Flow Classification by Histograms

or How to Go on Safari in the Internet

Augustin Soule$^{\odot}$, Kavé Salamatian$^{\odot}$, Nina Taft$^{\bullet}$, Richard Emilion†, Konstantina Papagiannaki‡
$^{\odot}$LIP6-UMPC University of Paris VI, $^{\bullet}$Intel Berkeley, †University of Orléans, ‡Intel Cambridge.

ABSTRACT

In order to control and manage highly aggregated Internet traffic flows efficiently, we need to be able to categorize flows into distinct classes and to be knowledgeable about the different behavior of flows belonging to these classes. In this paper we consider the problem of classifying BGP level prefix flows into a small set of homogeneous classes. We argue that using the entire distributional properties of flows can have significant benefits in terms of quality in the derived classification. We propose a method based on modeling flow histograms using Dirichlet Mixture Processes for random distributions. We present an inference procedure based on the Simulated Annealing Expectation Maximization algorithm that estimates all the model parameters as well as flow *membership probabilities* - the probability that a flow belongs to any given class. One of our key contributions is a new method for Internet flow classification. We show that our method is powerful in that it is capable of examining macroscopic flows while simultaneously making fine distinctions between different traffic classes. We demonstrate that our scheme can address issues with flows being close to class boundaries and the inherent dynamic behaviour of Internet flows.

Categories and Subject Descriptors

I.5.1 [**Computing Methodologies**]: Pattern Recognition—*Models*; C.2.m [**Computer Systems Organization**]: Computer Communication networks—*Miscellaneous*

General Terms

Algorithms, Measurement.

Keywords

Flow classification, Internet traffic, parameter estimation.

1. INTRODUCTION

The Internet today is a large highway that is traversed everyday by packets from millions of users throughout the world. These packets together create flows that are intertwined at gateways and routers to create larger and larger aggregated flows as they travel deeper into the network. The goal of traffic engineering in IP networks is to manage these aggregated flows such that they traverse the network efficiently, problem-free and with high performance (i.e., low delay, low loss, etc). A variety of mechanisms are used to manage traffic flows such as load balancing, routing policies, traffic shaping, etc. It is useful to have some knowledge of the statistical and dynamic behavior of the flows in order to best manage them. This general goal has motivated a good deal of research on Internet traffic characterization [16, 7, 12].

Before implementing any traffic control, a network operator needs to specify which flow granularity level will be the controllable entity. Flows can be defined using a wide variety of parameters such as the type of source or destination (user, subnet, etc), the protocol controlling the flow (TCP, UDP, HTTP, etc), or the application sourcing or sinking the flow. Flows at each level of aggregation consist of lower level entities (packets, connections, sessions, etc) that may have specific characteristics that are related to the source, destination and path traversed by a flow. These flows are bundled and intertwined via different queueing and multiplexing elements from a variety of devices (firewalls, gateways, routers, etc) to create aggregated flows. Such highly aggregated flows appear not only in backbone carrier networks, but also in Tier-2 networks, corporate networks of companies with thousands of employees, and campus networks.

Aggregated flows can be rather complex and hard to characterize. They will typically exhibit macroscopic behavior that may not be trivially related to lower level behavior. In order to keep the management of highly aggregated flows relatively simple, it is useful to focus on the macroscopic behavior and to classify traffic into a small number of classes. By characterizing each class one can then have a better understanding of how to treat different traffic classes. The idea of separate treatment for different traffic classes has been considered in a number of environments such as Diff-Serv, IntServ, ATM networks, or load balancing policies for heavy flows [13]. Traffic classification is also applicable to the problem of detecting denial of service (DDOS) attacks. By studying "normal" traffic and "attack" traffic one can develop different classes for these traffic flows.

In this paper we focus on highly aggregated flows and de-

SIGMETRICS/Performance'04, June 12–16, 2004, New York, NY, USA.
Copyright 2004 ACM 1-58113-664-1/04/0006 ...$5.00.

velop a very general method for classifying them. There are two benefits of trying to classify flows. First, it enables us to learn more about the traffic in that we identify characteristics that are common to a set of flows (those within a same class), and identify those that differ from one class to another. Second, we enable a mechanism to examine any single flow and determine which class it belongs to. This in turns gives the operator the opportunity to manage flows from one class differently than those in another.

We identify two issues that need to be addressed in this process. On the one hand, it is desirable for traffic classification to use only a few classes because this helps keep the management of flows simpler. On the other hand, the macroscopic behavior may mask lower level behavior that could be important in the identification of different types of flows. Consequently, we want a classification scheme that is powerful in that it can differentiate key differences in flows, but yet one that operates at the macroscopic level since control of flows is typically carried out on highly aggregated streams.

1.1 Features and Issues in Classification

Many researchers that have conducted traffic studies, have found themselves unsatisfied by looking merely at first and second order moments. The mean of traffic flows often says very little about those flows because many aggregated flows exhibit a wide range of behavior in terms of variability [9]. Characterizing flows by computing simple variance metrics is also often insufficient because there are many sources of variability in aggregated flows, including diurnal patterns, bursty behavior and simple noise fluctuations [9]. In this paper we argue that when doing traffic classification the entire distributional properties of flows should be used.

The most common classification of flows is into the so called elephants and mice classes. The idea of discussing flows in terms of elephants and mice has arisen due to three observations; first that a small fraction of the flows (e.g., 10 traffic load (e.g., 90 that heavy flows can be two or three orders of magnitude larger than light flows; and third, that there are many flows with extremely small bandwidths (and often zero for long periods of time) [12]. Such observations lead us to know that different classes of flows arise naturally in the internet, however they do not leave us with a mechanism to observe a single flow and determine its class. This can be difficult for those flows that are “in the middle”, i.e. those that are not clearly elephants nor clearly mice. The problem of *good* classification has to do with how well those “flows in the middle” can be distinguished one from another. Flows that do not exhibit persistent behavior over time can also be difficult to classify because they may be heavy for some period of time and then become lighter. Thus the dynamic behaviour of flows further complicates the classification task. For these reasons we believe that making use of the entire distributional properties of flows will enable the best classification.

1.2 Our Method

In this work, we investigate the applicability of statistical inference techniques for identifying classes as well as deciding the likelihood that a particular flow belongs to a specific class. Our approach is as follows. We use samples of average flow bandwidths obtained from packet traces. We first convert these samples into histograms. Our classification is performed over the ensemble of histograms. By using the entire distribution we incorporate many aspects of a flows behavior, *e.g.* all order moments or tail behavior. These histograms are grouped into a small set of classes and each class is modeled via a random distribution. In particular we use as a prior Dirichlet random distributions because these distributions are very general and capable of representing a wide variety of distributions. To find the parameters of the Dirichlet distribution for each class, we use maximum likelihood estimation. Our estimation algorithm is implemented using a stochastic version of the Expectation Maximization (EM) algorithm. Our method outputs the distribution describing each class, and also an *a posteriori* probability or *membership probabilities*, i.e. the probability that a given flow belongs to each class. We apply a Maximum *a posteriori* criterion saying that a flow belongs to a particular class if the probability that it belongs to that class is higher than the probability that it belongs to any other class.

