This data set was collected at the servers of the University of Auckland for 72 hours in 2001 and is publicly available. However, it is out-dated since many services have changed their behavior since then, for example P2P services started to use randomly changing port numbers.

Rotsos et. al. [5] give a custom probabilistic graphical model for the flow-based traffic classification. They use a semi-supervised approach by labeling a small portion of the data set manually, and using both labeled and unlabeled data, since in most cases one has to deal with large sets of flows without being able to label them. They employ a Naive Bayes training method and use an Expectation Maximization algorithm to infer the model parameters as well as the categories of the unlabeled flows. This line of research is relevant for our line of work from a methodological perspective as it enables one to use partially labeled data.

3.2 Overview of Machine Learning Methodology

In recent years techniques form Bayesian statistics became very popular in diverse fields such as machine learning, bioinformatics, finance and signal processing. The common aspect in all these applications is the presence of noisy data and the uncertainty regarding the underlying data generating process. Moreover, plenty of expert knowledge is available, however it is not clear how to incorporate this into a rigorous statistical framework. Bayesian techniques offer an elegant solution to this problem by the use of probabilistic models in a general and well-defined computational framework. A Bayesian model consists of two components:

- a prior distribution summarizing expert knowledge in terms of unobserved parameters,
- a likelihood component which describes the conditional probability of observed data given a particular setting of parameters.

By calculating the posterior probabilities of the parameters one can infer desired information about the data generation process as well as carry out model comparison and selection.

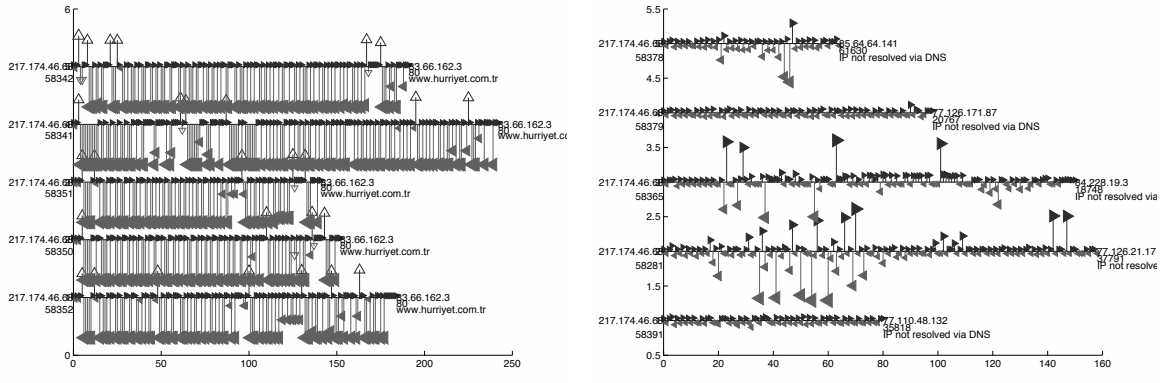
We are using a ***Bayesian*** machine learning approach for traffic analysis and the clustering of services and users. In the Bayesian approach, a model with a set of parameters is proposed to explain the behavior of the data. Among all possible parameter sets, the one with the highest probability of explaining the data is selected. For example, using stochastic blockmodels one can model users and services according to usage statistics. In the Bayesian framework this is done as follows: let D be a matrix that describes the observed usage amounts of services vs users. Let U and S be category assignment of all users and services, and B be the parameter showing the usage trend of users in a user category to the services in a service category (for all user-service category pairs). The modeling step involves introducing a conditional probability of observing D given U, S and B and the best category assignments and usage parameters B^*, U^*, S^* are determined as those that maximize this probability. Moreover, if more than one model

are proposed for a given data set, one can compare the expectations of observing the dataset under the given models. The expectation of observing data under a model is calculated by integrating the probability of observing D for a give parameter set over all possible parameter sets by taking the prior probabilities of parameter sets into the account. For example in the above scenario, we could have to decide between two models which try to divide users and services into different number of categories.

Quite generally approaches as described above require exhaustive search among all possible parameter sets for parameter estimation, and integration over parameters when the goal is model selection. These operations may be intractable to compute or there may be no easy mathematical closed-form equations for them. In such cases one employs approximation methods such as *Expectation-Maximization* or *Gibbs Sampling*.