The method developed in this paper is very general for three reasons. First, it uses histograms that capture the entire range of traffic behavior and all statistical characteristics of flows (i.e., all the moments, not just the first and second moments as is usually done). Second, since the form of Dirichlet processes is a very general polynomial prior, our Dirichlet-based model can capture well flows with almost any type of distribution. Third, because we describe a class via a probability distribution, our approach is more general than schemes that rely on a tuple (e.g., mean and variance), or previous methods [12, 7] that define thresholds. In thresholding methods a flow is in one class if its mean bandwidth (for example) is above the threshold and in another class if it is below. Our approach is more general because it allows large flows to have small moments of non-large behavior but still remain classified as a large flow. In [12] the authors identified the problem of flows frequently changing between classes. By incorporating the entire histogram of a flow’s behavior, rather than a single value, the likelihood that a flow needs to be reclassified because it experiences a small moment of atypical behavior is greatly reduced.

The literature has widely used zoological references such as elephant, mice, tortoise, dragonflies, etc., [2] to name different classes of flows. We will not deviate from this tradition and we thus consider our schemes as analogous to going on a safari in which one wishes to potentially discover new animals and their behavior, or to hunt (find and identify) animals of predefined types.

One of the main contributions of this paper lies in the development of a new method for flow classification. The fundamentals of this method were developed recently in the mathematics community [8]. In this work, we adapted the method to Internet flows. This is the first time that this method has been evaluated on any dataset. We first validate the method on synthetic data in which we can know ahead of time what the “proper” classification of a flow is. Our method yielded 100 few synthetic data trials. Second, we apply our method to prefix level flows collected inside a large carrier’s backbone. We consider the scenario of two classes and confirm that our method generates not only a set of classes that matches our intuition about elephants and mice, but one that can also classify those “flows in the middle”. We then consider a scenario in which flows are separated into four classes. We see that these four classes do exhibit distinct behavior. We thus conclude that our method is capable of detecting fine differences in macroscopic flows. Finally we examine the temporal stability of our classification and illustrate that our method can handle the dynamicity of flows well.

In section 2, we present our method for creating empirical histograms from flow measurements, and our rationale for histogram-based classification. Our model and the related background material are given in Section 3. We discuss random distributions, parametric inference for these distributions and the stochastic version of the Expectation Maximization method for Maximum Likelihood estimation that is used to estimate the parameters of our model and the membership probabilities. We validate our model in Section 4. In Section 5 we apply our method to Internet backbone flows when either two or four classes are used. The issue of the stability of the classification is addressed in Section 5.3. Finally, we conclude in Section 6.

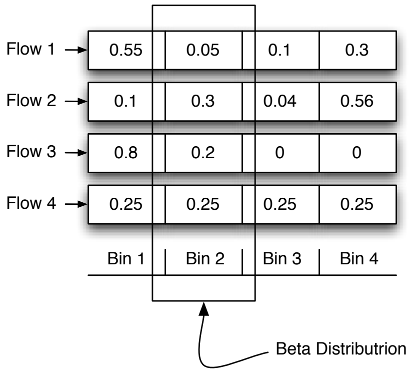


Figure 1: Converting flow data to histograms

2. HISTOGRAM-BASED CLASSIFICATION

In this section we describe the data used for classification, how we built the histograms, and our motivation for doing so. Since core routers use IP destination prefixes announced by the Border Gateway Protocol (BGP) [17] for routing purposes, the natural granularity level for inter-domain traffic engineering is that of network prefixes appearing in routing tables. We therefore define flows throughout this paper as the sequence of packets going toward a specific BGP prefix (i.e., those that appear in a BGP table). We believe that this granularity level is a natural candidate for traffic classification for a few reasons. First, such flows can easily be manipulated by simply changing the next hop address in the routing table for that particular flow. Second, routing policies tend to be applied to a network prefix as a whole, since network prefixes are the smallest routable entities in the Internet.

2.1 Data used for classification

The data used in this paper comes from packet traces collected in the core of a major Tier-1 ISP network. Optical splitters are used in conjunction with passive monitoring equipment to collect 44-byte headers from every IP packet traversing monitored links. We use packet traces, collected on July 24, 2001, from two different OC-12 links in the USA, one from an east coast PoP (Point of Presence) and the other from a west coast PoP. The links used are two hops away from the periphery of the network so that traffic toward specific destinations exhibits sufficient level of aggregation. Our traces contain 3 1/2 days of continuous data.

The packet trace collection was accompanied by the collection of the BGP routing tables at the corresponding PoPs. Those BGP tables are default-free and contain approximately 120K entries. We calculated the volume of traffic headed toward each BGP destination and computed the average bandwidth of each flow over 5 minute time intervals. We found that in any given measurement interval, approximatively 90 the network prefixes had no traffic traveling toward them. We thus define a flow to be *active* if it transmits at least one packet during the measurement interval. We found that in a typical measurement interval, approximately 2000 flows were active.

2.2 Creating histograms

Much of our work was done considering 24-hour periods from these traces. Since we measure the flows every 5 minutes, this yields 288 measurements for each flow. We consider the ensemble of the active flows and build our histograms as follows. Let $\mathfrak{B}$ denote a set of L bins, $\mathfrak{B} = \{[b_0 = 0, b_1), [b_1, b_2), \ldots, [b_{L-1}, b_L = +\infty)\}$, with $0 < b_1 < \ldots < b_l < \ldots < b_{L-1}$. Let m denote the number of measurements for the flows (e.g., 288 in the 24-hour traces). We define X_{il} to be the proportion of time among m time slots where the bandwidth of flow i was in the interval $[b_{l-1}, b_l)$. Thus X_{i*} gives an empirical histogram for flow i over the set of bins $\mathfrak{B}$. In Figure 1 we give a dummy example with 4 flows and 4 bins to help clarify our notation. Each value represents a sample value for X_{il}. Since we have $\sum_{l=1}^{L} X_{il} = 1$, we indeed have a proper histogram for each flow i. When considering all the flows together, then X_{*l} gives a vector of samples for bin l. To simplify the notation, when we write X_l we imply X_{*l}. The vector X_l gives a set of of samples on the proportion of time that an arbitrary flow transmits in the range defined by the l-th bin. An example of this vector is indicated in Figure 1 via the encircled set of values. We can thus define the vector $\mathcal{X} = (X_1, ..., X_L)$ (i.e., a vector of vectors) to represent our entire data collection. This $\mathcal{X}$ can be viewed two ways. It represents our entire set of histograms for all the flows. It also represents the bin distributions for all bins. In essence, X_l denotes the likelihood that a flow transmits in the range defined by bin l. For modeling purposes, we will make use of this second view. We will discuss the issue of bin sizing in section 5.

In Figure 2 we plot three flows and the corresponding histograms they generate. We used logarithmic bin centers due to the wide range of bandwidth that single flows can span and to ensure that no bins are empty. These three flows generate three different histograms. Although the first two flows appear somewhat similar over time, their histograms capture the differences; for example, the second flow spends more time in the 10^4 range of values than the first flow.