4. Model

We describe the network flows as Markov processes and our basic assumption is that similar flows are different instances generated by Markov process with the same parameters, while distinct flows come from Markov processes with a different set of parameters and are therefore classified as belonging to different groups/clusters. Thus as we inspect flows one by one, a decision has to be made whether the given flow is member of one of the clusters which we have already established from inspection of the previous flows (and if so, which cluster), or whether it belongs to a new cluster. The natural probabilistic framework to model this type of approach is the Dirichlet process mixture model [11].



(a) A cluster of 5 flows with large down packets.

(b) A cluster of 5 flows with small up and down packets.

Figure 1: Visualization of two network flow clusters obtained from real data. Each horizontal line represents a flow. The x-axis represents the time, y axis represents packet size in kB and flows have been vertically offset for clarity. Black upward (grey downward) arrows designate up (down) packets. Arrow lengths represent packet sizes. It can be seen that the flows in each cluster have similar properties. In (a), the prevalence of long grey arrows shows that the flow consists mostly of down packets. In part (b), sizes of up and down packets are comparable and mostly small.

Specifically, we employed a Gibbs sampler to infer the number of clusters and cluster assignments of the flows for both synthetic and real data. A network flow f is a chain of packets $s_{1:T}$ transmitted according to a protocol between a source node and destination

node. Each packet has the following properties: arrival time, protocol, up/down flag, and size. In our simulation studies, we have only used up/down information and size, $s_t = \{up/down, size\}$. Moreover, the size in bytes is quantized into S levels, so each packet s_t is an element of a state space with cardinality $2 \times S$. Each flow is generated by a first order Markov process $p(s_t|s_{t-1};\theta)$ where θ are the model parameters (initial state distribution and transition matrix). The complete data set $F = \{f^1, f^2, \ldots, f^N\}$ is generated by a Dirichlet mixture process.

At each step, a new flow is generated by either one of the available Markov models, or a new Markov model is introduced with a probability dictated by a Polya-urn scheme. The parameter α is the concentration parameter, which affects the tendency to generate new Markov models. In the process, flows that are generated by the same Markov model form a cluster.

As the number of flows increase, the inference for the cluster assignments can become intractable, since the total number of possible ways in which N objects can be clusters grows super exponentially with N. We have therefore used a collapsed Gibbs sampler for sampling from the probability distribution over the possible ways of clustering the set of N flows.

5. Results

In the experiments done with synthetic data, we have observed that the total number of clusters formed by the Gibbs sampler is close to the ground truth, *i.e.* the number of clusters from which the synthetic data has been created, and the clusters successfully group together flows that are generated from the same Markov model.

In order to further validate our approach we captured network traffic generated by a variety of popular applications: video streaming (VIDEO), voice over IP (VoIP) and peer to peer (P2P). A DPI tool [12] was used to determine the type of each flow in the captures. We only include flows identified by the DPI tool as a protocol corresponding to the above application types. The 9681 flows obtained in this way were subsequently clustered. Figure 2 shows the cluster assignment of the flows lumped into the categories VIDEO, VoIP and P2P. The thickness of a line connecting a category with a cluster indicates the fraction of data size of a given category assigned to the category in question. The barplots to the right of each cluster indicate the fraction of category flow data size contained therein. As can also be seen from Figure 1, displaying two clusters obtained from the classification of real data, our method is clearly able to cluster flows with similar characteristics together. A straightforward classification mechanism is as follows: we cluster a flow using our method and to choose the application type acccording to the most likely application type in that cluster. Figure 2 suggests that for our dataset the cluster conditional densities for Video and P2P are rather crisp suggesting that our simple approach is able to distinguish between those applications in an unsupervised way. Our current research is directed to the investigation of semisupervised discriminative approaches and an evaluation of the classification performance on a larger test set.

6. Discussions and Conclusions

We have shown that our method distinguishes flows directly associated with different application types with high accuracy. In classifications that avoid DPI such a pre-