2.3 Motivation

In section 1 we discussed why we believe the entire distributional properties of flows should be used for classification, rather than, for example, using a tuple such as the mean and variance (or standard deviation). We illustrate here, via an example, why using the mean and standard deviation alone can be insufficient. Although we haven't described

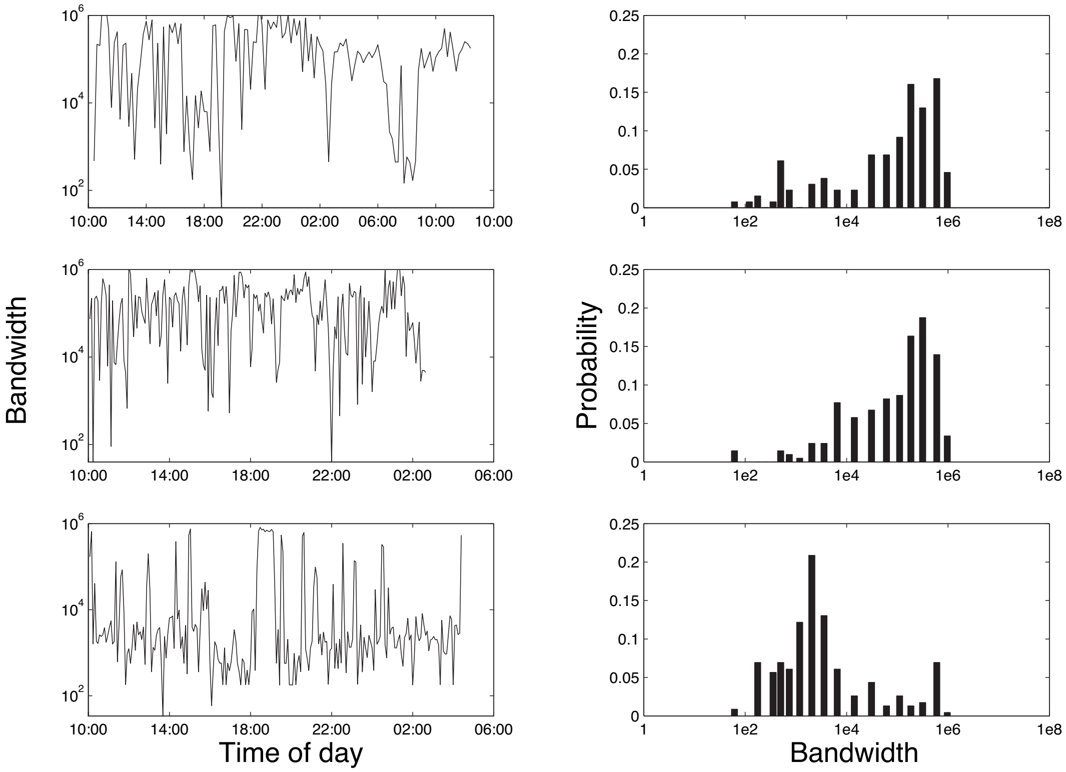


Figure 2: Temporal variation of a flow

our method yet, we use it here merely for motivational purposes. We used one of our 24-hour traces and our method for classifying the flows into two classes. With a classification of the flows into elephants and mice, we computed the mean and variance of each flow in each class. In Figure 3, we plot the standard deviation versus the mean. Each circle represents the (mean,std) tuple for one elephant flow. Each plus sign represents the tuple for a single mouse. If the elephants and mice were cleanly separable using only these two parameters, then the circles and pluses would appear as two distinct clusters on such a graph. Although many elephants are clustered toward the top right of the graph and many mice are clustered near the bottom left, there are clearly many flows "in the middle" which are not easy to differentiate.

An astute reader might postulate that there is a circular argument here. We have used our own method and shown that the (mean,std) tuple is insufficient. This could be an artifact of our method. While that might be true, our experience testing a variety of methods leads us to hypothesize that the conclusion implied by this figure holds more generally. Especially since we are using the entire distribution of the flows for classification. To check this, we also created the same type of plot for methods such as [7, 12], and simple threshold methods (such as "select largest-N flows as elephants" or "select flows contributing to top x *lack of clear clusters*. We were thus motivated to study classification using the entire distributions of flows.

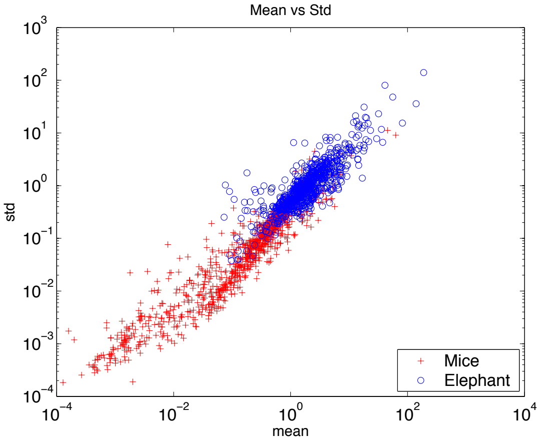


Figure 3: Two feature classification using mean and standard deviation

3. METHODOLOGY

In this work we propose to model the vector $\mathcal{X} = (X_1, ..., X_L)$

that represents the empirical histogram of a flow using a mixture of Dirichlet processes. In order to explain this we first present some background material on random distributions, Dirichlet densities and Dirichlet processes. We discuss what this type of model implies for bin distributions. We then explain why we use a mixture of such processes. Finally, we present an algorithm that does two things: estimates the parameters of the mixture model, and computes the *a posteriori* probability, *i.e.* the *membership probabilities* (the probability that a flow belongs to a class). The class assignment will use a MAP (Maximum *A Posteriori*) criterion, *i.e.* a flow is classified in class k if the probability that it belongs to class k is larger than the probability that it belongs to any other class. This can be viewed as a "soft" class assignment, as opposed to a "hard" one in which a flow can only be classified in one class. Soft assignments are intuitively more appealing in the dynamic context of the Internet as they essentially recognize that a flow may behave like a flow in another class at times.

3.1 Theoretical framework

Because we try to analyze the histogram of a flow rather than its values, we are dealing with observations that are themselves probability distributions. In other words, each flow yields a histogram that can be seen as a realization coming from a stochastic source generating random histograms. By way of analogy, we can say that random distribuions are a source that generates histograms much the same way that a random variable represents a source that generates a single value, or that a random vector represents a source that generates vectorial observations. A formal definition of random distributions is given as follows.

Definition 1: A *random distribution* (RD) is a measurable map from a probability space $(\Omega, \mathcal{F}, \mathbf{P})$ to the space $\mathbf{P}(V)$ of all probability measures defined on a fixed measurable set $(V, \mathcal{V})$. If $X : \Omega \longrightarrow \mathbf{P}(V)$ is a RD, its distribution $\mathcal{P}_X$ is then a probability measure on $\mathbf{P}(V)$.

Any parametric distribution in which the parameter is randomized is an example of a random distribution. For example $\lambda(\omega)e^{-\lambda(\omega)x}$ where $\lambda(\omega)$ is itself a random variable forms an exponential RD.

An example of a random distribution for discrete random variables is the following. Let $V = \{1, \ldots, L\}$ be a finite set, then $\mathbf{P}(V)$, the set of all probability distributions defined over the set V, can be identified to be the set

$$\mathcal{X} = (X_1, ..., X_L) : (\Omega, \mathcal{P}) \to \mathbb{R}_+^l \text{ such that} \sum_{l=1}^{L} X_l = 1.$$

.

This last RD is very useful in the context we are studying as each histogram coming from a flow is a discrete probability distribution defined over a finite set of bins. Recall that X_l is a random variable that denotes the likelihood of a flow transmitting in the range defined by bin l. Therefore this last example of an RD describes the set of histograms we have to deal with in flow classification.