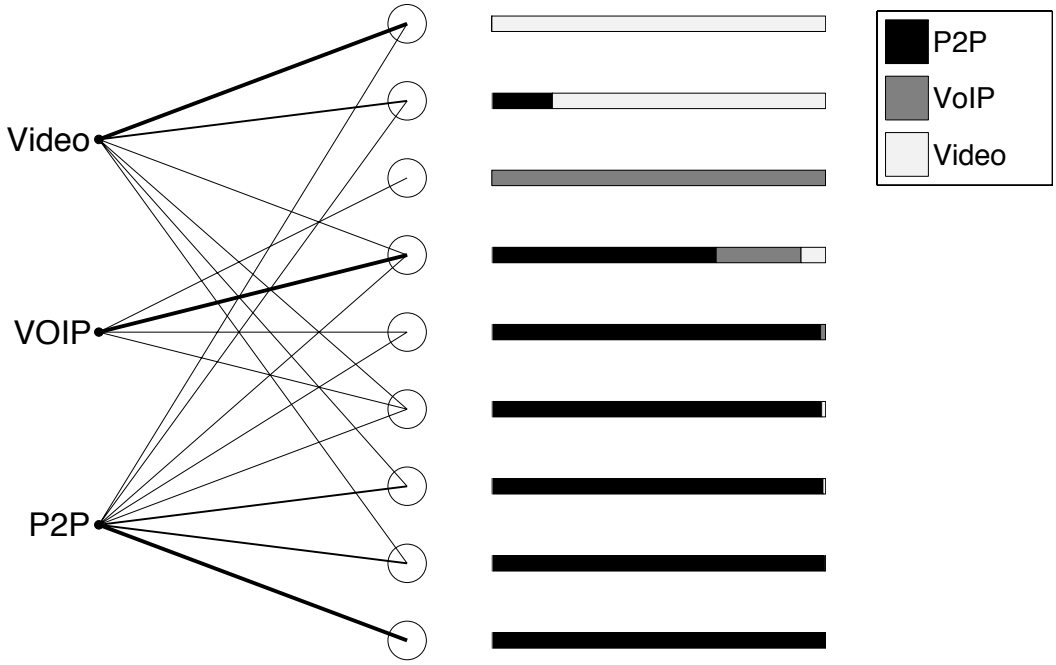


Figure 2: Result of the clustering of 9681 flows categorized as video, voice over IP (VoIP) and peer to peer (P2P), see text for further details.

selection is not available and instead all we have is a collection of flows. By selectively capturing flows associated with different types of applications and classifying these into groups we essentially establish a dictionary which may be used for type inference when new flows or collections of flows are encountered. Whenever a new type of application is observed, our nonparametric method allows for the creation of new clusters, which means adding new words to the dictionary. We thereby establish correspondences between applications and a collection of words, very similar to the indexing of documents by keywords. This type adaptivity is naturally suited for the analysis of network traffic, since new types of applications emerge frequently. Another important point is that many types of applications are not necessarily discernible by a single tell-tale flow but rather by a collection of flows triggered by them. This is particularly the case for file-sharing and generally for peer-to-peer network traffic. Thus having a framework that accommodates classification of a collection of flows as well as individual flows will have much more flexibility in its performance. We are currently working on an extension and refinement of such an approach.

7. Acknowledgments

This research has been partly supported by Ericsson Telekomünikasyon A.Ş. within the scope of research being conducted in FP7 CELTIC project MEVICO. The authors would also like to thank Ipoque GmbH for making available their PACE DPI tool.

References

[1] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: multilevel traffic classification in the dark," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 229–240, 2005.

[2] M. Iliofotou, H.-c. Kim, M. Faloutsos, M. Mitzenmacher, P. Pappu, and G. Varghese, "Graph-Based P2P Traffic Classification at the Internet Backbone," *IEEE INFOCOM Workshops 2009*, pp. 1–6, Apr. 2009.

[3] A. Dainotti, W. de Donato, and A. Pescapé, "TIE: a community-oriented traffic classification platform," *Traffic Monitoring and Analysis*, pp. 64–74, 2009.

[4] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, and G. Varghese, "Network traffic analysis using traffic dispersion graphs (TDGs): techniques and hardware implementation," tech. rep., 2007.

[5] C. Rotsos, J. Van Gael, A. W. Moore, and Z. Ghahramani, "Probabilistic graphical models for semi-supervised traffic classification," *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference on ZZZ - IWCMC '10*, p. 752, 2010.

[6] A. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pp. 50–60, ACM, 2005.

[7] A. Dainotti, W. de Donato, A. Pescape, and P. Salvo Rossi, "Classification of Network Traffic via Packet-Level Hidden Markov Models," *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, pp. 1–5, 2008.

[8] T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[9] S. Lee, H. Kim, D. Barman, C.-k. Kim, T. Kwon, and Y. Choi, "NeTraMark: a network traffic classification benchmark," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 22–30, 2011.

[10] J. Erman, A. Mahanti, and M. Arlitt, "Internet traffic identification using machine learning," in *IEEE Globecom*, Citeseer, 2006.

[11] R. M. Neal, "Markov chain sampling methods for dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[12] Ipoque, "PACE: Protocol and Application Classification Engine," *Ipoque GmbH, a Rohde & Schwarz Company*.