The source generating random distributions is governed by a multidimensional probability distribution that defines the probability of an histogram $\mathcal{X} = (X_1, ..., X_L)$. Clearly bin sizes (X_i) are dependent of each other as they are jointly constrained by the condition that $\sum_{k=1}^{L} X_k = 1$. This means that classical distributions are not applicable in this context. The Dirichlet distribution defines a multivariate distribution on the space of random distributions that is frequently used in the context of RD.

Definition 2: A *Dirichlet distribution density*, with parameter vector $\alpha = (\alpha_1, ..., \alpha_L)$ is given by

$$f(x_1, x_2, ..., x_L | \alpha_1, \alpha_2, ..., \alpha_L) = \frac{\Gamma(\alpha_1 + ... + \alpha_L)}{\Gamma(\alpha_1)...\Gamma(\alpha_L)} \prod_{l=1}^{L} x_l^{\alpha_l - 1}$$

where $\alpha_1 > 0, ..., \alpha_L > 0$, $x_l > 0$, and $\sum x_l = 1$. This defines the joint density probability function of $(X_1 = x_1, X_2 = x_2, ..., X_L = x_L)$. We denote this Dirichlet distribution by $\mathcal{D}(\alpha_1, ..., \alpha_L)$

Remarks:

The popularity of the Dirichlet distribution is due to several convenient properties that we list here :

1. We should notice that the problem of estimating the real distribution of a flow, based on an observed empirical distribution over k bins, can be formalized as estimating the parameters $\Theta = (\theta_1, \ldots, \theta_L)$ of a multinomial distribution based on an observed empirical histogram $\mathcal{X} = (X_1, \ldots, X_L)$. Now if a random variable Z follows a multinomial distribution $\mathcal{M}(\theta_1, \ldots, \theta_L)$ with unknown parameters $\Theta = (\theta_1, \ldots, \theta_L)$, and if the prior distribution on the unknown parameter $\pi(\Theta)$ is a Dirichlet distribution $\mathcal{D}(\alpha_1, \ldots, \alpha_L)$, the posterior probability $\mathbb{P}\text{rob}\{\Theta | X = (x_1, \ldots, x_L)\}$ will also follow a Dirichlet distribution given by $\mathcal{D}(\alpha_1 + x_1, \ldots, \alpha_L + x_L)$, i.e. the prior distribution has the same form as the posterior distribution. In other words the Dirichlet distribution is the conjugate prior for the multinomial distributions. This property reduces the updating of the prior based on the observed value, to a simple update of the parameters in the prior density. It is therefore natural to use a Dirichlet distribution in the context of inference of a finite distributions.

2. The Dirichlet distribution has the following nice property that is particularly useful. If $\mathcal{X} = (X_1, \ldots, X_L)$ has a Dirichlet distribution, $\mathcal{D}(\alpha_1, \ldots, \alpha_L)$, then the marginal distribution of each component X_l follows a beta distribution [1]

$$X_l \sim \mathcal{B}(\alpha_l, A - \alpha_l)$$

where A, defined as $A = \sum_{l=1}^{L} \alpha_l$, is called the mass-value. The mean is given by $\mathbb{E}\{X_l\} = \frac{\alpha_l}{\sum_{l=1}^{L} \alpha_l}$ and the variance is given by $\text{Var}(X_l) = \frac{\alpha_l (A - \alpha_l)}{A(A+1)}$. In other words the variance of all components X_l is governed by the mass-value, A.

3. The Dirichlet distribution can be simulated easily by the following normalization construction. Suppose $Z_1, ...,$

[1] The beta distribution $\mathcal{B}(\alpha, \beta)$ is defined with a pdf

$$f(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} (1 - x)^{\beta - 1} x^{\alpha - 1}$$

where $0 \geq x \geq 1$.

Z_L are L random variables following a gamma distributions $\gamma(\alpha_1, 1), ..., \gamma(\alpha_L, 1)$[2] respectively, where

$$\gamma(a, b)(x) = \frac{1}{\Gamma(a)} b^a e^{-bx} x^{a-1} I_{(x>0)}.$$

If we normalize each random variable Z_l by the sum $Z = Z_1 + ... + Z_l$, then Z_l/Z has a beta distribution, and the multivariate random vector $(\frac{Z_1}{Z}, ..., \frac{Z_L}{Z})$ will follow a Dirichlet distribution $\mathcal{D}(\alpha_1, ..., \alpha_L)$.

These three properties make the Dirichlet distribution very attractive for modeling random distributions. Moreover Dirichlet distributions, and more specially the mixtures of Dirichlet distributions (to be defined later in the paper), have demonstrated, in practice, a good ability to model a very large spectrum of different distributions observed in the real world [14, 5, 6].

A random distribution following the Dirichlet distribution is called a Dirichlet process. The mean distribution of the Dirichlet Process will be defined as $\mathbb{E}\{\mathcal{X}\} = (\mathbb{E}\{X_1\}, \ldots, \mathbb{E}\{X_L\}) = (\frac{\alpha_1}{A}, \ldots, \frac{\alpha_L}{A})$. As explained in the second remark above each bin will follow a beta distribution and their variance will be jointly governed simply by the mass-value A which acts as a dispersion parameter that controls the dispersion of the Dirichlet process $\mathcal{X}$ around its mean $\mathbb{E}\{\mathcal{X}\}$. More generally for every arbitrary distribution $f(x)$ with cumulative distribution function $F(x)$, we can construct a Dirichlet process with a mean distribution that approximates the distribution $f(x)$ over the set of L bins $\mathfrak{B} = \{[b_0 = -\infty, b_1), [b_1, b_2), \ldots, [b_{L-1}, b_L = +\infty)\}$. For this purpose it suffices to set the values of $\alpha_i = A(F(b_i) - F(b_{i-1}))$. The resulting Dirichlet process will have a mean value given by $((F(b_1) - F(b_0)), \ldots, ((F(b_l) - F(b_{l-1}))$ and a variance that can be set arbitrarily by choosing the mass-value A. This last construction illustrates the power and the flexibility of Dirichlet distributions in the context of this study.

3.2 Modeling multiple classes

Mixed Dirichlet Processes (MPD) are often used as a flexible and practical way for modeling prior distributions in nonparametric bayesian estimation. The rationale for using Dirichlet mixtures is well explained in [14]. They are becoming increasingly popular for modeling distributions when conventional parametric priors would impose unreasonably stiff constraints on the distributional assumptions. Examples of applications include empirical Bayes problems [5], nonparametric regression [10] and density estimation [6].

In this paper we want to classify observed flows based on the similarity of their distribution. We assume that the observed empirical histograms are coming from a source governed by a random distribution. Rather than finding a single distribution to represent all flows, it makes intuitive sense to think of each class of traffic as having its own distribution. The entire ensemble of n flows $\mathcal{X} = \{X_{i*},\ i = 1, \ldots, n\}$, that contains the empirical histogram for each flow, is modeled as a mixture of multiple Dirichlet processes defined as $\sum_{k=1}^{K} p_k \mathcal{D}(\alpha_1^k, ..., \alpha_L^k)$, where each component $\mathcal{D}(\alpha_1^k, ..., \alpha_L^k)$ represents a traffic class, and each p_k represents the weight assigned to the class. This mixture defines the so called *a priori* probability. Now each observed histogram is assumed to come from one of these components. The classification problem consists of determining from which source component each histogram could have originated. To solve this problem, we need to find out the *a posteriori* probability, *i.e.* the probability that a flow belongs to a class given the histogram of the flow.

MDPs inherits the nice properties of Dirichlet processes we described in the previous section. Any particular probability density can be approximated over a bin set $\mathfrak{B}$ by a MDP with suitable parameters. Moreover the mass-value of each component controls the extent to which the model is allowed to diverge from its specified mean behavior. So MDP doesn't contain as much *a priori* as a normal or Poisson distribution.

Let K denote the number of traffic classes into which we want to classify our Internet flows. We model our observed histograms by assuming that the distribution of bins $Pr(X_1, \ldots, X_L)$ can be described by a finite mixture of K Dirichlet distributions:

$$Pr(X_1, \ldots, X_L) = \sum_{k=1}^{K} p_k \mathcal{D}(\alpha_1^k, ..., \alpha_L^k)$$

where the coefficients $p_1, \ldots, p_K$ denote the weight, or contribution, of each Dirichlet density. This gives the prior distribution, that is the probability that one observes $(x_1, \ldots, x_L)$ given that the parameters are fixed at $p_1, \ldots, p_K$ and $\alpha_1^k, ..., \alpha_L^k$ for $k = 1, \ldots, K$. However in practice these parameters are unknown and in order to finalize our model, we need to estimate them. Based on this *a priori* probability we need also to obtain the *a posteriori* or the class membership probability, *i.e.* the probability that a flow belongs to a class given the histogram of the flow. In the next section we present the estimation procedure for estimating these parameters based on our data.

3.3 Estimation procedure

Before discussing the estimation of the Dirichlet mixture process, we first recall the mixture problem when the observations are real vectors. Let $X_1, X_2, ..., X_n \in \mathbb{R}^L$ be n observations (flows) from a random vector of dimension L. The problem consists in estimating the distribution $\mathcal{P}_X$ of X when $\mathcal{P}_X$ is supposed to be a convex combination $\sum_{k=1,\ldots,K} p_k P_k$, in which the distributions P_k belong to a specific parametric family, say the exponential family. Several methods have been proposed to estimate the mixing weights p_k and the parameters of the components P_k; here we use one of the most efficient methods called SAEM, a Simulated Annealing Expectation Maximization algorithm [3]. SAEM is a stochastic approximation of the popular Expectation Maximization (EM) algorithm [4] that is less sensitive to local minima problems.

The EM algorithm is a general method of finding the maximum likelihood estimates of the parameters of an underlying distribution from a given data set when the data is incomplete or has some unknown parameters. The EM method is based on an iteration between an Estimation and a Maximization step. The usage of the EM algorithm in the case of mixture models is well described in [1].

SAEM, as first described by Celeux and Dielbot in [3], modifies the EM methods to get rid of common problems encountered such as slow convergence or local maxima. Instead of using a prior distribution for the unknown parameter it involves a stochastic step that simulates the unknown data in order to obtain complete data and to uncover hidden

[2] Recall that the family of gamma distributions contains exponential, as well as erlang distributions

variables.

Our algorithm takes as inputs the histograms, the number of desired classes K, and a sequence of values γ_q. These values γ_q are used to control the tradeoff between the influence of the stochastic step and the EM steps. Let $\{\gamma_q\}$ be a sequence of positive real numbers decreasing to zero at a sufficiently slow rate, with $\gamma_0 = 1$. Each time the algorithm iterates, repeating the E and M steps, the impact of the stochastic EM component is successively reduced (by multiplying with smaller and smaller γ_q). When γ_q approaches zero, our algorithm reduces to a pure EM algorithm.

Our algorithm outputs three things: the weights, p_k, of each Dirichlet process; the Dirichlet parameters $\alpha = (\alpha_1, ..., \alpha_L)$; and the class membership probabilities t^q_{ik}, where t^q_{ik} denotes the probability that flow i belongs to class k at the q-th iteration of the algorithm. This algorithm asymptotically estimates the parameter of the mixture model since p_{qk}, t^q_{ik} and the density parameters converge as $q \to \infty$ [3].

A general formulation of the SAEM for the large class of mixtures of density functions belonging to the exponential family has the form:

$$d(x, a) = d^{-1}(a)e(x) \exp < a^T . b(x) >$$

where the parameter a is a vector with transpose a^T, $d(a)$ is a normalizing factor, e and b are fixed but arbitrary functions and $< . >$ is the standard inner product.

In adapting this to our problem, the case of Dirichlet mixtures, we need to set the parameters as follows: $a = (\alpha_1, \ldots, \alpha_L)$, $b(x) = (\log(x_1), \ldots, \log(x_L))$, $d(a) = \frac{\Gamma(\alpha_1)\ldots\Gamma(\alpha_L)}{\Gamma(\alpha_1+\ldots+\alpha_L)}$ and $e(x) = x_1^{-1} \ldots x_L^{-1}$. The inputs are the n vectors $X_{i*}, i = 1, ..., n$ where each observation X_i* is a normalized histogram. The number of components in the mixture is a given integer K assumed to be known.

Our algorithm is given in the figure labeled Algorithm 1. This algorithm contains three main steps:

- A simulation step that introduces some noise into the process by making a random class assignment. This noise helps to push the algorithm out from local minima. However since the parameter γ_q is decreasing, the noise decreases as well, and the algorithm will converge to a stable estimate. A threshold $c(n)$ is used where $0 < c(n) < 1$ and $\lim_{n\to\infty} c(n) = 0$. This threshold determines whether or not one needs to return to the initialization step and essentially start over.

- A maximization step that updates the parameter values a^{q+1}_k, as well as the mixing weights $p_{(q+1)k}$, such that the likelihood is maximized. (Recall that the a^{q+1}_k variables in the algorithm correspond to the α variables in our model as stated above.)

- An estimation step in which we update the membership probabilities t^q_{ik}, i.e., the probability that flow i belongs to class k (at the q-th iteration through the algorithm). Recall that this is our *posterior* distribution (in Bayesian terms).

4. VALIDATION

In this section we validate our classification algorithm using synthetic data. Synthetic data is needed for this stage because we need to know ahead of time what the "true" classification is of each flow so that we may compute error rates in the model's classification. We test our estimation method using "hard" cases in order to test whether it can make fine distinctions in flow behavior that might otherwise be blurred by simplistic classification schemes.

4.1 Bin sizing for histograms

The choice of the number of bins and the location of the bin centers ($\mathfrak{B}$) is important. On the one hand the larger the number of bins the more accurately our empirical histogram will represent the real distribution. On the other hand, if there are too many bins, some bins might remain empty and the estimation algorithm can fail (because an empty bin gives a likelihood of zero). Our experience has shown that 20 bins achieve a good tradeoff between these two issues. As explained before it is desirable that bins are not empty, however the choice of bin centers can affect the accuracy of the classification algorithm particularly if some bins end up being empty. As a heuristic we try to find an algorithm that selects the bin centers such that all bins have roughly the same number of members.

Using these bins, we derive histograms for each observed flow. Recall that the goal is to find K classes that represent the ensemble of all of these histograms. This number K thus also defines the number of Dirichlet processes in the mixture model.

4.2 Test Cases

To test the estimation procedure we have defined three test sets.

1. In the first test, we use the normalization of a gamma distributed random vector procedure, described in the third remark under the definition of the Dirichlet distribution, to generate histograms following a Dirichlet distribution. Two series of histograms over 20 bins containing respectively 500 and 100 flows are generated using the above described method. Each sequence of histograms follows a distinct Dirichlet distribution with distinct parameters. We have applied the previously described estimation procedure to reclassify blindly the histograms. The estimation procedure has given a set of values of α as well as the posterior probability of class membership for each histogram. The results of the classification are very good. All flows are correctly reclassified and moreover the parameters of the α's of the initial Dirichlet processes are estimated very faithfully.

2. In the second test case, we have assumed that flows are generated by two classes whose flows follow a normal distribution. The first class of flows is normal with the parameters $\mu_1 = 200$ and $\sigma_1 = 10$; while the second class of flows has parameters $\mu_2 = 210$ and $\sigma_2 = 20$. We generated 500 flows in class 1 and 100 flows in class 2 for a total of 600 flows. Each flow consists of 288 samples (equivalent to the realistic situation of one day of measurements with 5 minute reporting intervals). We pick two classes whose means are fairly close together because this creates a hard test case in that the method needs to be able to distinguish two somewhat similar classes.

   These flows are transformed to empirical histograms as described in Section 2. Our estimation of the two

```
Initialization step :
  Assign randomly each flow $i$ to a class.
  Simulation step :
    Generate randomly $t_{ik}^{(0)}$ $(i = 1, \ldots, n)$ representing the initial a posteriori probability that a flow $i$ is in class $k$ where $1 <= k <= K$.
  for $q = 0$ to $Q$ do
      Stochastic step:
        Generate random multinomial numbers $e_{qi} = (e_{qi}^k)$ following the probability distribution $\{t_{ik}^q\}$ where all the $e_{qi}^k$ are 0 except one of them equal to 1.
        if $\frac{\sum_{i=1,\ldots,N} e_{qi}^k}{N} < c(n)$ for some $k$ then
            Return back to initialisation step.
        end
      Maximization step :
        Estimate the mixing weights $p_{(q+1)k} = \frac{1}{n}[(1-\gamma_q)\sum_{i=1,\ldots,n} t_{ik}^q + \gamma_q \sum_{i=1,\ldots,n} e_{qi}^k]$.
        and the parameter value $a_k^{q+1} = (1-\gamma_q)\frac{\sum_{i=1,\ldots,n} t_{ik}^q b(f_i)}{\sum_{i=1,\ldots,n} t_{ik}^q} + \gamma_q \frac{\sum_{i=1,\ldots,n} e_{qi}^k b(f_i)}{\sum_{i=1,\ldots,n} e_{qi}^k}$
      Estimation step :
        Update the a posteriori probability of a flow $i$ belonging to class $k$ $(t_{ik}^{q+1})$ according to $t_{ik}^{q+1} = \frac{p_{(q+1)k} h_{(q+1)k}(f_i)}{\sum_{r=1\ldots K} p_{(q+1)r} h_{(q+1)r}(f_i)}$
  end
```

Algorithm 1: SAEM Algorithm

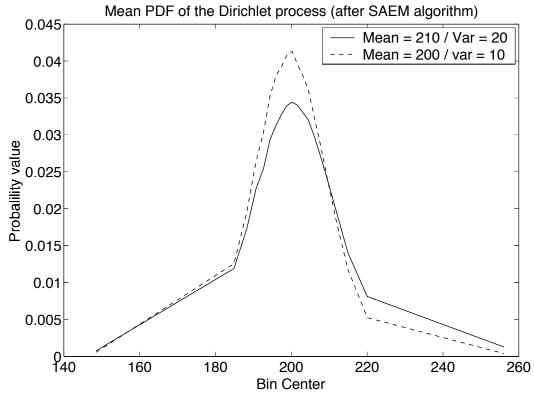


Figure 4: Estimated Classes for Synthetic Dataset 1

classes is given in Figure 4. Because the Dirichlet distribution is a multi-dimensional entity, it is difficult to plot. To provide some way to visualise as to what our model produces, we provide here the PDF of the mean of the Dirichlet mixture process $E(\mathcal{X})$, which is itself a random variable. We plot the mean PDF for each of the two derived classes.

The results of the classification are very significant. Out of 600 flows, only one flow is misclassified (a flow with a mean of $\mu_1 = 200$ and $\sigma_1 = 10$ is classified in the other class)! To the best of our knowledge, no other classification technique is able to reach such a high success ratio when classifying classes that are so similar.

3. In the third test case we generate two sequences of flows following two distributions with differing weight in the tail. The first distribution is simply a gamma distribution, the second distribution is a mixture of the gamma distribution with a Pareto distribution whose parameter is equal to 2.5 (this leads to a heavy tailed distribution). We generated 500 flows from each class and the histograms were constructed over 20 bins as described formerly. Then we applied our estimation procedure to these 1000 flows. The results are also excellent here. The classification had a 100 classification procedure can be readily applied even to separate a heavy tailed distribution from a light tailed one.

These three test cases validate the classification algorithm and justify its application to realistic data that will be described in the next section. We point out that the last two test cases illustrate that even when the data does not follow a gamma or beta distribution (as would be implied by the model), the Dirichlet process can be made to fit the data well since it is such a flexible polynomial prior. This is why Dirichlet processes are so good for modeling datasets with unknown distributions.

5. RESULTS

The data used for this classification study was described in Section 2. Recall that we have two traces from two backbone links, each of which we split into three back-to-back 24-hour traces. Thus each trace now contains 288 measurements (one measurement taken every 5 minutes) for approximately 2000 flows. We grouped these 288 measurements into 20 bins.

We now apply the proposed algorithm to our collected packet traces from backbone OC-12 links. We studied the traces from both links, but include only one here for ease of presentation; the results are similar in both cases. We classify flows using both 2 classes, and then 4 classes. We explain what is learned when allowing more classes to be considered.

Our goal in this section is not to present definitive results on how internet flows should be classified, but rather to illustrate the proposed method and to see what we may learn from it. We are aware of the fact that the results presented here are not sufficient for finalizing general insight about the behavior of aggregate flows. A study applying this classification approach to a large set of data traces is currently under way and is not in the scope of this paper.

5.1 Classification with two classes

We now take one of our packet traces and classify all the flows into two classes. Figure 5-a shows the distribution (PDFs and CDFs) of the mean for each class. Recall that we are plotting the PDF of the mean of the Dirichlet process, $E(\mathcal{X})$, which is a random variable. We give this as our visualization because Dirichlet processes are not easy to plot since they are multivariate distributions. We found that 749 flows (41 belong to class 1 and that 1051 flows (59 2. In the second class, the vast majority of flows typically have small values. The mean behavior of class 2 has an exponential like form that is evidenced by the close linear alignment over the range of 100 bytes/sec to 1 Mbyte/sec. In the first class, the vast majority of flows experience larger values and are almost never close to zero. This empirical classification corresponds to the well known elephants and mice phenomenon. In Figure 5-b we can see that flows in class 1 contribute roughly 90 throughout the day. We therefore use the standard terms and label the first class of flows as elephants and the second class as mice. While we picked the target number of classes here, these figures show that the classes selected by our method conform to our intuition and experience with Internet flows.

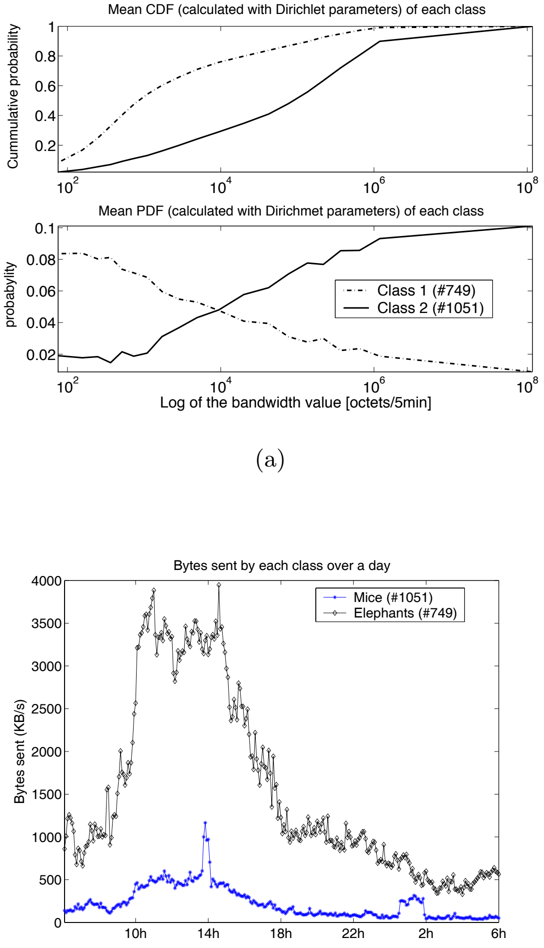
Note that within the elephant class, we can still see some flows whose mean bandwidth is small. This is as it should be. Flows can have small mean bandwidths but may also experience a few large bursts. Our method is classifying some of these flows as elephants. Similarly a mouse flow that experiences one short burst could have a reasonably large mean, but should still be classified as a mouse if most of the time its bandwidth is small. Our method is thus handling well the “flows in the middle” or those that exhibit a bit of both behaviors.

5.2 Classification with more classes

We now go hunting on safari to see if we can discover other animals with differing behavior. In this section we consider classifying our flows into four classes. The motivation for doing so is the following. When using a small number of classes, there may be batches of flows within one class that behavior substantially differently from the rest of the flows in that same class. Using a larger number of classes permits such finer level distinctions to be drawn. By running our algorithm with a target of four classes, we will see via the output whether it draws meaningful differences among classes. What we did was to tell our method to subdivide each of the two elephant and mice classes into two classes, yielding a total of four.

In Figure 6 we plot the CDF of the flow means. We do see here four distinct CDFs for the four classes. Probing futher, we examine the contribution of each class to the overall traffic in Figure 6-b. We see that class 1 contains 20 flows that generate 70 together contain 58% of the flows but only generate 13 traffic. We will call these flows elephants (class 1), buffalos (class 2), mice (class 3) and dragonflies (class 4).

Note that the original elephant class has now been split into two classes (called elephants and buffalos). The new class of buffalos has substantially lower average rate than the elephants. To see whether there is more meaning to this differentiation, we examine the time series of some sample flows from each of these two classes in Figure 7. It appears that buffalo flows are more prone to burst suddenly, and then quickly drop off to low levels, whereas elephants flows have more inertia thus exhibiting large volumes for long durations. This figure thus suggests that buffalos have a stronger spiky behavior (short lived bursts) than elephants do.

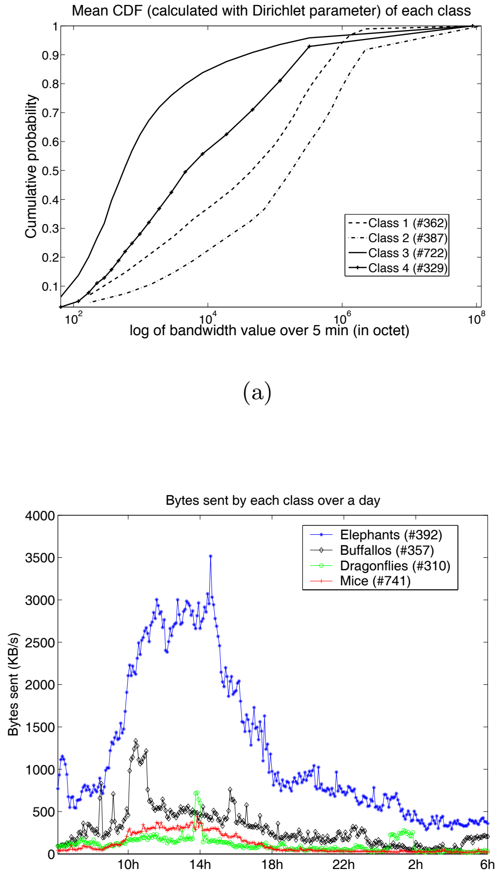


(a)

(b)

Figure 5: (a)Mean class PDF/CDF for a two classes classification (b) Contribution of each classes to the total traffic during 24 hours

To validate the hypothesis that our algorithm is differentiating buffalos from elephants based on their burst behavior we did the following. For each buffalo and elephant flow we calibrated a two state hidden Markov model. In each state, the flow is assumed to follow a Gaussian distribution and transitions between states occur when the flow goes from a low rate (with small mean) to a high rate (with large mean) and vice-versa. We used the well known Baum-Welches equations to infer the transition matrix for each flow, as well as the parameters of the Gaussian distribution in each state. Details of this procedure are given in [11]. We computed the mean holding times in each state based



(a)

(b)

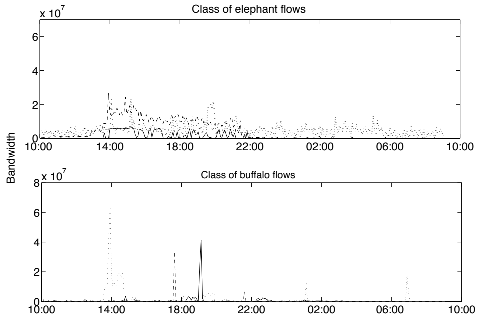
Figure 6: Results of classification for 4 classes. (a) Mean CDF of the 4 inferred classes (b) Contribution of each classes to the total traffic during 24 hours

on the transition matrix.

The CDF of the holding times in the high state for both buffalos and elephants is given in Figure 8. This plot clearly shows that buffalos have short holding times in the high state, implying that their bursts are typically much more short-lived than those of elephants. This distribution computed over all flows validates the behavior illustrated in Figure 7; namely that buffalos are more spiky than elephants.

This analysis reveals the power of our classification method. It differentiates flows not only according to their mean behavior but also according to specific aspects of their variability behavior. It is drawing a distinction related to the time scale of variability. This shows that our method can draw fine distinctions between groups of flows that are based on it knowing more than simply the mean and standard deviation. Classification based on the (mean,std) tuple would not be able to make use of temporal characteristics.

To complete the story let's also examine the mice and dragonflies. The lower curve in Figure 5 corresponding to mice shows two burst episodes (at 14h and 1h). When these flows are further separated in two classes, as in Figure 6, we see that the (new) mice do not exhibit any bursts, and that all the bursts have been attributed to dragonflies. Our classification is able to draw fine distinctions between classes because we give it entire histograms to work with, and because it uses a very general modeling distribution (e.g, the random Dirichlet distribution).



(a)

Figure 7: Typical temporal (24 hours) behaviour of Elephants and Buffalos

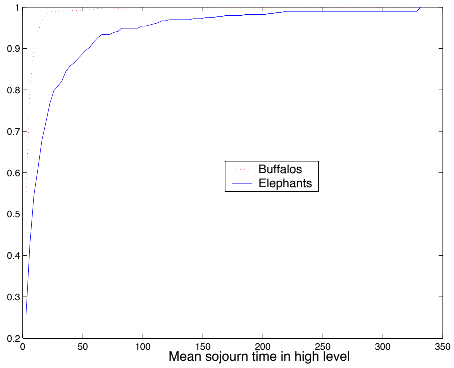


Figure 8: Mean holding times in the high state for elephants and buffalos

An important point that is out of the scope of this paper is the issue of exactly how many classes are needed to characterize the set of flows on an aggregated link. The answer to this question is not as simple as "the larger the number of classes, the more precise the classification". The level of precision needed will clearly depend upon what the classification is used for. One promising approach is based on the entropy inspirated approach developed in [15], where it is shown that the log-likelihood of an inferred classification is upper bounded by the entropy of the source generating the observations.

This subject is currently under investigation.

5.3 Stability analysis

One known problem observed in previous attempts to classify network flows, was the instability of classification. This came from two sources: first, different classifications arose when the scheme is executed at different moments in time; second, there are many "flow in the middle" that had a tendency to oscillate between classes, morphing from elephant to mice and vice-versa, in successive snapshots [12]. We decided to evaluate the sensitivity of our approach to these stability problems. For this purpose we have run the classification algorithm with 4 classes on each of the three consecutive days separately. We compare the obtained mean class behaviour in Figure 9.

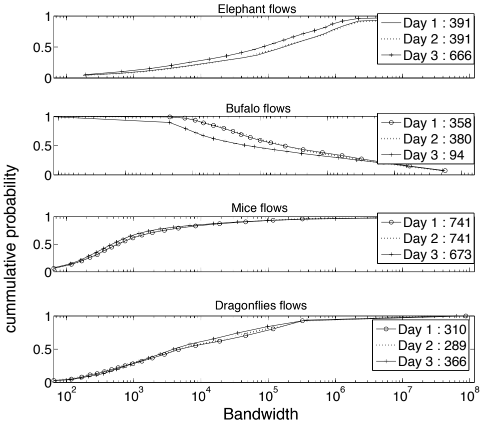


Figure 9: Mean class behaviour on three different days

We see that our classification of mice and dragonflies remains very consistent over the three days. The class definitions for elephants and buffalos are very consistent for the first two days. In the third day we see that the buffalos species starts to experience some extinction as they morph to elephants. The number of buffalos falls from 380 to 97 on day 3, while the number of elephants rises from 392 to 666. The mean behavior of elephants also changes on the third day. We point out that this third day is a Saturday, and is thus most likely related to the difference between weekday and weekend traffic patterns.

The number of class changes from day 1 and 2 is very small (around 90 changes in 1800 flows). The given analysis shows that the approach developed in this paper is able to detect changes due to weekly variation pattern. It also shows that the classes of mice and dragon flies are very stable during the time. This suggests that the proposed classification is at least robust at a small of some days. This result need also more confirmation over larger data set covering longer period of time.

6. CONCLUSIONS

In this paper we develop a new method for classification of Internet prefix level flows. We argue that classification of highly aggregated flows should be done using histograms that capture the entire distributional properties of flows. This enables classification schemes that are able to draw fine distinctions about flow behavior even when operating on macroscopic flows.

When using historgrams for classification we need models based upon random distributions. We use Dirichlet processes to model traffic classes because they are very flexible distributions that can easily be parametrized to fit a wide variety of distributional forms. Because we use histograms for classification combined with Dirichlet processes, we build an extremely flexible classifier. We use a mixture of Dirichlet processes with one process per class. The parameters of the mixture model are estimated using a variant of the Expectation Maximization algorithm, called the Stochastic Annealing EM.

We validated our model against three hard synthetic test cases. Each of the test cases yielded 99% or 100 classification. We then applied our method to data collected from inside a Tier-1 carrier network. When using two classes, our method defined two classes whose properties behave according to the elephants and mice phenomenon we have come to expect. Our method is more meaningful than threshold based methods because it can classify flows that exhibit a bit of both behaviors (by determining which behavior is predominant). When using four classes, we see that our method is capable of drawing fine distinctions between the classes. Our method draws a distinction between elephants and buffalos (both large flows) according to their burst behavior. The buffalo class exhibits greater short-term spikiness than the elephants. This illustrates the benefit of using histograms since a distinction that includes a temporal notion has been incorporated into the classification.

Our goal was to introduce a new method and provide a proof of concept by illustrating its abilities on Internet flows. A wider study using a larger data set is needed before general conclusions can be drawn about the data itself. Such a study is under way and the next step in our study will be to develop a class of parametric models for the mean class behaviour. This parametric class will be used as a prior in real time operational traffic classification by using bayesian estimation.

7. ADDITIONAL AUTHORS

8. REFERENCES

[1] Jeff Bilmes. A gentle tutorial on the EM algorithm including gaussian mixtures and baum-welch. Technical Report TR-97-021, International Computer Science Institute, Berkeley, CA, 1997.

[2] N. Brownlee and KC. Claffy. Understandin internet traffic streams : Dragonflies and tortoise. *IEEE communication magazine*, pages 110–117, october 2002.

[3] Gilles Celeux, Didier Chauveau, and Jean Diebolt. On stochastic versions of the EM algorithm. Technical Report RR-2514, INRIA, 1995.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. 39:1–38, 1977.

[5] Michael D. Escobar. Estimating normal means with a dirichlet process prior. *Journal of the American Statistical Association*, 89(425):268–277, march 1994.

[6] Michael D. Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[7] Cristian Estan and George Varghese. New directions in traffic measurement and accounting. In *Proceedings of the First ACM SIGCOMM Workshop on Internet Measurement Workshop*, pages 75–80. ACM Press, 2001.

[8] R.Emilion Mixtures of orthogonal random distributions and clustering. In *Compte Rendu Academie des Sciences de Paris I*, 2002(335):189–193.

[9] A. Lakhina, K. papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *ACM Sigmetrics*, New York, June 2004.

[10] P. Muller, A. Erkanli, and M. West. Bayesian curve-fitting using multivariate normal mixtures. *Biometrika*, 83(1):67–79, 1996.

[11] A. Oveissian, K. Salamatian, and A. Soule. Flow classification on short time scale. Technical Report *, LIP6, 2003.

[12] K. Papagiannaki, N. Taft, and C. Diot. Impact of flow dynamics on traffic engineering design principles. In *IEEE Infocom*, Hong Kong, March 2004.

[13] S. Sarvotham J. Rexford and K.Shin. Load-sensitive routing of ling-lived ip flows. In *Proc. ACM SIGCOMM*, september 1999.

[14] Christian P. Robert. *The Bayesian Choice*. Springer, 2001.

[15] Kavé Salamatian and Sandrine Vaton. Hidden markov modeling for network communication channels. In *Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 92–101. ACM Press, 2001.

[16] Shriram Sarvotham, Rudolf Riedi, and Richard Baraniuk. Connection-level analysis and modeling of network traffic. ACM SIGCOMM Internet Measurement Workshop, August 2002.

[17] Iljitsch van Beijnum. *BGP Building Reliable Networks with the Border Gateway Protocol*. O'Reilly, 2002.